

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 1

# Cuprinsul cursului de azi

1. Aspecte organizatorice legate de cursul de IA
1. Prezentarea cursului de IA

# Echipa noastră



**Bogdan**



**Alexandra**



**Sergiu**



**Irina**

**Curs  
Săpt. 1-14  
(seria 36)**

**Lab + P  
Săpt. 1-7  
(361, 362)**

**Lab + P  
Săpt. 1-7  
(363, 364)      (361, 362, 363, 364)**

# Structura primului semestru

- <https://www.unibuc.ro/studii/structura-anului-universitar/>

## STRUCTURA ANULUI UNIVERSITAR 2022 – 2023

Studii Universitare de Licență și Masterat

Cauta:

### PERIOADA

### ◆ ACTIVITATEA ◆

#### Semestrul I

03.10.2022 – 23.12.2022

Activitate didactică

24.12.2022 – 08.01.2023

Vacanță de iarnă

09.01.2023 – 22.01.2023

Activitate didactică

23.01.2023 – 12.02.2023

Sesiune de examene

- 13 cursuri (11 cursuri în 2022 – nu facem pe 1 decembrie, 2 cursuri în 2023)
- 2 ore de laborator + 2 ore de proiect pe săptămână (se mai pierd pe 30 noiembrie, 1, 2 decembrie)

# Orar

## CTI Grupa 361

Universitatea din Bucuresti, Facultatea de Matematica si Informatica, str. Academiei 14, Bucuresti

	8 8:00 - 9:00	9 9:00 - 10:00	10 10:00 - 11:00	11 11:00 - 12:00	12 12:00 - 13:00	13 13:00 - 14:00	14 14:00 - 15:00	15 15:00 - 16:00	16 16:00 - 17:00	17 17:00 - 18:00	18 18:00 - 19:00	19 19:00 - 20:00
Lu				Micluta M  ProiectareaBD  L-204					Bobe R  ProiectareaBD  ONLINE			
Ma	Chiroscă A / Chiroscă G  ArhSistCalcul  Magurele, lab.			Diaconu A  IA [sapt 1-7]  L-321	Ciocan I  IA [sapt 8-14]  L-321	Diaconu A  IA [sapt 1-7]  L-321	Ciocan I  IA [sapt 8-14]  L-321	Dragulici D  SistOper  L-308				
Mi												
Jo			Stanculescu F  ArhSistCalcul  3(Titeica)	Alexe B  InteligArtificiala  3(Titeica)	Dragulici D  SistOper  3(Titeica)							

- Curs: Bogdan – săptămânal
- Laborator: Alexandra/Sergiu (săpt. 1-7) și Irina (ultimele 7 săptămâni) - săptămânal
- Proiect: Alexandra/Sergiu (săpt. 1-7) și Irina (ultimele 7 săptămâni) - săptămânal

# Materiale

- Pe team-ul nostru din MS-TEAMS

Curs (Inteligentă Artificială - CTI-3-2022/2023-FMI) | Microsoft Teams

The screenshot shows the Microsoft Teams interface for the 'Curs' channel of a team named 'Curs'. The left sidebar shows navigation links like Activity, Chat, Calendar, Assignments, and Teams. The main area displays a list of files in the 'Bibliografie' folder. The table has columns for Name, Modified, and Modified By.

Name	Modified	Modified By
Artificial Intelligence - A modern approach...	A few seconds ago	DUMITRU BOGDAN
Hands_On_Machine_Learning_with_Scikit...	A few seconds ago	DUMITRU BOGDAN

# Examen iarnă - evaluare și notare

Nota = min(Curs + Laborator + Proiect&Teme + BonusLab, 10)  
4p              3p              3p              maxim 1p

- Examen Curs (4 puncte) – scris, în sesiune, 2 ore + eventual bonus de la curs (nu puteți depăși 4p)
- Test Laborator (3 puncte) – în sesiune (în aceeași zi cu Examen Curs), 2 ore;
- Proiect (1,5 puncte) – proiect la prima parte (învățare automată, îl primiți în săptămâna 4), termen limită de predare – săptămâna 6, prezentarea proiectului în săptămâna 7.
- Teme (1,5 puncte) – teme la partea a doua (căutare informată și neinformată), prezentare în cadrul laboratorului.

Nu există praguri, note minime impuse. 4,99 înseamnă restanță.

# Restanță/reexaminare - evaluare și notare

Nota = min(Curs + Laborator + Proiect&Teme + BonusLab, 10)  
4p                  3p                  3p                  maxim 1p

- Examen Curs (4 puncte) – scris, în sesiune, 2 ore + eventual bonus de la curs (nu puteți depăși 4p)
- Test Laborator (3 puncte) – în sesiune (în aceeași zi cu Examen Curs), 2 ore;
- Proiect & Teme – păstrăm notele din timpul semestrului
  - dacă în timpul semestrului nu puteți participa din condiții obiective la proiect + teme îmi scrieți până la 15 octombrie ca să vedem cum puteți recupera această parte pentru restanță/reexaminare.

**Păstrăm bonusul de la curs și de la laborator. Vă puteți păstra nota de la curs/laborator din precedentele sesiuni ale anului curent. Nu există praguri, note minime impuse. 4,99 înseamnă restanță.**

# Mărire - evaluare și notare

Nota = min(Curs + Laborator + Proiect&Teme + BonusLab, 10)  
4p              3p              3p              maxim 1p

- Examen Curs (4 puncte) – scris, în sesiune, 2 ore + eventual bonus de la curs (nu puteți depăși 4p)
- Test Laborator (3 puncte) – în sesiune (în aceeași zi cu Examen Curs), 2 ore;
- Proiect & Teme – păstrăm notele din timpul semestrului
  - dacă vreți să refaceti partea de proiect sau teme îmi dați un email până pe 15 august

**Păstrăm bonusul de la curs și de la laborator.**

**Obligatoriu dați ambele probe (Scris + Test Laborator).**

# Restanță pentru 2023/2024

- nu ținem cont de niciun punctaj din acest an.

# Rotunjirea notelor obținute

- orice notă  $x < 5$  se rotunjește la partea întreagă  $[x]$ 
  - nota 4.99 se rotunjește la nota 4;
  - nota 3.4 se rotunjește la nota 3.
- pentru orice notă  $x \geq 5$ :
  - dacă partea fractionară  $\{x\} \geq 0.5$  atunci rotunjim la  $[x] + 1$ 
    - nota 9.5 se rotunjește la nota 10;
  - dacă partea fractionară  $\{x\} < 0.5$  atunci rotunjim la  $[x]$ 
    - nota 9.45 se rotunjește la nota 9;

# Bonusul de la laborator

- puteți acumula maxim 1 punct bonus de la laborator;
- la examenul final, se adaugă acest bonus la notă;
- pentru prima parte a semestrului aveți alocat un bonus de 0,5 puncte
  - Alexandra și Sergiu stabilesc care sunt aceste condiții (posibil să punteze prezența la 2 ore de laborator cu 0,05 puncte);
- pentru a doua parte a semestrului aveți alocat un bonus de 0,5 puncte
  - Irina stabilește care sunt aceste condiții;

# Bonusul de la curs

- puteți acumula maxim 1 punct bonus de la curs;
- la examenul final, cu tot cu bonus nu puteți depăși 4 puncte la curs;
- la sfârșitul fiecărui curs veți primi întrebări din materia predată la acel curs;
- vom folosi platforma Kahoot!
- număr de întrebări flexibil în funcție de fiecare curs;
- premiem la fiecare curs primii 15 studenți: primii 5 studenți iau 0.15p, următorii 5 studenți iau 0.10p, următorii 5 studenți iau 0.05p;

# Kahoot - test

- intrați de pe telefon/laptop pe site-ul [www.kahoot.com](http://www.kahoot.com) și selectați play
- introduceți codul de pe pagina mea/proiector
- vă alegeti un username, ideal prenume.nume (bogdan.alex)
- **cititi întrebarea de pe pagina mea/proiector, răspundeți pe telefon/laptop selectând răspunsul corect.**
- facem un test rapid să vedem că e totul ok!

# Regulament de integritate

- regulament privind activitatea studenților la UB:  
[http://fmi.unibuc.ro/ro/pdf/2015/consiliu/UB\\_Regulament\\_studenti\\_2015.pdf](http://fmi.unibuc.ro/ro/pdf/2015/consiliu/UB_Regulament_studenti_2015.pdf)
- regulament de etică și profesionalism la  
FMI:[http://fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament\\_etica\\_FMI.pdf](http://fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament_etica_FMI.pdf)

Se consideră **incident minor** cazul în care un student/ o studentă:

- a. preia codul sursă/ rezolvarea unei teme de la un coleg/ o colegă și pretinde că este rezultatul efortului propriu;

Se consideră **incident major** cazul în care un student/ o studentă:

- a. copiază la examene de orice tip;

- **3 incidente minore = un incident major = exmatriculare**

# Lucrare de licență?

- dacă vă place materia, luați notă mare la examen (minim 9) și vreți să vă faceți lucrarea de licență sub îndrumare mea putem discuta
- subiectul lucrării de licență poate fi propus de voi sau de mine (ideal alegem o temă interesantă pentru amândoi)
- trebuie să fie pe specializarea mea (inteligentă artificială)
- număr limită : 7 studenți

# Cuprinsul cursului de azi

1. Aspecte organizatorice legate de cursul de IA
  
1. Prezentarea cursului de IA

**Ce este  
Inteligenta Artificiala?**

# Homo sapiens

- omul intelligent;
- putem percepe, înțelege lumea care ne înconjoară, putem manipula obiecte și realiza predicții despre ce se va întâmpla în mediul în care suntem;
- ne dezvoltăm inteligența de-a lungul vietii pe baza învățării din experiențe, ne ajută la rezolvarea de probleme;
- putem realiza fără efort sarcini care pentru un computer sunt la acest moment (destul de) greu de realizat: descrierea în limbaj natural a ceea ce se întâmplă în jurul nostru, interacțiune cu mediul înconjurător (obiecte, persoane), comunicarea cu ceilalți, etc.

# Agenti inteligenți

- agent = entitate care percepă și acționează într-un anumit mediu
- exemple: om, robot de curățenie, mașină autonomă, program software care detectează spam-urile în Inbox, program software care detectează fețele oamenilor într-o imagine
- agent intelligent: agenți înzestrăți cu o “inteligentă” care îi ajută să ia deciziile optime în mediile lor, asemenea oamenilor

# Inteligentă Artificială – definiție



- Inteligență - abilitatea de a rezolva probleme
- Inteligență artificială - construirea de computere / scrierea de programe software inteligente pentru rezolvarea de probleme sau luarea deciziilor în mod automat în situații în care oamenii și-ar folosi inteligența



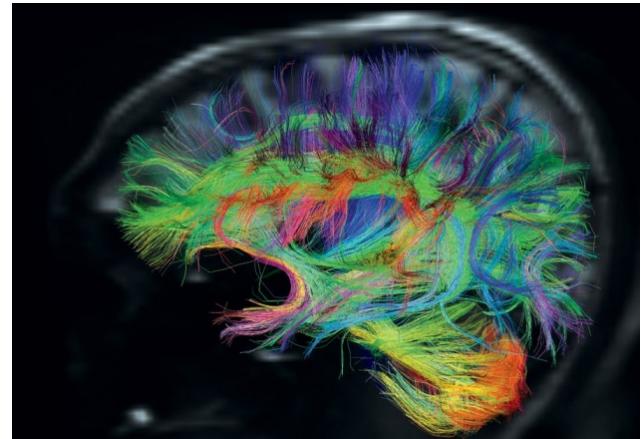
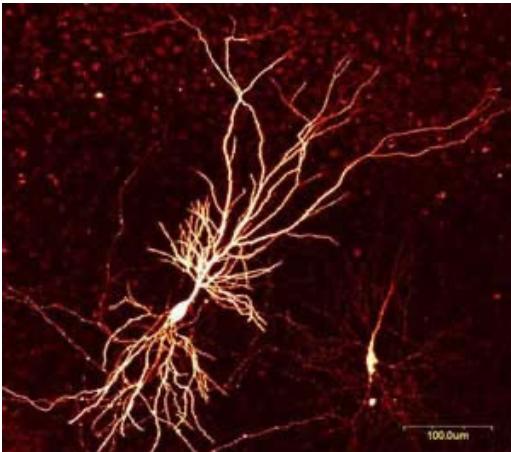
**INTELIGENȚĂ, *inteligēnțe*, s. f.** 1. Capacitatea de a înțelege ușor și bine, de a sesiza ceea ce este esențial, de a rezolva situații sau probleme noi pe baza experienței acumulate anterior; deșteptăciune. ♦ **Inteligență artificială** = domeniu al informaticii care dezvoltă sisteme tehnice capabile să rezolve probleme dificile legate de inteligența umană. ♦ **Persoană inteligentă.** 2. (în forma *inteligēnție*) Totalitatea intelectualilor; intelectualitate (2). [Var.: **inteligință, intelligēnție** s. f.] – Din fr. **intelligence**, lat. **intelligentia**, germ. **Intelligenz**, rus. **интеллигентия**.

# Inteligentă Artificială – definiție

	gândire	comportament
om	<b>1. ... simulează gândirea umană</b>	<b>2. ... se comportă ca un om</b>
rațiune	<b>3. ... simulează gândirea rațională</b>	<b>4. ... se comportă rațional</b>

# Definiția 1: simulează gândirea umană

- studiază creierul uman din perspectiva felului în care este procesată informația pentru rezolvarea de probleme
- științe cognitive = psihologie + IA + lingvistică + filozofie + științe neuronale
- două paradigmă: simbolică vs. conecționistă

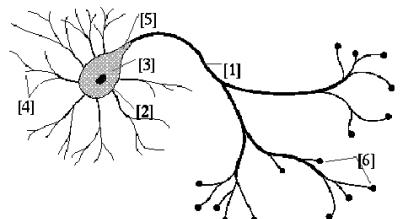


# Paradigma simbolică

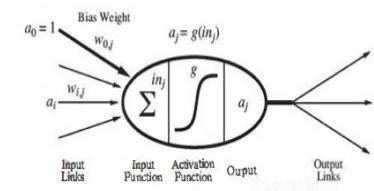
- sistemul cognitiv reprezintă cunoștințele și stările de lucruri corespunzătoare prin simboluri
- gândire = manipulare de simboluri
- când scriem un program într-un limbaj de nivel înalt (C, C++, Python) nu ne punem problema să descriem reprezentarea exact la nivel de bit a datelor, ci folosim tipuri de date prefabricate (simboluri)

# Paradigma conecționistă

- sistemul cognitiv reprezintă cunoștințele și stările de lucruri corespunzătoare prin activarea unor unități simple (neuroni) pe baza unor pattern-uri și valori de activare

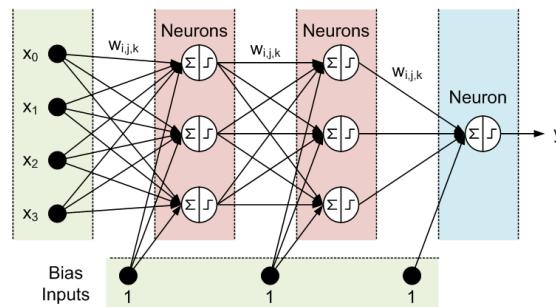


1.Axon 2.Nucleus (Body) 3.Soma (Body) 4. Dendrite 5. Axon Hillock 6. Terminals (Synapses)



- model matematic pentru neuron

- rețea neuronală



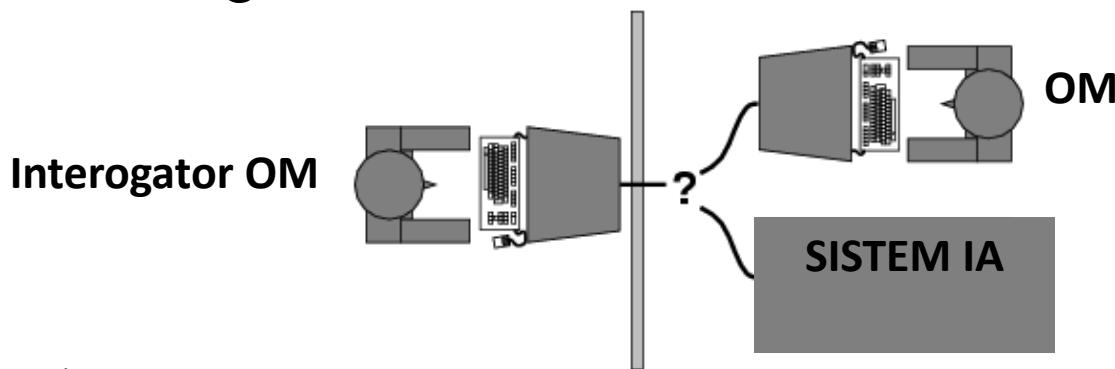


"I propose to consider the question,  
'Can machines think?'"

-Alan Turing

# Definiția 2: se comportă ca un om

- Testul Turing



- un interogator uman nu poate distinge în timpul unei conversații scrise dacă interlocutorul este calculator sau om
- Ce abilități ar trebui un calculator să aibă pentru a trece de testul Turing?
  1. procesarea limbajului natural (comunicare)
  2. reprezentarea cunoștințelor (stocare informații)
  3. deducție automată (a faptelor pe baza cunoștințelor)
  4. învățare automată (detectare de pattern-uri)

# Testul Turing

- Turing a prezis că până în anul 2000, un sistem informatic IA va putea să păcălească 30% din interogatori pentru 5 minute
- Premiul Loebner
  - 2008: 12 interogatori – 5 minute pentru a conversa simultan cu 2 entități diferite (om sau sistem AI) Câștigătorul, chatbot-ul Elbot, a reușit să păcăleacă 3 din 12 interogatori.



# Testul “total” al lui Turing

- Include semnal video – interogatorul uman poate testa capacitatele perceptuale ale interlocutorului
- Abilități necesare:
  1. procesarea limbajului natural (comunicare)
  2. reprezentarea cunoștiințelor (stocare informații)
  3. deducție automată (a faptelor pe baza cunoștiințelor)
  4. învățare automată (detectare de pattern-uri)
  5. vedere artificială (perceperea obiectelor, a scenei)
  6. robotică (manipularea obiectelor, mișcare)

Subdomenii ale Inteligenței Artificiale

# Definiția 3: simulează gândirea rațională

- gândirea ca proces care urmează anumite legi → Logica
- pornind de la premise corecte și urmând aceste legi se ajunge la concluzii corecte = raționamente
  - “Socrate este un om; toții oameni sunt muritori; prin urmare Socrate este muritor.”
- descrie problema în notație formală de logică și apoi aplică regulile de deducție pentru rezolvare
- dezavantaje:
  - complexitate computațională crescută
  - descrierea incertitudinii

# Definiția 4: se comportă rațional

- agent rațional care acționează pentru atingerea optimă a scopurilor luând decizii (inferențe)
  - scopuri exprimate în termeni de utilitate
  - a fi rațional = a maximiza utilitatea
  - în situații de incertitudine, alege decizia cu cea mai mare utilitate estimată (expected utility)
- comportamentul omului nu rezultă întotdeauna din rațiune + inferențe:
  - gesturi reflexe (când punem mâna pe ceva încins avem reflexul de a ne retrage imediat mâna)

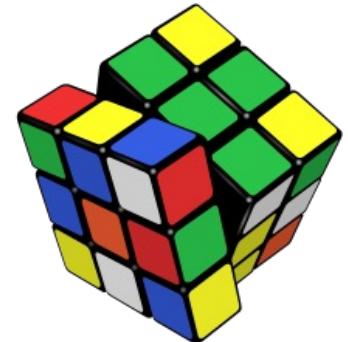
# Abordări în Inteligența Artificială

Două mari abordări:

- sisteme bazate pe reguli și logică
  - sisteme expert
- sisteme bazate pe învățare din date
  - învățare automată = machine learning

# Sisteme bazate pe reguli și logică

- reprezentarea cunoștințelor unui expert uman pentru rezolvarea unei probleme este exprimată sub formă de reguli
- regulile sunt codificate de calculator sub formă de blocuri de tipul *if-then-else*
- simulează inteligență, nu poate învăța, se limitează numai la regulile pe care le are



8		4	6		7
1				4	
5	9	3		6	5
			7		
4	8	2		1	3
5	2				9
	1				
3		9	2		5

# Învățarea automată (machine learning)

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Cum ați scrie un program pe calculator care vrea să găsească mașinile într-o imagine?



# Învățarea automată (machine learning)

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Cum ați scrie un program pe calculator care să filtreze mailurile spam?

The screenshot shows a list of detected spam emails. Each item has a warning icon (an exclamation mark inside a circle) and the subject line.

Spam Alert	Email Subject
!	The CHOICE Home Warranty
!	Final, Final (2)
!	Keep Your Home Cool with \$50 Off! – Keep Your Home Co...
!	Request To Terminate Your Google™ Account Has Bee...
This month	
!	Mr. Tony Adams (2)
!	ZENITH BANK PLC
!	Congratulation You Have Won – You have Won \$2.5Million ...
!	John Mill
!	COMPENSATION UNIT, IN AFFILIATION WITH THE UNITE...
!	UNITED NATION AND EUROPEAN UNION OFFICIAL WIN...

At the bottom left, there is a green circular icon with a white letter 'J' and the email address 'urytopk4@crocus.ocn.ne.jp'.

# Esența învățării automate

- Există un tipar
- Dar nu îl putem exprima programatic / matematic
- Avem date / exemple în care regăsim acest tipar



## Traditional Programming

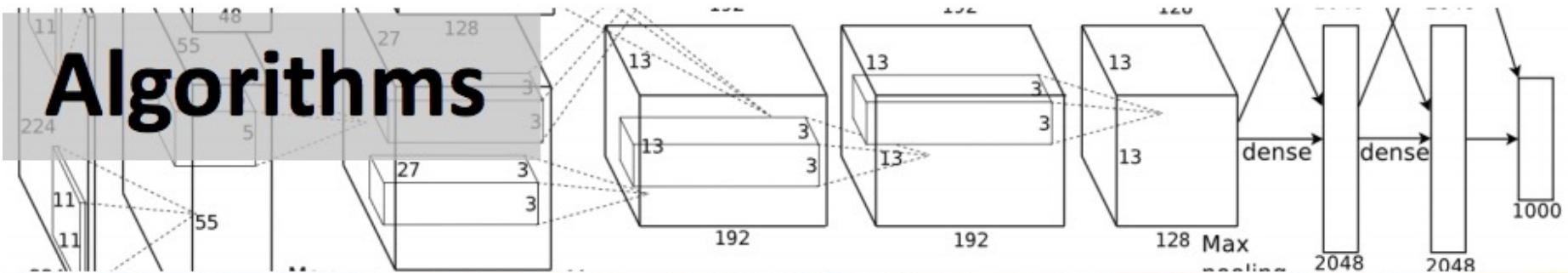


## Machine Learning



# Ingrediente pentru Învățare Automată

**Algorithms**



**Data**



**Computation**

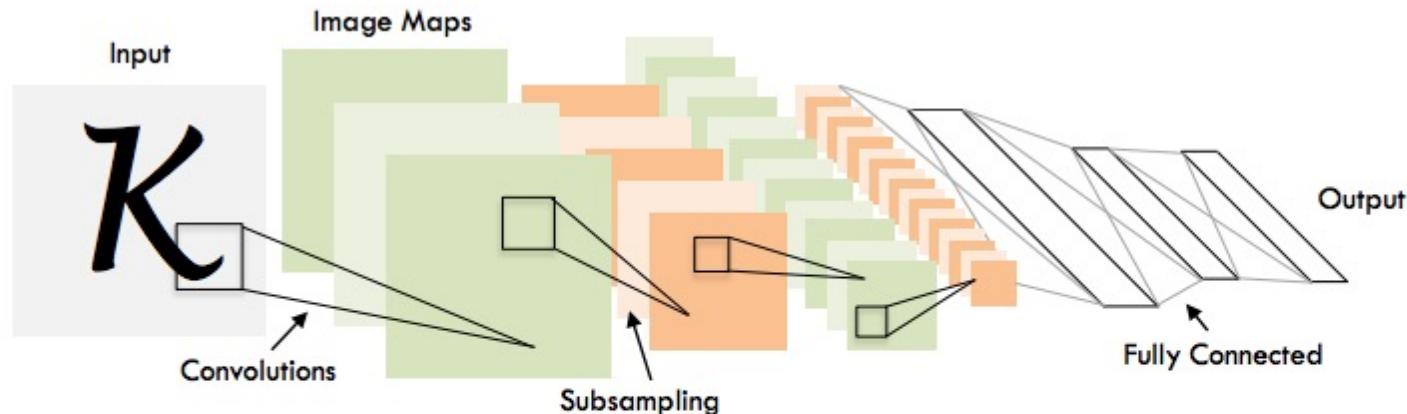


## Clasificarea imaginilor 1,431,167 imagini adnotate cu 1000 clase de obiecte



# 1998

LeCun et al.



# of transistors



$10^6$

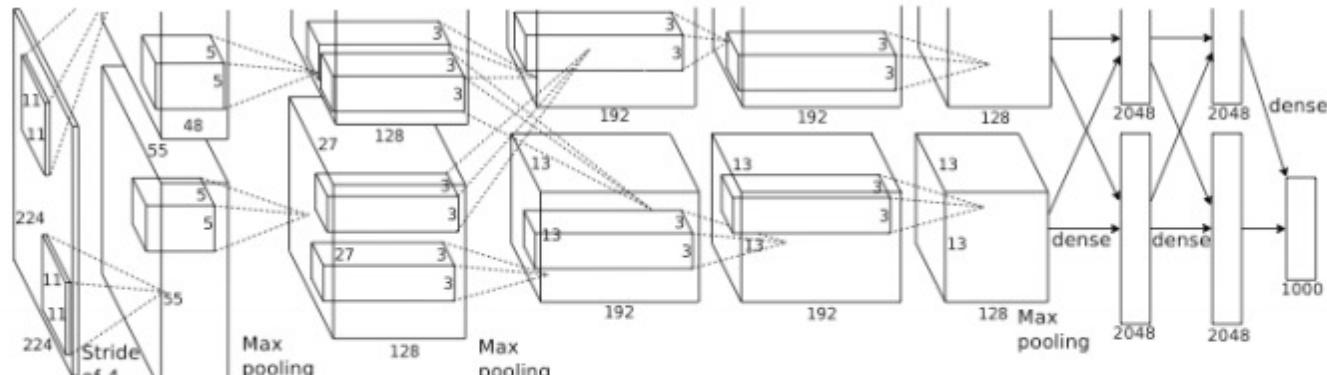
pentium® II

# of pixels used in training

$10^7$  NIST

# 2012

Krizhevsky et al.



# of transistors



$10^9$

GPUs



# of pixels used in training

$10^{14}$  IMAGENET

# Strong AI versus Weak AI

- ✗ • *strong / generic / true AI (vezi definiția lui Turing)*
  - mașini care gândesc la un nivel apropiat sau superior oamenilor
  - nu avem aşa ceva în zilele noastre
    - nu există nicio dovadă că suntem aproape

# Exemplu de strong AI



# Exemplu de strong AI



*Terminator 2*

# Strong AI versus Weak AI



- ***strong / generic / true AI (vezi definiția lui Turing)***
  - mașini care gândesc la un nivel apropiat sau superior oamenilor
  - nu avem aşa ceva în zilele noastre
    - nu există nicio dovadă că suntem aproape
- ***weak / narrow AI (se focusează pe o anumită problemă)***
  - simulează inteligența artificială
  - se bazează pe detectare de pattern-uri
    - paradigma dominantă azi în inteligența artificială



# Prima parte a semestrului: curs

## 1. Introducere în învățare automată

Etichetă

Caracteristică

Vector de caracteristici

Clasificare

Antrenare

Supervizare

Inferență

Model

Regresie

Overfitting

Matrice de confuzie

Acuratețe

Precizie

Recall

## 1. Metode de clasificare

K-NN

Naïve Bayes

SVM (Support Vector Machines)

Perceptron

Rețele neuronale feedforward multistrat

## 2. Metode de regresie

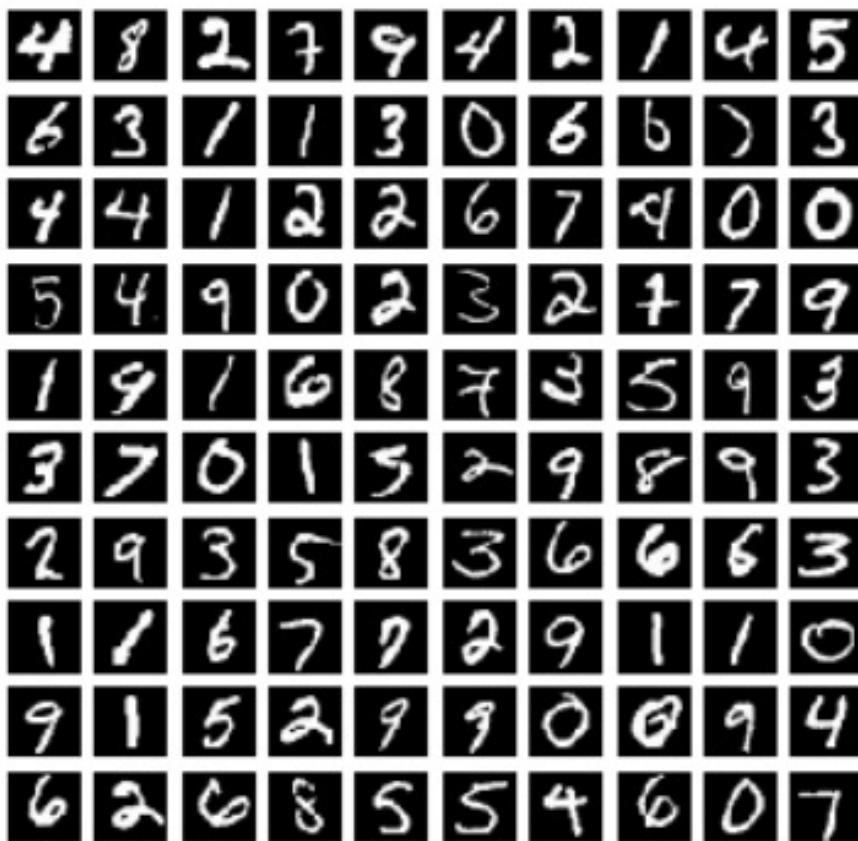
Regresie liniară simplă

Regresie liniară multiplă

# Prima parte a semestrului: laborator

- setul de date MNIST (cifre scrise de mâna). Setul original conține 50.000 de exemple de antrenare, 10.000 de exemple de testare.

Primele 100 exemple de antrenare



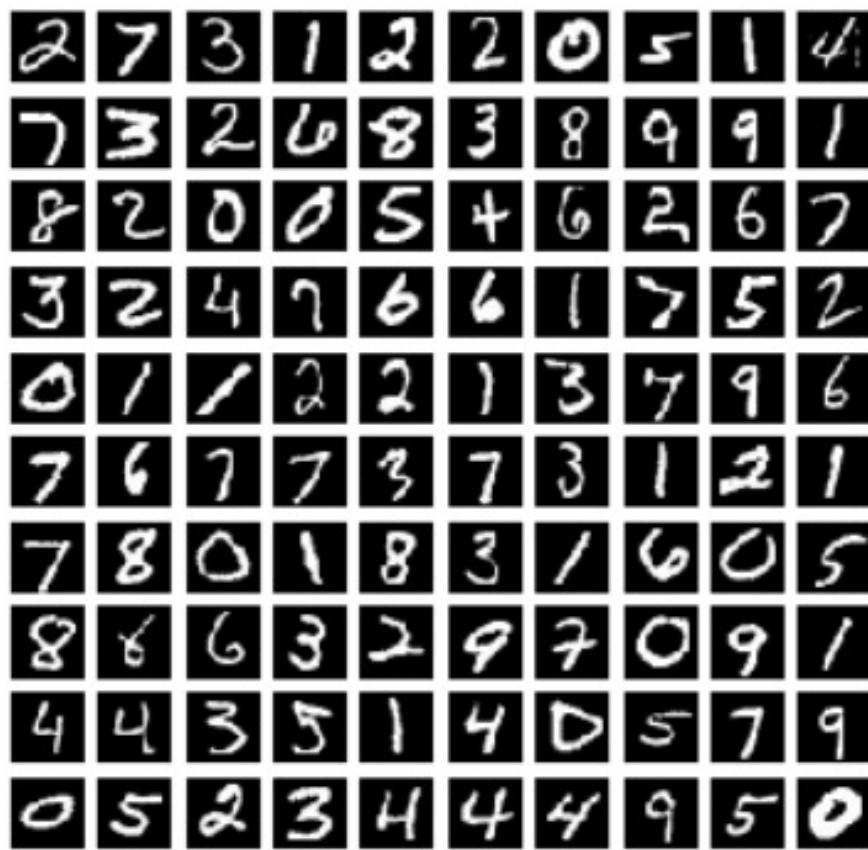
Etichetele corespunzătoare primelor 100 exemple de antrenare

```
[ [4 8 2 7 9 4 2 1 4 5]
  [6 3 1 1 3 0 6 6 7 3]
  [4 4 1 2 2 6 7 4 0 0]
  [5 4 9 0 2 3 2 7 7 9]
  [1 9 1 6 8 7 3 5 9 3]
  [3 7 0 1 5 2 9 8 9 3]
  [2 9 3 5 8 3 6 6 6 3]
  [1 1 6 7 9 2 9 1 1 0]
  [9 1 5 2 9 9 0 0 9 4]
  [6 2 6 8 5 5 4 6 0 7] ]
```

# Prima parte a semestrului: laborator

- setul de date MNIST (cifre scrise de mâna). Setul original conține 50.000 de exemple de antrenare, 10.000 de exemple de testare.

Primele 100 exemple de testare



Etichetele corespunzătoare primelor  
100 exemple de testare

```
[ [2 7 3 1 2 2 0 5 1 4]  
[7 3 2 6 8 3 8 9 9 1]  
[8 2 0 0 5 4 6 2 6 7]  
[3 2 4 7 6 6 1 7 5 2]  
[0 1 1 2 2 1 3 7 9 6]  
[7 6 7 7 3 7 3 1 2 1]  
[7 8 0 1 8 3 1 6 0 5]  
[8 6 6 3 2 9 7 0 9 1]  
[4 4 3 5 1 4 0 5 7 9]  
[0 5 2 3 4 4 4 9 5 0] ]
```

# Proiectul de anul trecut - Clasificarea cuvintelor în funcție de complexitatea lor

- Corpus de propozitii din trei domenii (biblie, biomedicină, dezbatere parlamentul european)
- Scopul proiectului este de a identifica dacă un cuvânt dat dintr-o propoziție este complex sau nu (clasificare binară)
- Complexitatea unui cuvânt într-o propoziție = cât de greu de înțeles este cuvântul în propoziție (regresie -> clasificare binară)

2198	bible	Hezekiah prayed before Yahweh, and said, "Yahweh, the God of Israel, who sit above the cherubim, you are the God, ever	Hezekiah	0
2199	bible	Then Hezekiah said to Isaiah, "The word of Yahweh which you have spoken is good."	Hezekiah	0
2200	bible	Hezekiah said, "They have come from a far country, even from Babylon."	Hezekiah	1
2201	bible	In those days Hezekiah was sick even to death: and he prayed to Yahweh; and he spoke to him, and gave him a sign.	Hezekiah	1
2202	bible	These also are proverbs of Solomon, which the men of Hezekiah king of Judah copied out.	Hezekiah	1
4574	biomed	Moreover, to delineate the brain areas involved in these distinctive olfactory processes we used transgenic "rescue" of Gl	hippocampus	0
4575	biomed	Amyloid burden worsens significantly with age, and by 9 mo, the hippocampus and cortex of untreated mice are largely fi	hippocampus	0
4576	biomed	The contribution of the hippocampus to olfactory memory is presently controversial [2,13–18], but is deemed unlikely for	hippocampus	1
4577	biomed	The subsequent transgene-driven re-introduction of GluR-B, specifically in piriform cortex and hippocampus, reversed the	hippocampus	1
4578	biomed	It is unlikely that morphological changes during development led to impairment of spatial learning and motor coordinatio	hippocampus	1
4579	biomed	The high-mannose core oligosaccharide is added in the ER and is later modified and elaborated in the Golgi apparatus [26	apparatus	0
7397	european	We are pleased to welcome an EFTA delegation to our gallery today, led by Mrs Katrín Júliusdóttir, with members from Ic	Liechtenstein	0
7398	european	Colleagues, I welcome the delegation from the EEA-EFTA Parliaments, that is our colleagues from Iceland, Liechtenstein a	Liechtenstein	0
7399	european	Agreements between the EU, Iceland, Liechtenstein and Norway (Financial Mechanisms 2009-2014 and imports into the E	Liechtenstein	0
7400	european	Accession of Liechtenstein to the Agreement between the EC and Switzerland concerning the establishment of the State r	Liechtenstein	1
7401	european	- Mr President, I would like to add for paragraph 19a the following wording: 'underlines the need to involve and work tog	Liechtenstein	1
7402	european			0

# Date proiect

- Antrenare:
  - 7662 exemple de antrenare organizate într-un fișier excel .xlsx
  - fiecare exemplu de antrenare contine o pereche (propozitie, cuvânt) și o eticheta = clasa 0 sau clasa 1 în funcție de complexitatea cuvântului în propoziție (cât de ușor/greu poate fi înțeles la nivel semantic acel cuvânt)
  - clasa 0: cuvântul nu este complex
  - clasa 1: cuvântul este complex
- Testare:
  - 1338 exemple de testare organizate într-un fișier excel .xlsx
  - pentru fiecare exemplu de testare de forma (propozitie, cuvânt) trebuie să preziceți eticheta binară
  - public test (40%) vs private test (60%)

# Exemple de testare

1303	bible	Does he keep turning the soil and breaking the clods?	clods
1304	biomed	During skeletal development, the anabolic activity of osteoblasts [1] is favored over the catabolic activity of osteoclasts [1]	mass
1305	biomed	At 1 mo of age, the mice lacked visible amyloid aggregates that might act as an intractable reservoir of peptide remainin peptide	
1306	biomed	All experiments were approved by the Institutional Animal Care and Use Committee of Kansas State University.	Care
1307	europarl	This Convention, adopted in June 2007 by the 96th session of the International Labour Conference, is aimed at establish	Convention
1308	biomed	All studies involving animals were reviewed and approved by the University of Pittsburgh's Institutional Animal Care and Animal	
1309	biomed	Lateral inhibitory circuits were postulated, in analogy to retina [34,35], to mediate contrast enhancement [24], for which retina	
1310	bible	Then Eliakim, Shebna and Joah said to Rabshakeh, "Please speak to your servants in Aramaic, for we understand it; and language	
1311	bible	For Herod had laid hold of John, and bound him, and put him in prison for the sake of Herodias, his brother Philip's wife. Philip	
1312	bible	They are burning like an oven that the baker stops stirring, from the kneading of the dough, until it is leavened.	kneading
1313	biomed	In the absence of a dominant (wild-type [wt]) allele, genetic interactions between recessive alleles (referred to here as " interactions	
1314	europarl	In recent months the Presidency, together with the Commission, has invested a lot of effort at the highest level in trying	level
1315	bible	This I say for your own profit; not that I may ensnare you, but for that which is appropriate, and that you may attend to distraction	
1316	europarl	As you know, the Committee on Budgets voted on more than 1 600 amendments to the 2008 draft budget.	Budgets
1317	biomed	These strains represent important resources and provide the genetic resource to positionally clone numerous quantitati	resources
1318	europarl	In the meantime, we have arrived at this shift via various political approaches and instruments.	meantime
1319	biomed	Thin layers of articular cartilage line the bones of synovial joints and provide a smooth, wear-resistant structure that red bones	
1320	europarl	The charging of heavy goods vehicles for the use of certain infrastructures - Greening of transport and internalisation of continuation	
1321	biomed	These results were thought to support a direct role for annexin A7 in surfactant secretion, but in these studies annexin A secretion	

# Proiectul de acum doi ani

- Tweet-uri în limba italiana
- Scopul proiectului este de a identifica tweet-urile misogine

id	text	label
1	@KassemAmin4 @Laylasexdgr Fatti trovare te lo do volentieri e ti sborro in bocca	1
2	@meb Tu dovresti ricominciare dai semafori a fare la lavavetri..ma tranquilla tanto il vitalizio ti resterà in gola	1
3	Amore,sei presentabile? XchÃ“ cosÃ– via Skype ti faccio conoscere i miei parenti..." e fu cosÃ– che da un tablet ho conosciuto i parenti terroni della mia tettona,pr	1
4	@Il_nulla Salvo poi mandare la culona a Mosca, aummaumm, per farsi dare da Putin commesse multimiliardarie e metterlo in quel posto ai FRATELLI della UE (tra	1
5	@GiorgiaMeloni @FratellidItalia Vediamo Gentiloni, Ã“ ora di finirla di essere servili! Pensiamo al danno che la Francia ci ha fatto con Sarkozy e la culona della Mel	1
6	@alinefenaoly Che cagna che sei anche io voglio fare pasqua così con te mmmmmm	1
7	@Alex82564331 Bella figura da scopare	1
8	Puttana â€” ma che cazzo ne sai tu ma la smetti <a href="https://t.co/BNpkOWXXQ0">https://t.co/BNpkOWXXQ0</a>	1
9	@elheartmash Ti inculo <a href="https://t.co/Bs92ZWOYr5">https://t.co/Bs92ZWOYr5</a>	1
10	Finalmente di nuovo con la mia polpettona culona coccodrilloso â»». <a href="https://t.co/HhVIJySILc">https://t.co/HhVIJySILc</a>	1
11	@ValeSantaSubito che puttanone	1
12	@ficamiatop Ti scopo la figura	1
13	Mi Ã“ piaciuto un video di @YouTube: <a href="https://t.co/hqmB5QNeN1">https://t.co/hqmB5QNeN1</a> 20 MOTIVI PER USCIRE CON UNA TETTONA	1
14	@SailorNuzz Ah tettona	1
15	@pheiticeira @SailorNuzz Hai visto suoi stories ?la tettona	1
16	@leone5264 @TwittGiorgio @ilgiornale Abbiamo giÃ avuto MONTI,, ed abbiamo visto cosa ha fatto „ assieme alla Fornero,, servi dell' Europa ,leccaculi della culo	1
17	Comunque Chiara Ferragni tettona sicuramente fra i regali piÃ¹ belli di questo 2018.	1
18	@sdeangelis56 Il problema Ã“ sorto da quando il premier francese fece assassinare Gheddafi in combutta con il presidente USA e la culona tedesca.	1
19	@marica_love23 @MattleyR Marica seguimi ti scopo io troia	1
20	Zitta puttana ti metto i coglioni in gola <a href="https://t.co/lXMZmVfB7V">https://t.co/lXMZmVfB7V</a>	1
21	#Isola sto rivedendo replica dell'isola. Elena schifosa lurida, ma chiudi la boccaccia Cattiva perfida	1
22	@sexyforyou74 Puttanona Adorata ??	1
23	@erisdelperchÃ© sei una figura di legno, giusto?	1
24	Tremenda culona rica.. @Esposos33 <a href="https://t.co/yTj3zRs9mA">https://t.co/yTj3zRs9mA</a>	1
25	@Melissa_Merz Che tettona?	1

# Corpus de tweet-uri în limba italiană

- Tweet-uri în limba italiana
- Scopul proiectului este de a identifica tweet-urile misogine

2200	Mauro ti voglio bene lo stesso ma PORCA TROIA CHE CAZZO HAI FATTO? #MilanInter	0
2201	Non avere una sola punta degna di tale nome pensa, porca puttana se pesa	0
2202	Adesso continuo a stare con l'ansia addosso porca puttana sono stanca	0
2203	raga porca puttana che casino che Ã“ successo, che porca merda	0
2204	giuro non so che mi succede sento la gente parlare e vorrei far stare tutti zitti Ã“ una cosa generale non sopporto le voci porca puttana sta andando avanti da giorni	0
2205	@acmilan Porca puttana avete piÃ¹ culo voi che la Kardashian!!! âš«T, @?????	0
2206	Assurdo, fino a trenta secondi dalla fine ero sereno, contento del fatto che comunque avevamo giocato una buona partita e che la tanto temuta fregatura nel recupero	0
2207	Ho aspettato 80minuti i due gol del Milan che avevo sentito in radio. Nn so cosa cazzo avessi sentito ma porca puttana maiali quanto girano i coglioni	0
2208	Piccinini che sfigato, il pistolero, il matador missile terra aria, figa ma come cazzo Ã“ messo#BarcellonaRoma	0
2209	Ma porca puttana, quello era rigore#BarcaRoma	0
2210	Ma vogliamo parlare di quanto sia di cattivo gusto il cointatore dei femminicidi al TG 2? Il contatore? Ma che cazzo Ã“ porca puttana	0
2211	Passo sul 20. Questi schifosi di Premium non la trasmettono la Roma". Cocco streaming. Si bloccano. Passo su Telelombardia per 10 minuti, pubblicitÃ . Zapping ca	0
2212	Era vinta sta partita porca puttana..li abbiamo massacrati sti gonzi di merda. Che rabbia.	0
2213	oi raga figa mi sono rotta la minchia di essere sempre la cazzo di sfigata di merda della situazione. indovinate chi domani non vedrÃ succede? IO PORCA PUTTANA	0
2214	FILIPPO PORCA PUTTANA DUE ACE PROPRIO OGGI NO	0
2215	Ã“ dannatamente bello porca puttana #harrystylesliveontourbologna	0
2216	raga lavoro da due (2) giorni e sono giÃ indietro con le serie tv porca puttana che schifo il lavoro	0
2217	c'hai ragione alessÃ , se vive na vorta sola, e porca puttana tra tutti noi eri forse l'unico che nse meritava de lasciÃ sto mondo cosÃ– presto, te che c'avevi ottomila	0
2218	col cazzo che gli scrivo ahaha ceh va bene ho fatto la stronza ma lui ancora di piÃ¹ stronzo troione da combattimento daddy baldracco puttanone zoccolo da corsa	0
2219	Gli dite a De rossi gioca con la Roma no col Barcellona porca puttana !!!!!!!	0
2220	@AlbertoD1998 Stessa cosa di ieri, porca troia anche voi che cazzo vuol dire farla a quellâ€™ora ahahahahah	0
2221	Rossa tu nn ci azzacchi nulla" ti sei ritrovata in mezzoCome alcuni de loro Ah "inculo alla balena"(So che leggete )"	0
2222	E voi @OfficialASRoma porca puttana dovete protestare cazzo! Quando succedono ste cose si va in 10 intorno a quella merda con i giacchetta gialla	0
2223	@filippothecap Ma come cazzo fai a non applaudire un gol del genere.... Ã“ normale! Ã“ spontaneo.....porca puttana	0
2224	E allora non sorridere, porca puttana.	0
2225	porca puttana mi manca Jonghyun	0

# Platforma Kaggle

- pentru participare la proiect
  - vă faceți cont pe platforma [www.kaggle.com](http://www.kaggle.com)
  - puteți participa numai pe bază de invitație accesând link-ul de mai jos:
    - <https://www.kaggle.com/t/3dfa2c8369e548d1afa7a7769590284c>
    - <https://www.kaggle.com/c/ub-fmi-cti-2021-2022/>
  - dați Join Competition și apoi puteți participa încărcând predicțiile voastre.
- 2 submisii/zi (sub forma unui fișier CSV – demo)
- puteți să vă alegeti la final cele mai bune două submisii pe care le considerați voi
- public test (40%) vs. private test (60%)
- termen limită duminică, 21 noiembrie, 23:59 UTC time
- prezentare proiecte în săptămâna 22-26 noiembrie

# Platforma Kaggle

The screenshot shows a web browser displaying a Kaggle competition page. The URL in the address bar is [kaggle.com/c/ub-fmi-cti-2021-2022](https://kaggle.com/c/ub-fmi-cti-2021-2022). The left sidebar contains navigation links: Create, Home, Competitions, Datasets, Code, Discussions, Courses, and More. Under Recently Viewed, there are links to 'Complex Word Classifi...', 'Chess Positions', 'random predictions', and 'brain tumor prediction'. The main content area shows the 'InClass Prediction Competition' for the challenge 'Complex Word Classification'. The challenge description states: 'The task of this challenge is to automatically classify whether or not a word in a given sentence is complex.' It shows 17 teams and 17 days remaining. The navigation bar includes Overview, Data, Code, Discussion, Leaderboard, Rules, Team, Host, My Submissions, Submit Predictions (which is highlighted in black), and more options. Below the navigation is an 'Overview' section with tabs for Description, Evaluation, and Add Page. The Description tab contains text about predicting complex words for various populations.

InClass Prediction Competition

## Complex Word Classification

The task of this challenge is to automatically classify whether or not a word in a given sentence is complex.

17 teams · 17 days to go

Overview Data Code Discussion Leaderboard Rules Team Host My Submissions **Submit Predictions** ...

Overview Edit

Description	Predicting which words are considered hard to understand for a given target population is a vital step in many Natural Language Processing applications such as text simplification. A system could simplify texts for second language learners, native speakers with low literacy levels, and people with reading disabilities. This task is commonly referred to as Complex Word Identification. Usually, this task is approached as a binary classification task in which systems predict a complexity value (complex vs. non-complex) for a set of target words in a text. In this challenge, the task is to predict the lexical complexity of a word in a sentence. A word which is considered to be complex has label 1, a word is considered to be simple (non-complex) has label 0.
Evaluation	+ Add Page

# Clasamentul final de anul trecut

#	△	Team	Members	Score	Entries	Last	Code
1	▲ 8	Alexandru-Theodor Ionescu		0.84109	42	10mo	
2	▲ 4	Mădălina Vînaga		0.83835	30	10mo	
3	▲ 10	Nicolescu Robert		0.83835	13	10mo	
4	▼ 2	Denisa Ionascu		0.83219	11	10mo	
5	▲ 9	Popescu Mircea-Mihail		0.82876	40	10mo	
6	▲ 6	TIBERIU MAXIM		0.82808	19	10mo	
7	▲ 4	Alex Serban		0.82808	16	10mo	
8	▼ 5	Andreea Stanciu		0.82534	16	10mo	
9	▲ 7	Dragos Balmau		0.82534	22	10mo	
10	▼ 9	Ursu-Balaci Daniel		0.82054	6	10mo	
11	▼ 6	ALEXANDRU SURDU-BOB		0.81095	9	10mo	

# Platforma Kaggle

← → C kaggle.com/c/ub-fmi-cti-2020-2021

**kaggle**

Search

InClass Prediction Competition

## Automatic Misogyny Identification

The task of this challenge is to automatically identify misogynous content in Twitter post in italian language.

21 days to go

Overview Data Notebooks Discussion Leaderboard Rules Join Competition

Recently Viewed

Automatic Misogyny Id...

Overview

Description	Evaluation
Given the huge amount of user-generated contents on the Web, and in particular on social media, the problem of detecting, and therefore possibly limit the diffusion of hate speech agains women, is rapidly becoming a fundamental problem especially for the societal impact of the phenomenon. In this challenge competitors have to train a classifier to predict if a a twitter written in the italian language has a mysogynous content. The training data consists of a collection of 5000 training examples (in a .csv file). These examples contain selected twitter posts from a large database. Each training example is labeled with 0 or 1. Training examples labeled with label 1 contain text that can by considered as hate speech agains women. The goal is to predict the label for the 1000 testing examples.	

# Clasamentul final de acum doi ani

Overview Data Code Discussion Leaderboard Rules Team Host My Submissions Late Submission ...

[Public Leaderboard](#)

[Private Leaderboard](#)

The private leaderboard is calculated with approximately 60% of the test data.

 Refresh

This competition has completed. This leaderboard reflects the final standings.

#	△pub	Team Name	Notebook	Team Members	Score ⓘ	Entries	Last
1	—	LaurentiuMarianIvan			0.83333	5	10mo
2	▲ 3	Albastroiu Dragos			0.81583	33	10mo
3	▲ 9	Petcu Cristiana Diana			0.79929	30	10mo
4	▲ 9	Melissa Unal			0.79929	14	10mo
5	▲ 16	Florentina Tache			0.79859	21	10mo
6	▲ 8	Teodor Daniel Mocanu			0.79310	6	1y
7	▲ 8	ANDRADA-GABRIELA DUMIT...			0.79310	16	10mo
8	▲ 15	Stefan Teodorescu			0.79113	10	10mo
9	▲ 23	TUDOR-ANDREI GAVRILITA			0.78955	18	10mo

# A doua parte a semestrului: curs

1. Rezolvarea problemelor prin căutare

Stare

Scop

Funcție succesor

Acțiuni

Spațiul stărilor

Arbore de căutare

Căutare neinformată

Căutare informată

2. Strategii de căutare neinformate

Căutare în lățime

Căutare în adâncime

Căutare cu cost uniform

Căutare cu adâncime limitată

Căutare cu adâncime incrementală

3. Strategii de căutare informate

Euristică

Greedy

A\*

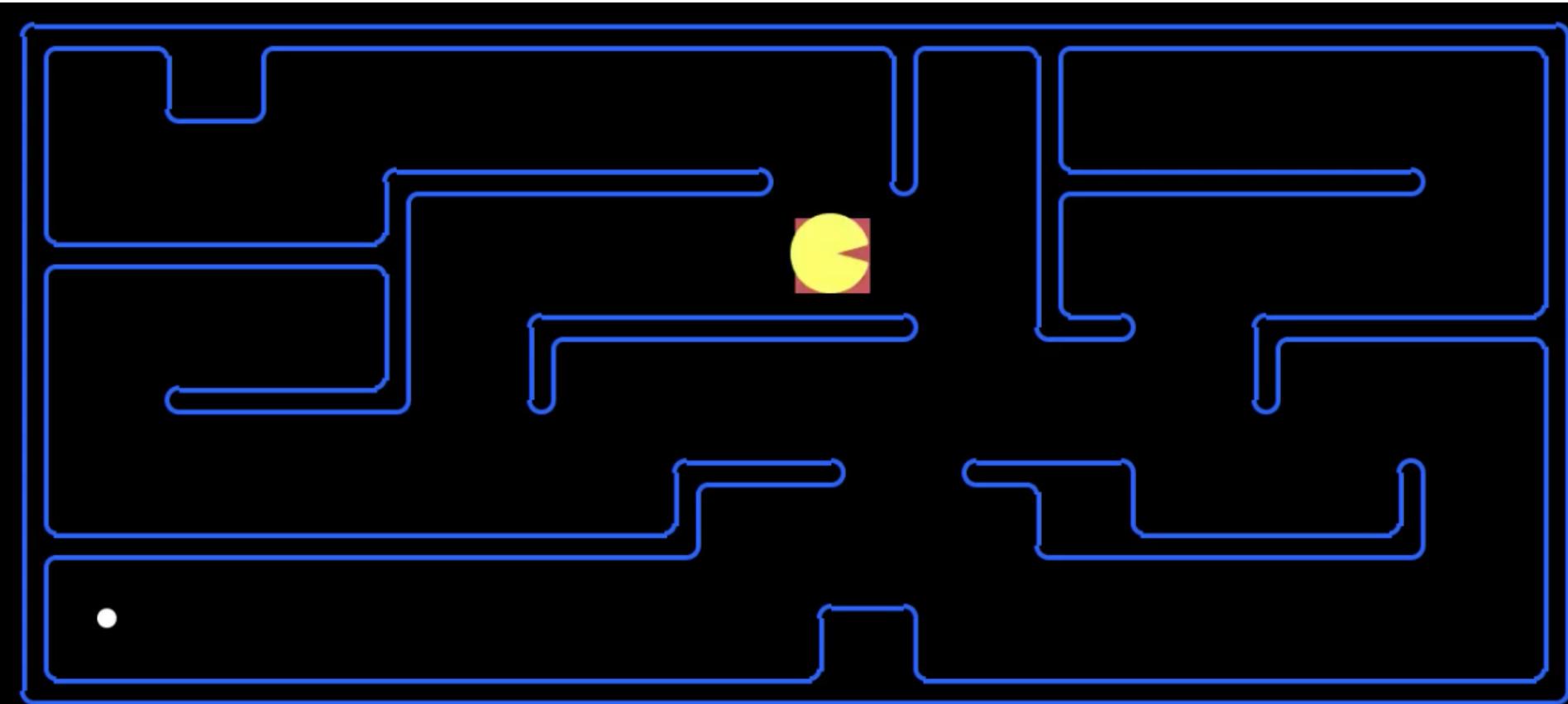
Euristică admisibilă

4. Strategii de căutare adversariale

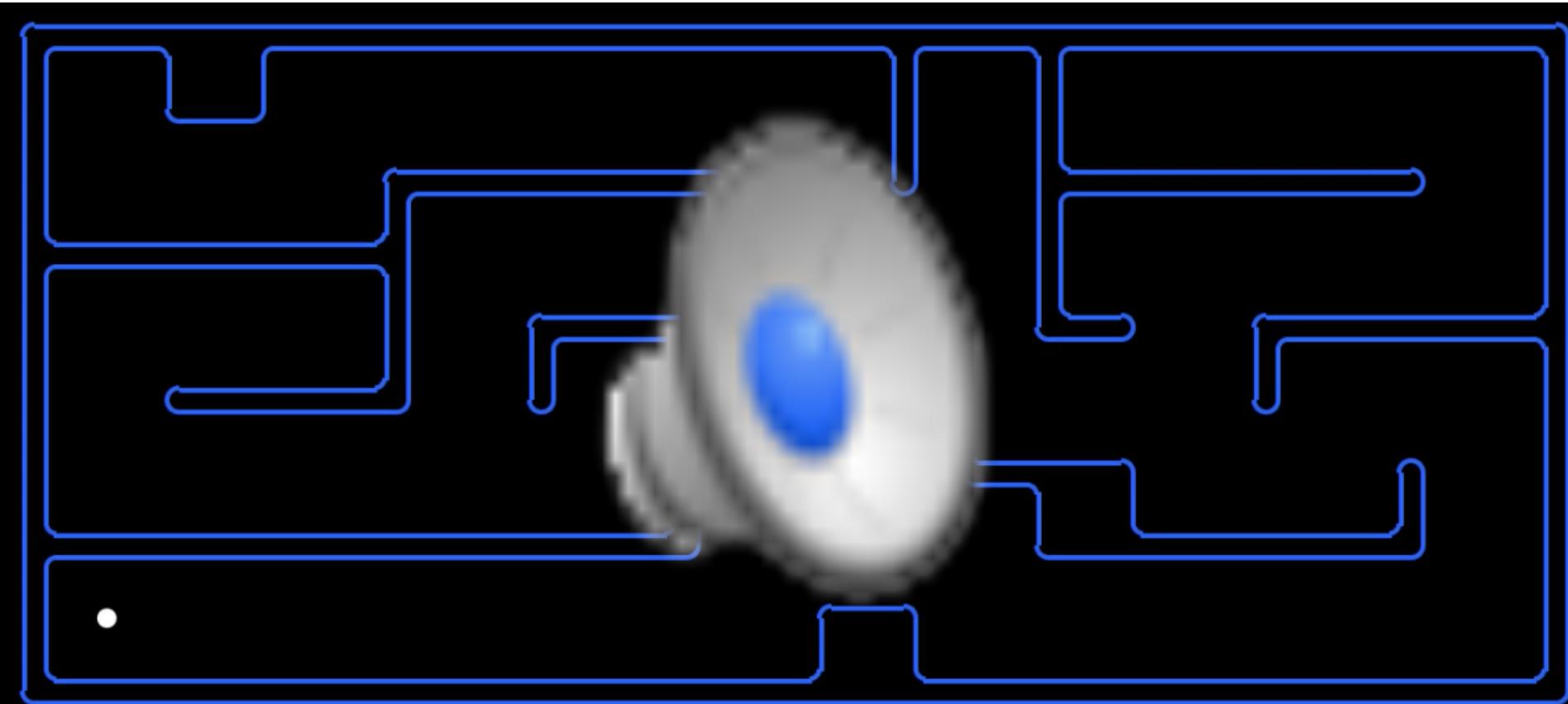
Algoritmul minimax

Alpha-Beta retezare

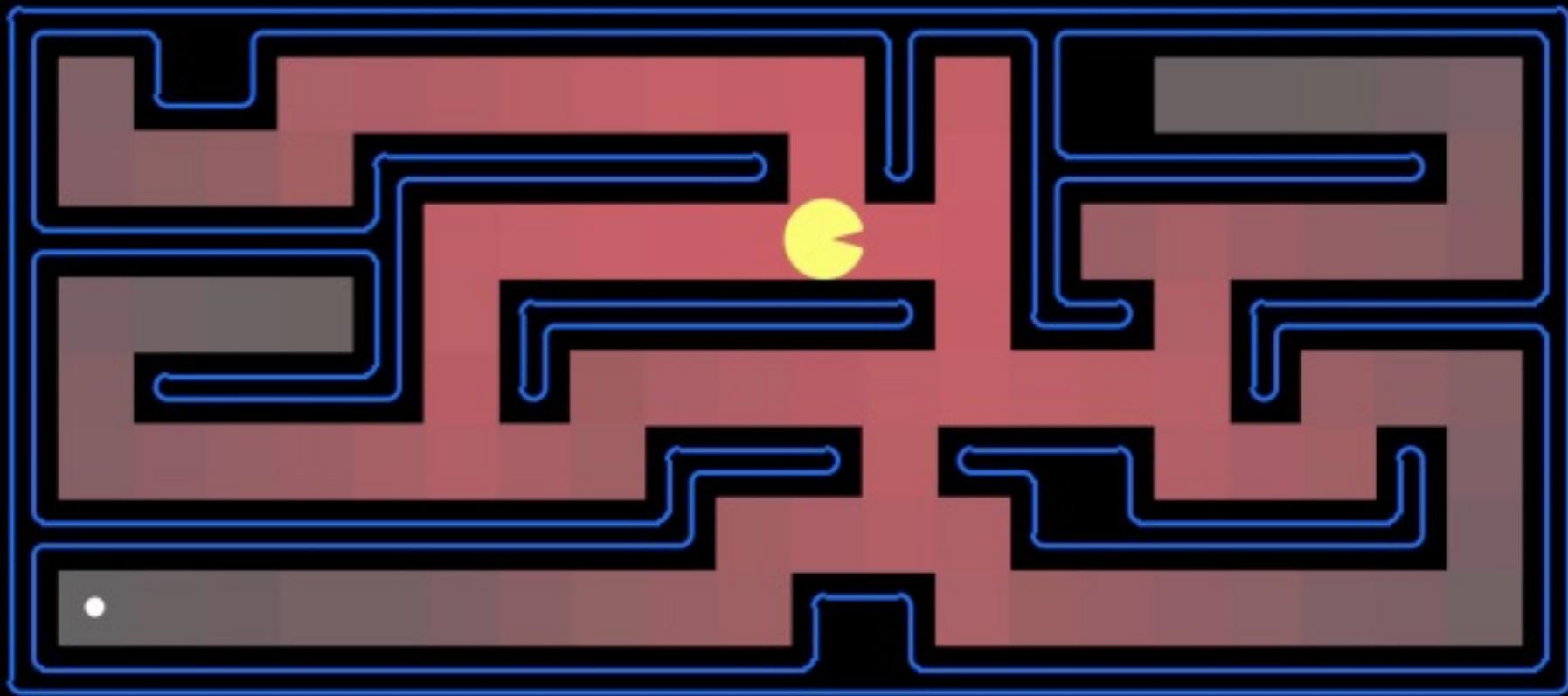
# Explorarea arborelui de căutare



# Explorarea arborelui de căutare



# Explorarea arborelui de căutare



**SCORE: 0**

Ordinea de explorare a nodurile este dată de culoare:

**Roșu intens – noduri exploreate la început**

**Gri – noduri exploreate la sfârșit**

# Problema 8-puzzle

## Descrierea problemei

Pe o tablă  $3 \times 3$  se găsesc 8 piese, numerotate de la 1 la 8. La fiecare moment o singură piesă se poate mișca cu o poziție, pe orizontală sau verticală, în limitele cadrului tablei, în locul rămas liber (poziția roșie). Se dă o configurație inițială și una finală a tablei. Trebuie să se găsească secvența de mutări care să aducă piesele din configurația inițială în cea finală.

2		3
1	8	4
7	6	5

Stare inițială

[2, X, 3, 1, 8, 4, 7, 6, 5]



1	2	3
8		4
7	6	5

Stare scop

[1, 2, 3, 8, X, 4, 7, 6, 5]

# Formularea problemei 8-puzzle

- Spatiul stărilor

- stare = configurație a pătratului  $3 \times 3$
- $9!/2$  stări posibile
- stare inițială
- stare scop

2		3
1	8	4
7	6	5

Stare inițială

1	2	3
8		4
7	6	5

Stare scop

[2, X, 3, 1, 8, 4, 7, 6, 5]

[1, 2, 3, 8, X, 4, 7, 6, 5]

- Acțiuni

- STÂNGA, JOS, DREAPTA, SUS
- fiecare acțiune (mutare) are cost = 1

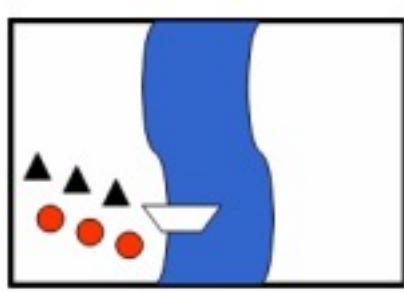
- Funcția succesor

- SUCCESOR([X, a, b , c, d, e, f ,g , h], STANGA) = NU EXISTA
- SUCCESOR([X, a, b , c, d, e, f ,g , h], SUS) = NU EXISTA
- SUCCESOR([X, a, b , c, d, e, f ,g , h], DREAPTA) = [a, X, b , c, d, e, f ,g , h]
- SUCCESOR([X, a, b , c, d, e, f ,g , h], JOS) = [c, a, b , X, d, e, f ,g , h]
- ...

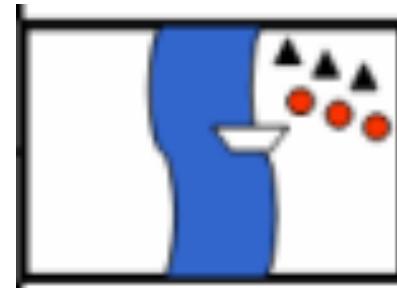
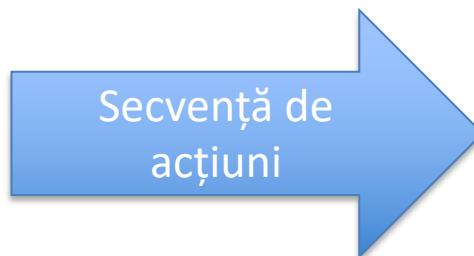
# Problema misionarilor și a canibalilor

## Descrierea problemei

Trei misionari și **trei canibali** se află la marginea unui râu, cu scopul de a trece pe celălalt mal. Ei au la dispoziție o barcă de două persoane. Dacă la un moment dat, pe un mal sau pe celălalt, numărul canibalilor este mai mare decât cel al misionarilor, canibalii îi vor mâncă pe misionari. Problema constă în a afla cum pot trece cele 6 persoane în deplină siguranță de pe un mal pe celălalt ☺. Barca nu merge singură, este nevoie de cel puțin o persoană în barcă.



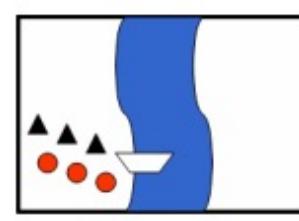
Stare inițială



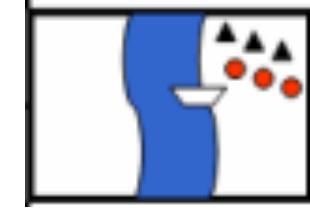
Stare scop

# Formularea problemei misionarilor și a canibalilor

- Spațiul stărilor
  - triplet  $(m, c, b)$ :  $m$  = numărul de misionari pe malul stâng,  $c$  = numărul de canibali pe malul stâng,  $b$  = prezența bărcii (0 sau 1) pe malul stâng
  - stare inițială:  $(3,3,1)$
  - stare scop:  $(0,0,0)$
- Acțiuni
  - pot duce de pe un mal către celălalt mal 1 sau 2 misionari, 1 sau 2 canibali, 1 misionar + 1 canibal ( $1m, 2m, 1c, 2c, 1m+1c$ )
- Funcția succesor
  - $\text{Succesor}((m, c, b), 1c) = (m, c + 1 - 2b, 1 - b), \dots$
  - $\text{Succesor}((m, c, b), 1m) = (m + 1 - 2b, c, 1 - b), \dots$
  - unele stări sunt invalide  $(2,3,0)$  sau ilegale  $(0,0,1)$

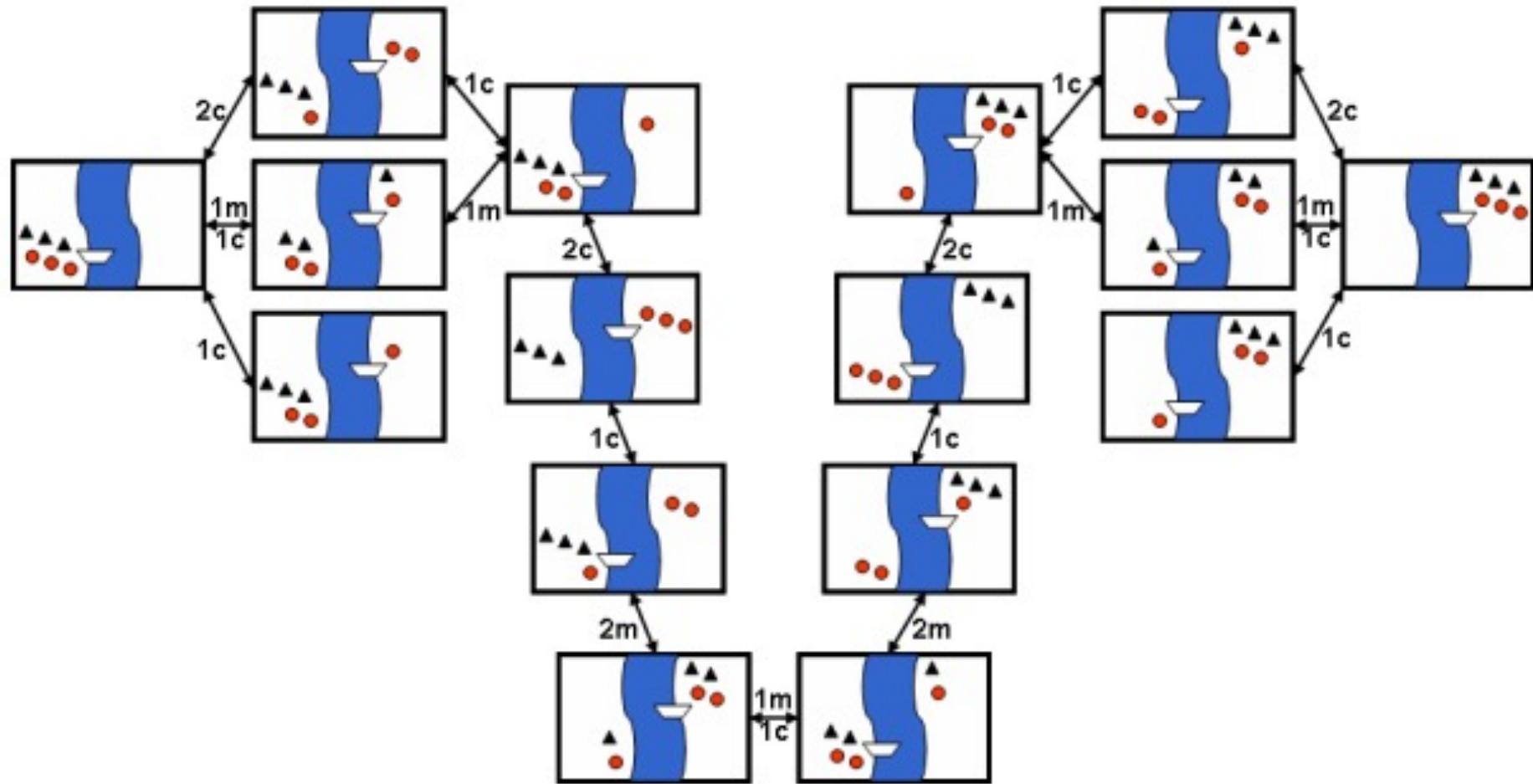


Stare inițială

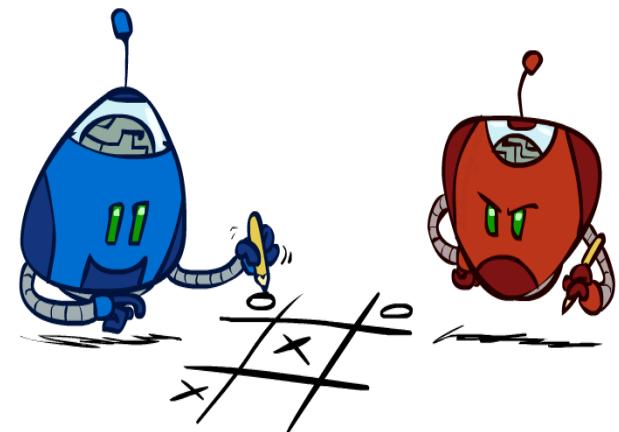
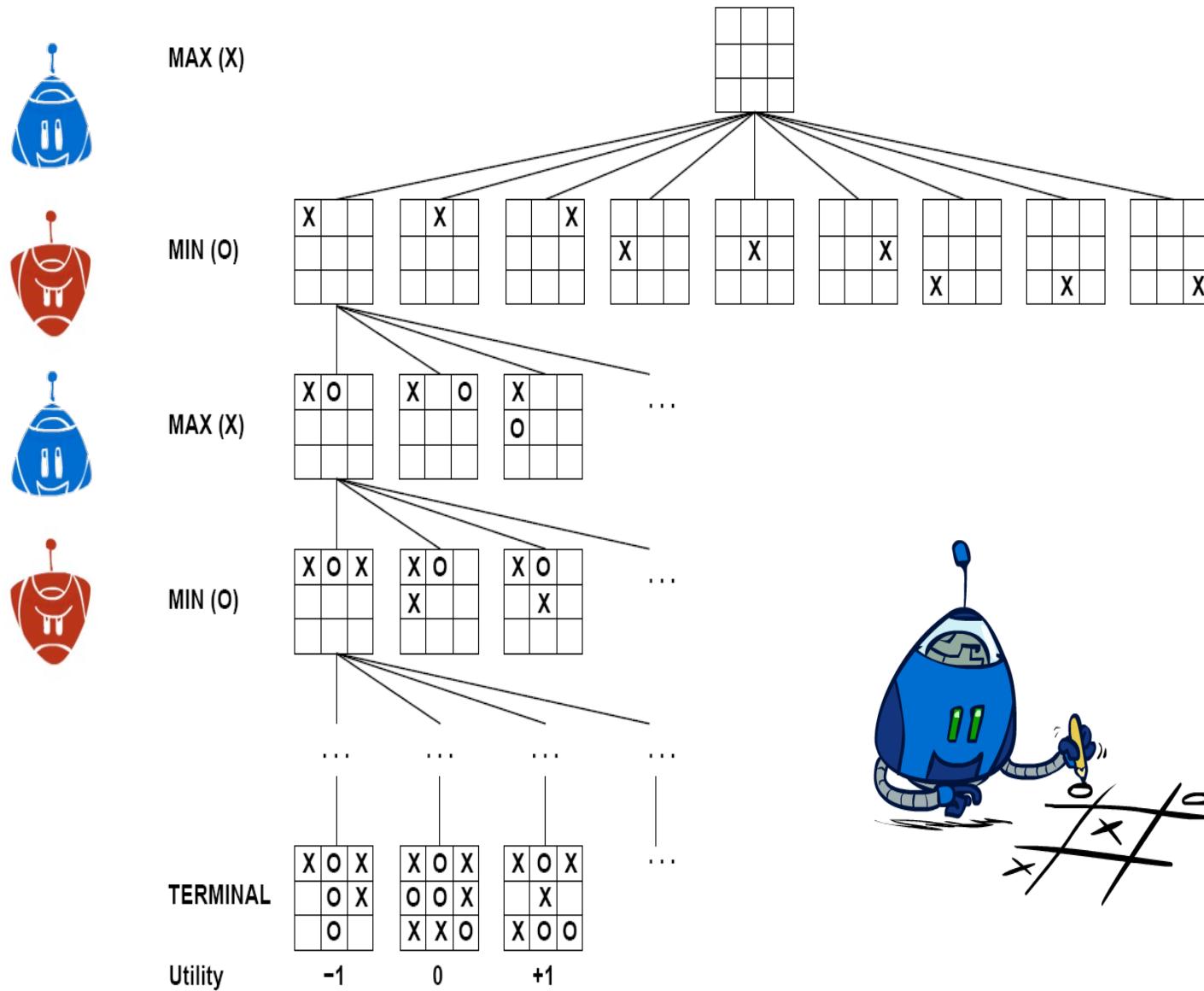


Stare scop

# Graful de stări al problemei misionarilor și a canibalilor

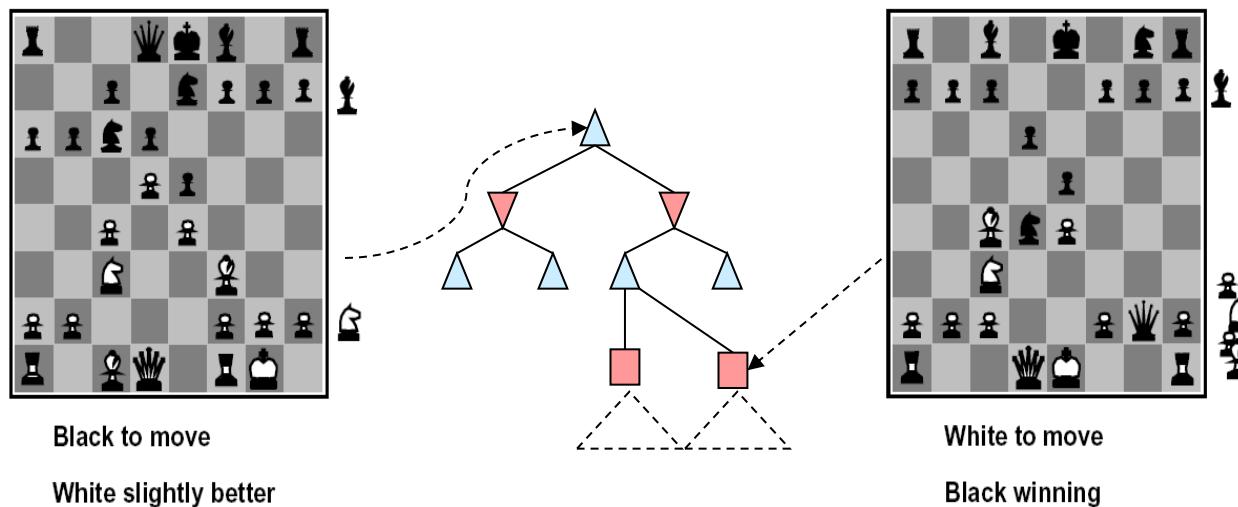


# Arbore de joc pentru X și 0



# Funcții de evaluare - șah

- Funcțiile de evaluare asociază un scor stărilor neterminale (este folosită de căutarea în adâncime limitată)



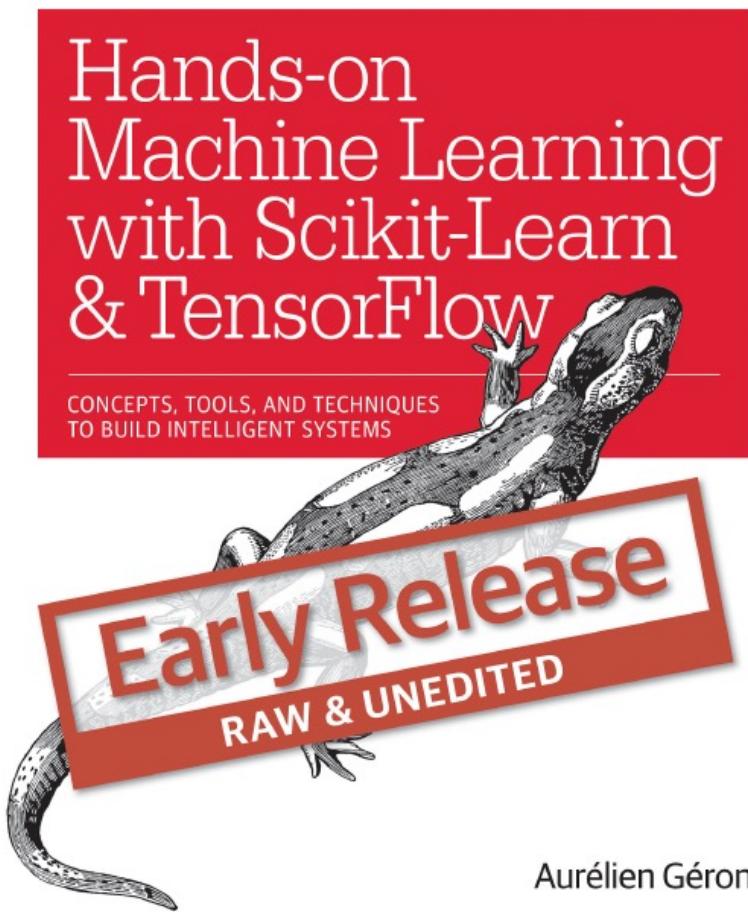
- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc: pentru șah se consideră funcții liniare în care ponderăm piesele de pe tablă

$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s)$$

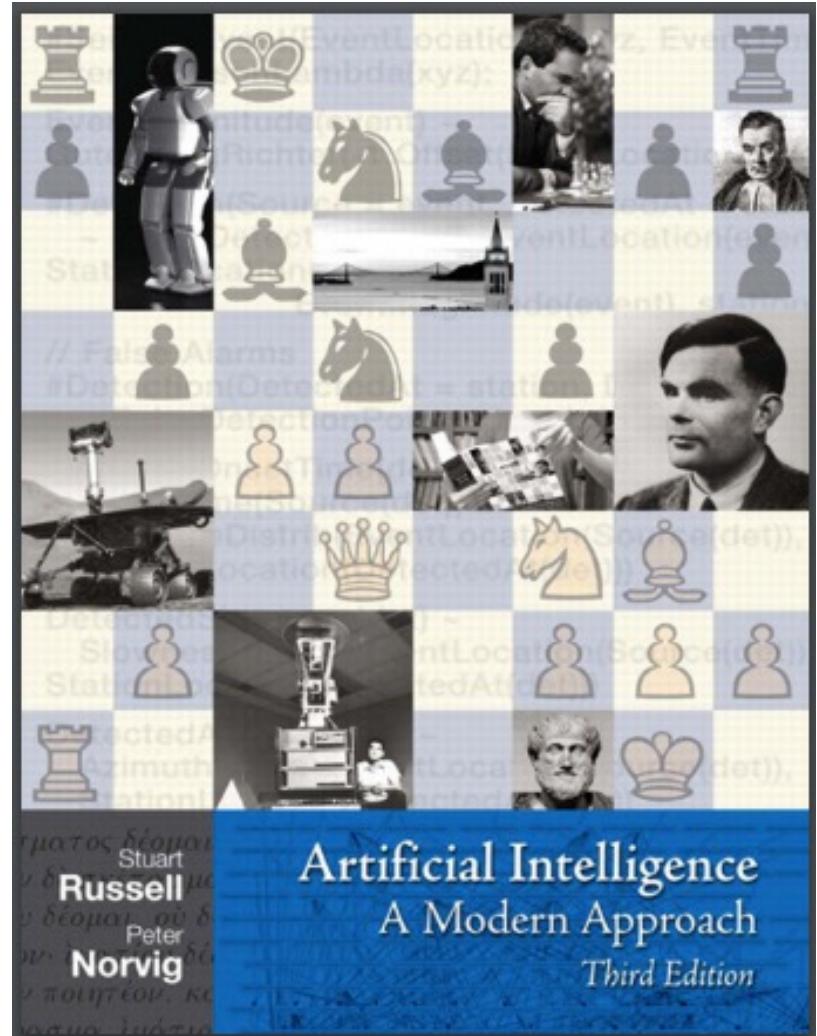
- unde,  $f_1(s) = (\#\text{regine\_albe} - \#\text{regine\_negre})$ ,  $w_1 = 100$ , etc.

# Bibliografie

O'REILLY®

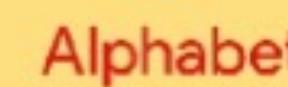
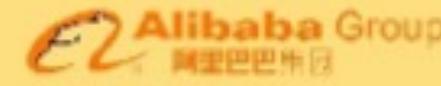


Prima parte



Prima și a doua parte

# **Aplicații de succes în Inteligenta Artificială**

	Q1 2009	Q1 2014	Q1 2019
#1	<b>ExxonMobil</b>		 Microsoft
#2	<b>PetroChina</b>		
#3	<b>Walmart</b> 	 Microsoft	 amazon
#4	<b>ICBC</b> 		
#5	 中国移动 China Mobile	<b>BERKSHIRE HATHAWAY INC.</b>	<b>BERKSHIRE HATHAWAY INC.</b>
#6	 Microsoft		
#7	 AT&T		
#8		 	
#9			
#10			
USS	1,900,000,000,000	3,000,000,000,000	6,000,000,000,000

Top 10 Publicly Traded Companies by Market Cap (FT Global 500)

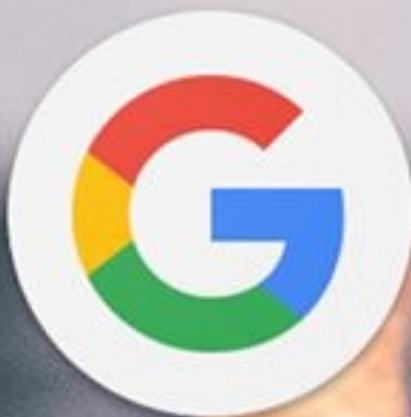


amazon  
alexa

Alexa



Cortana

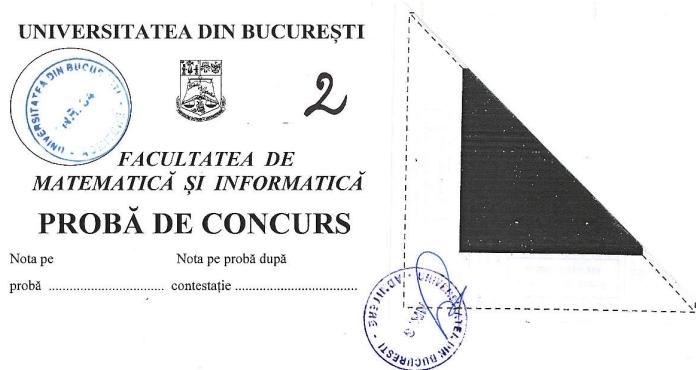


Google



Siri

# Corectarea automată a testelor grilă

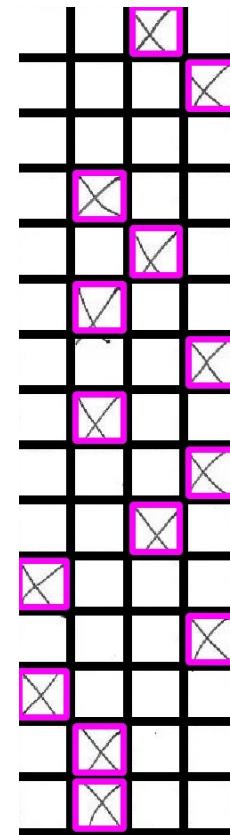


**TEST GRILĂ**

**INFORMATICĂ**  1      **FIZICĂ**  \_\_\_\_\_  
Număr varianță

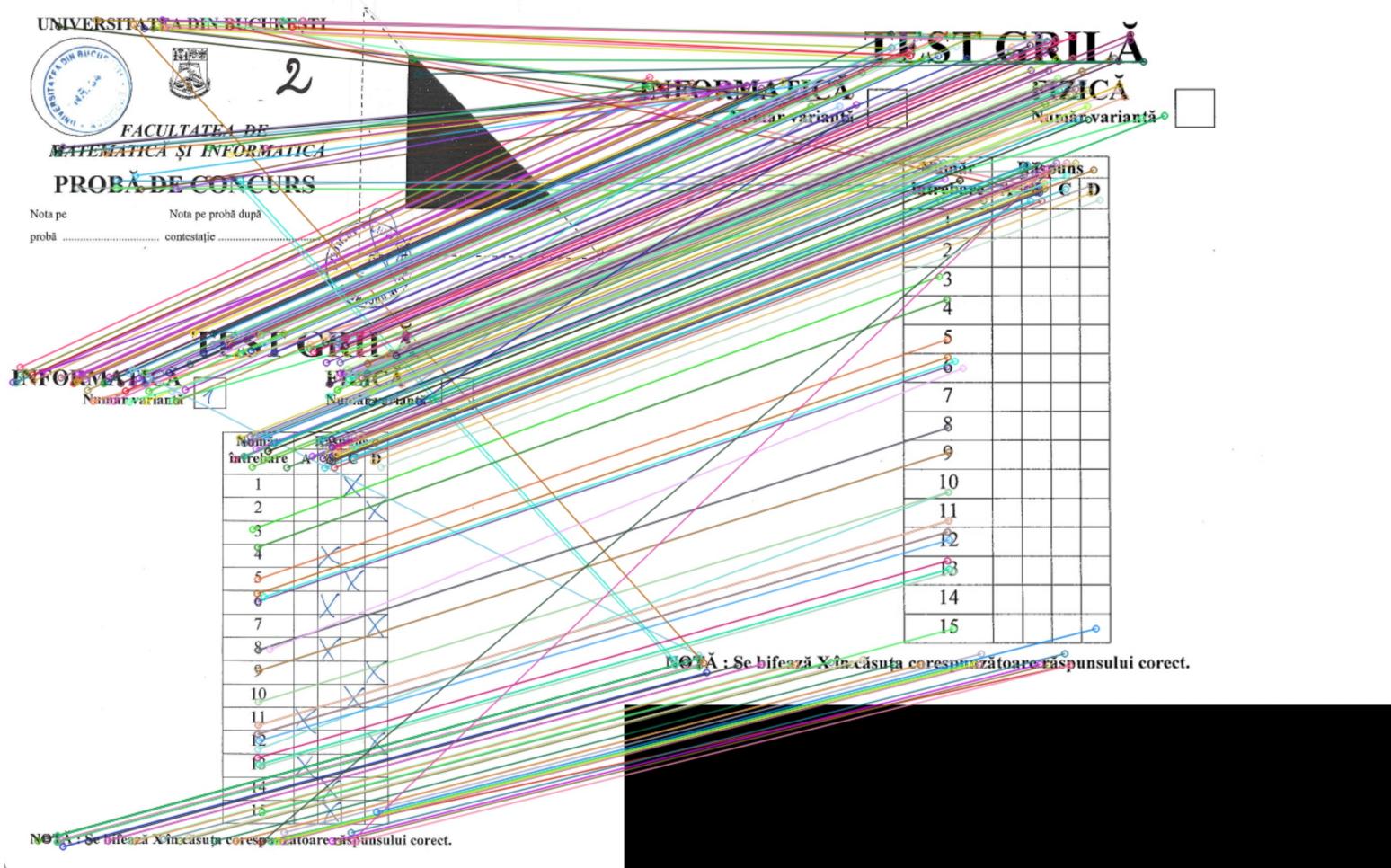
Număr întrebare	A	B	C	D
1			X	
2			X	
3				
4	X			
5		X		
6	X			
7		X		
8	X			
9			X	
10		X		
11	X			
12			X	
13	X			
14		X		
15		X		

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.



Numar Lucrare	Varianta	0X	1X	mX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Numar grile corecte	Nota
2	Informatica_1	1	14	0	C	D	B	C	B	D	B	D	C	A	D	A	B	B		2	2.2

# Corectarea automată a testelor grilă



# IA pentru jocuri

- antrenate pentru a găsi cele mai bune strategii de joc
- paradigma învățării pe bază de recompensă (reinforcement learning)
  - recompensă +/- pentru câștigarea/pierderea unui joc

# Şah: Deep Blue vs Kasparov

- Deep Blue (proiectat de IBM) îl învinge în 1997 pe Kasparov după ce în prealabil pierduse în 1996

- **1996: Kasparov câştigă**

“I could feel – I could smell  
kind of intelligence across 1

The 1996 match				
Game #	White	Black	Result	Comment
1	Deep Blue	Kasparov	1–0	
2	Kasparov	Deep Blue	1–0	
3	Deep Blue	Kasparov	½–½	Draw by mutual agreement
4	Kasparov	Deep Blue	½–½	Draw by mutual agreement
5	Deep Blue	Kasparov	0–1	Kasparov offered a draw after the 23rd move.
6	Kasparov	Deep Blue	1–0	

*Result: Kasparov–Deep Blue: 4–2*

- **1997: Deep Blue câştigă**

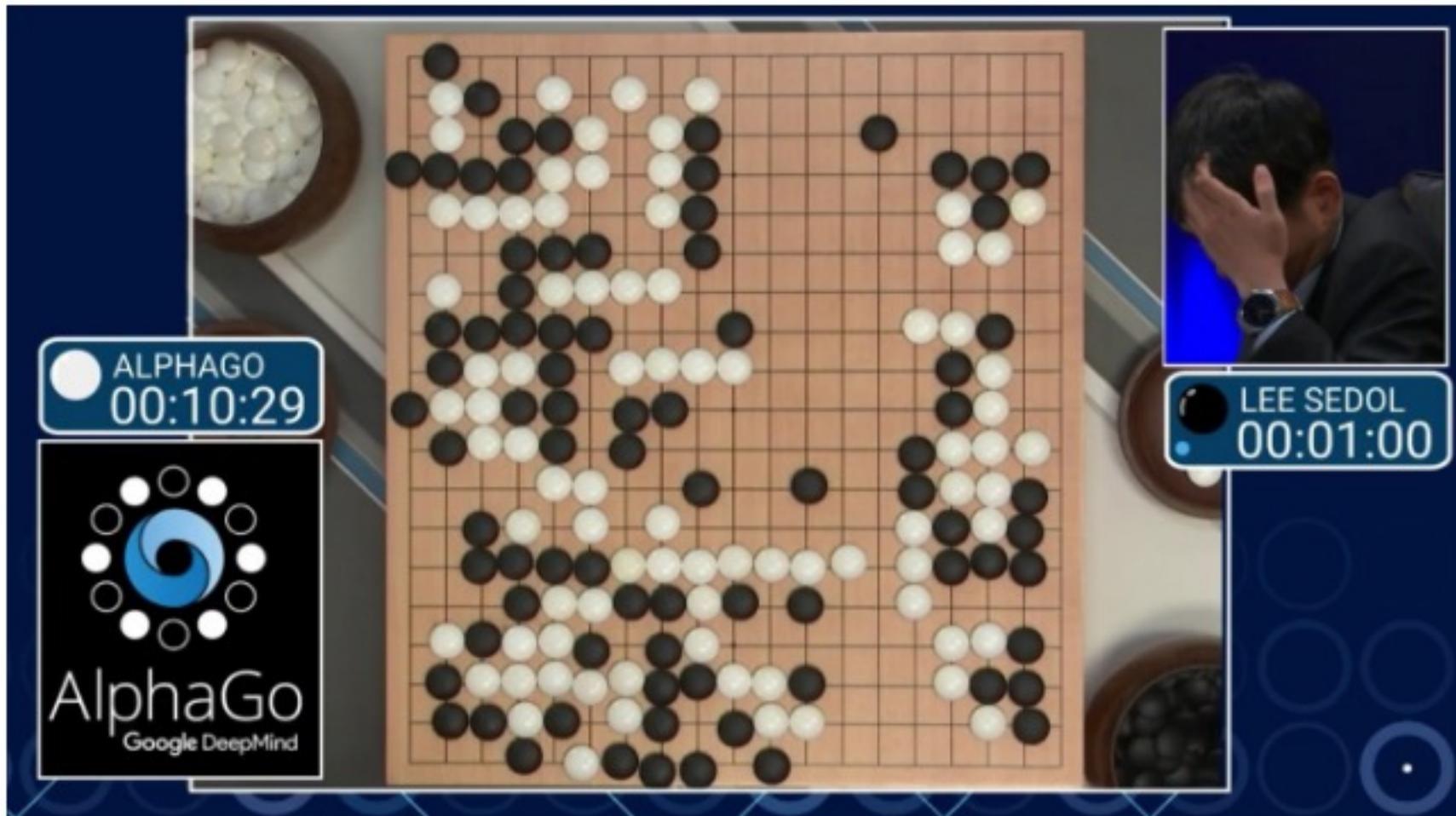
“Deep Blue hasn't proven anythi

The 1997 rematch				
Game #	White	Black	Result	Comment
1	Kasparov	Deep Blue	1–0	
2	Deep Blue	Kasparov	1–0	
3	Kasparov	Deep Blue	½–½	Draw by mutual agreement
4	Deep Blue	Kasparov	½–½	Draw by mutual agreement
5	Kasparov	Deep Blue	½–½	Draw by mutual agreement
6	Deep Blue	Kasparov	1–0	

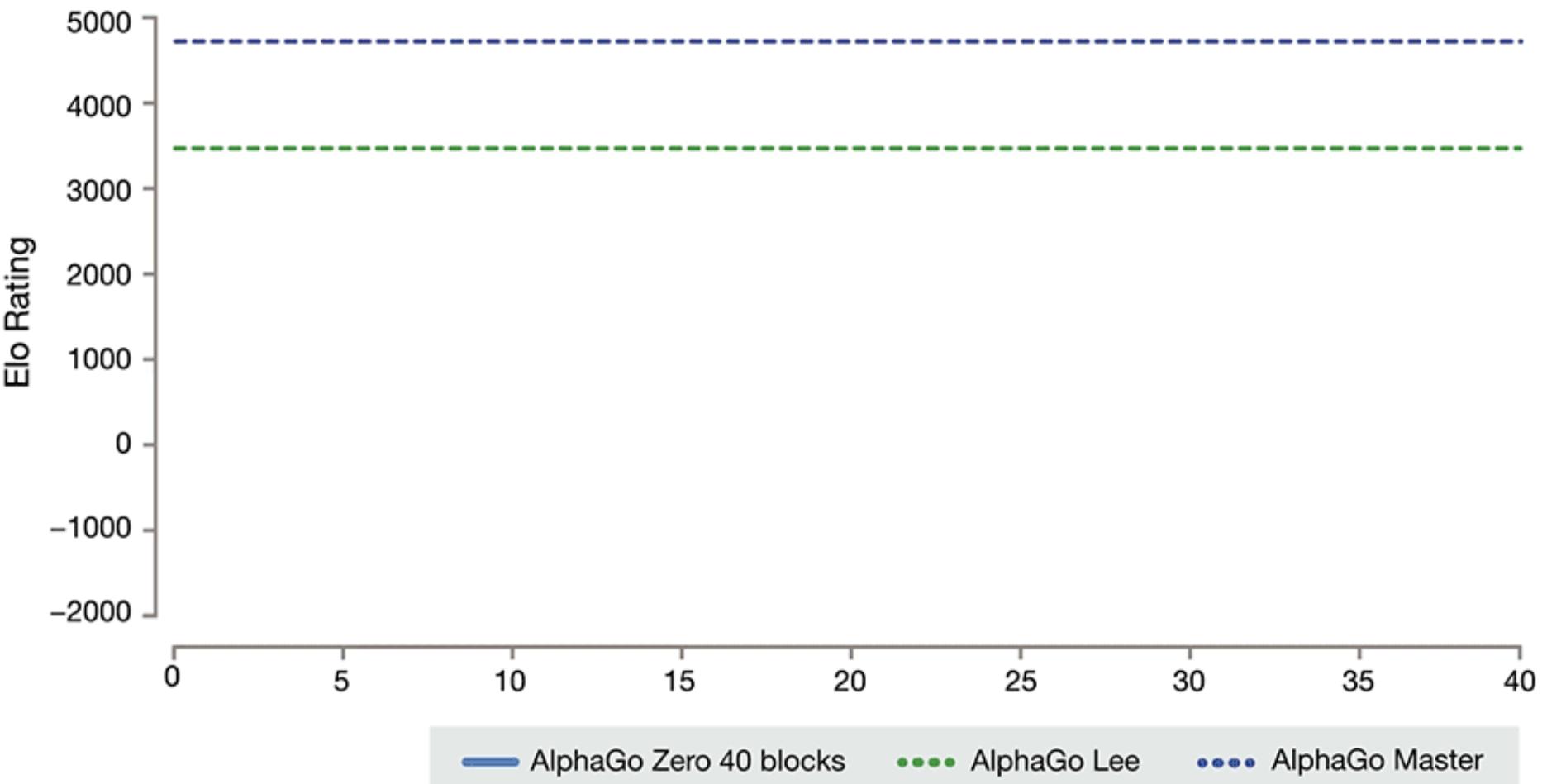
*Result: Deep Blue–Kasparov: 3½–2½*

# Go: AlphaGo vs Lee Sedol

- AlphaGo proiectat de Google DeepMind îl învinge în 2016 campionul mondial la GO



# Go: AlphaGo vs AlphaGo Zero



# Alte aplicații

## Spam filtering



## Machine translation



## Speech recognition



## Advertising and ad placement



## Recommendation systems

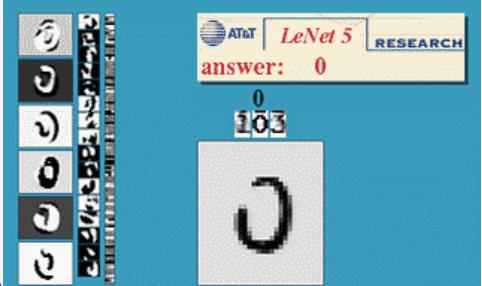


## Driving assistance systems

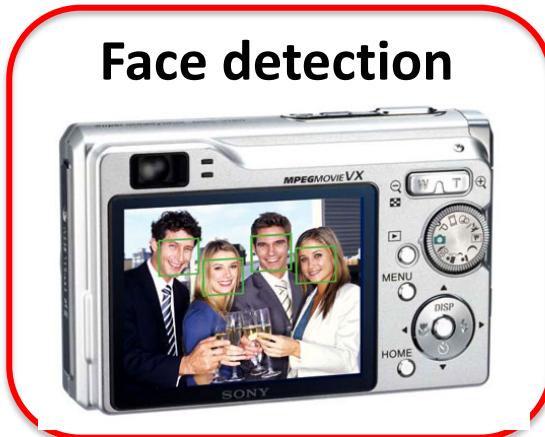


# Alte aplicații

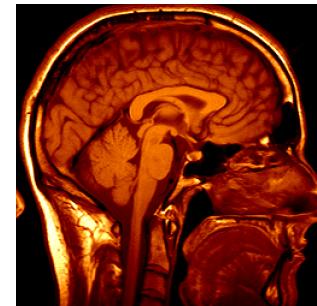
## OCR



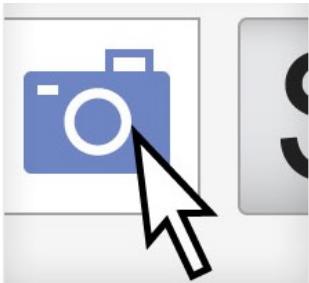
## Face detection



## Medical



## Visual search



## XBOX Kinect



## Video referee



# Kahoot!

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 2

# Recapitulare – cursul trecut

1. Aspecte organizatorice legate de cursul de IA
  1. Prezentarea cursului de IA

# Ce vom studia la acest curs de IA?

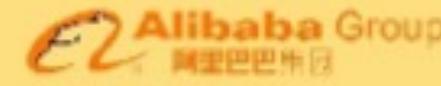
Testul Turing → Subdomenii ale Inteligenței Artificiale:

1. procesarea limbajului natural (comunicare)
2. reprezentarea cunoștințelor (stocare informații)
3. deducție automată (a faptelor pe baza cunoștiințelor)
4. învățare automată (detectare de pattern-uri)
5. vedere artificială (perceperea obiectelor, a scenei)
6. robotică (manipularea obiectelor, mișcare)

Vom studia:

- Învățare automată - primele 7 săptămâni
- Rezolvarea de probleme prin strategii de căutare (informată și neinformată) - următoarele 7 săptămâni

# **Aplicații de succes în Inteligenta Artificială**

	Q1 2009	Q1 2014	Q1 2019
#1	<b>ExxonMobil</b>		 Microsoft
#2	<b>PetroChina</b>		
#3	<b>Walmart</b> 	 Microsoft	 amazon
#4	<b>ICBC</b> 		
#5	 中国移动 China Mobile	<b>BERKSHIRE HATHAWAY INC.</b>	<b>BERKSHIRE HATHAWAY INC.</b>
#6	 Microsoft		
#7	 AT&T		
#8		 	
#9			
#10			
USS	1,900,000,000,000	3,000,000,000,000	6,000,000,000,000

Top 10 Publicly Traded Companies by Market Cap (FT Global 500)



amazon  
alexa

Alexa



Cortana

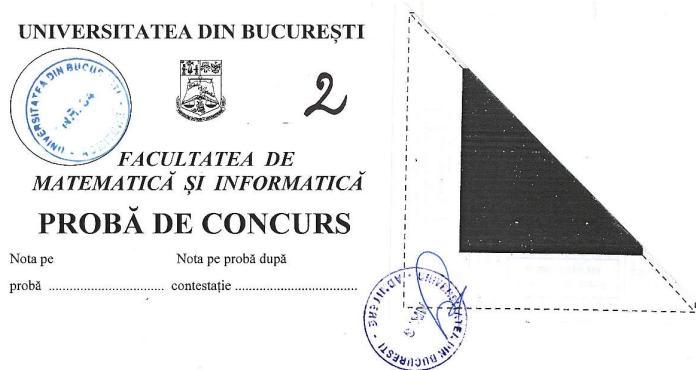


Google



Siri

# Corectarea automată a testelor grilă

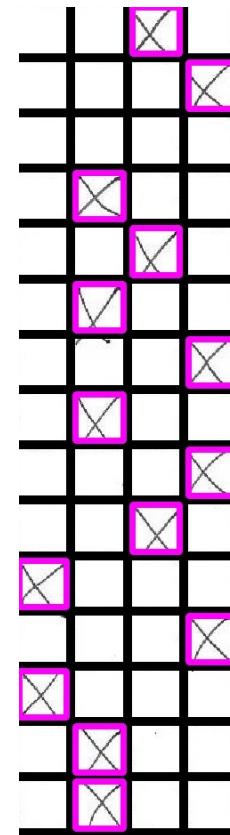


**TEST GRILĂ**

**INFORMATICĂ**  1      **FIZICĂ**  \_\_\_\_\_  
Număr varianță

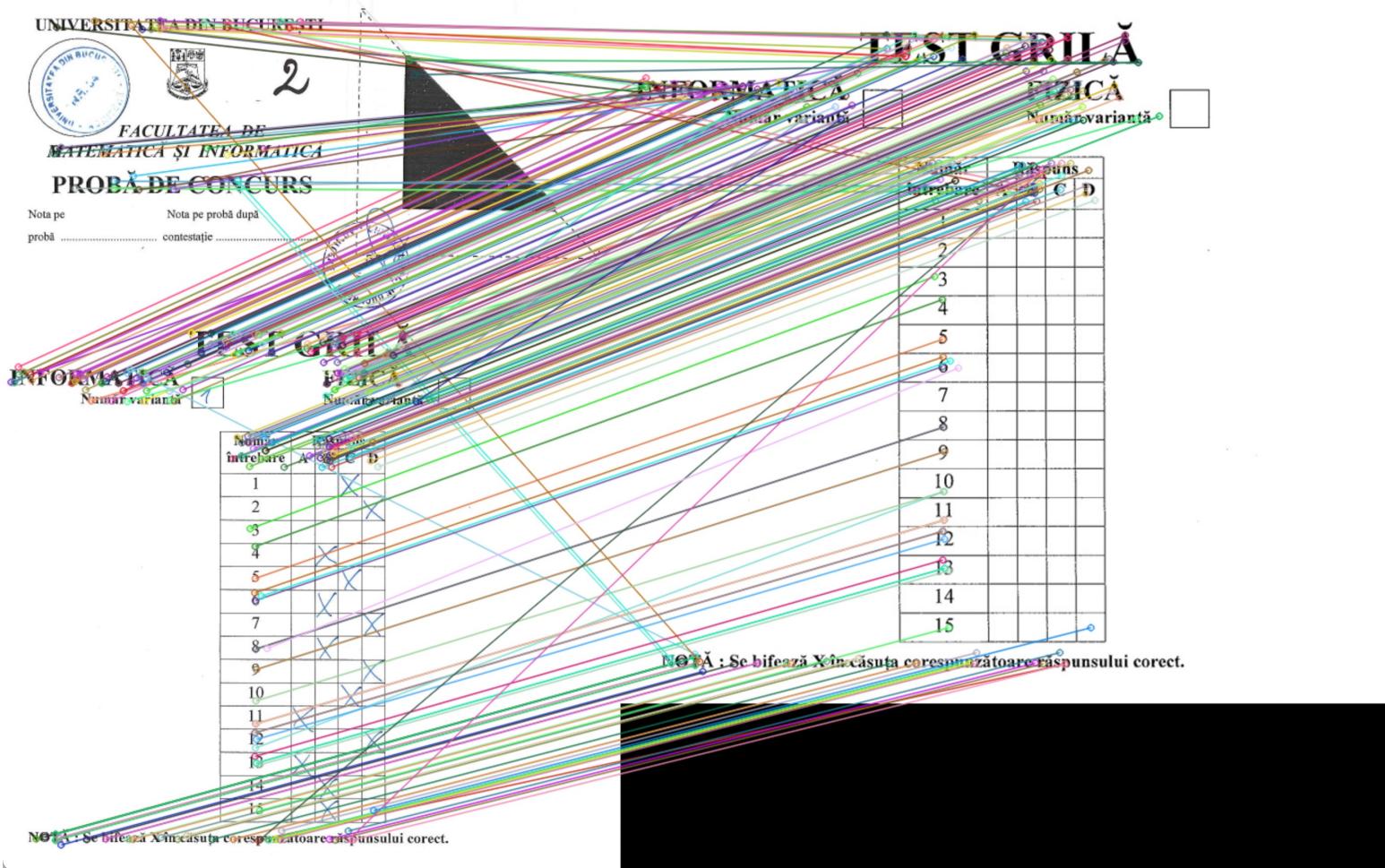
Număr întrebare	A	B	C	D
1			X	
2			X	
3				
4	X			
5		X		
6	X			
7		X		
8	X			
9			X	
10		X		
11	X			
12			X	
13	X			
14		X		
15		X		

NOTĂ : Se bifează X în căsuța corespunzătoare răspunsului corect.



Numar Lucrare	Varianta	0X	1X	mX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Numar grile corecte	Nota
2	Informatica_1	1	14	0	C	D	B	C	B	D	B	D	C	A	D	A	B	B		2	2.2

# Corectarea automată a testelor grilă



# IA pentru jocuri

- antrenate pentru a găsi cele mai bune strategii de joc
- paradigma învățării pe bază de recompensă (reinforcement learning)
  - recompensă +/- pentru câștigarea/pierderea unui joc

# Şah: Deep Blue vs Kasparov

- Deep Blue (proiectat de IBM) îl învinge în 1997 pe Kasparov după ce în prealabil pierduse în 1996

- **1996: Kasparov câştigă**

“I could feel – I could smell  
kind of intelligence across 1

The 1996 match				
Game #	White	Black	Result	Comment
1	Deep Blue	Kasparov	1–0	
2	Kasparov	Deep Blue	1–0	
3	Deep Blue	Kasparov	½–½	Draw by mutual agreement
4	Kasparov	Deep Blue	½–½	Draw by mutual agreement
5	Deep Blue	Kasparov	0–1	Kasparov offered a draw after the 23rd move.
6	Kasparov	Deep Blue	1–0	

*Result: Kasparov–Deep Blue: 4–2*

- **1997: Deep Blue câştigă**

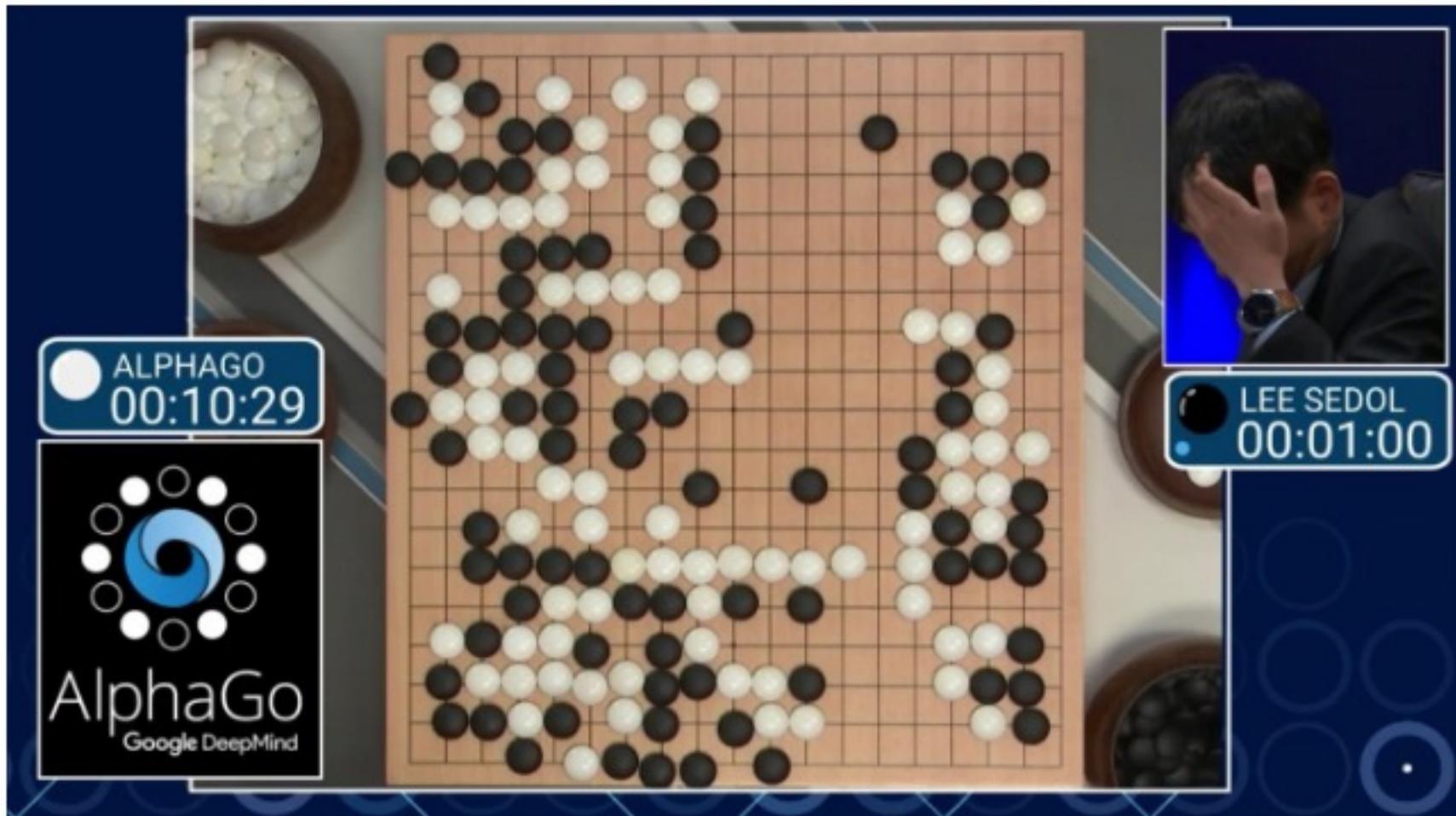
“Deep Blue hasn't proven anythi

The 1997 rematch				
Game #	White	Black	Result	Comment
1	Kasparov	Deep Blue	1–0	
2	Deep Blue	Kasparov	1–0	
3	Kasparov	Deep Blue	½–½	Draw by mutual agreement
4	Deep Blue	Kasparov	½–½	Draw by mutual agreement
5	Kasparov	Deep Blue	½–½	Draw by mutual agreement
6	Deep Blue	Kasparov	1–0	

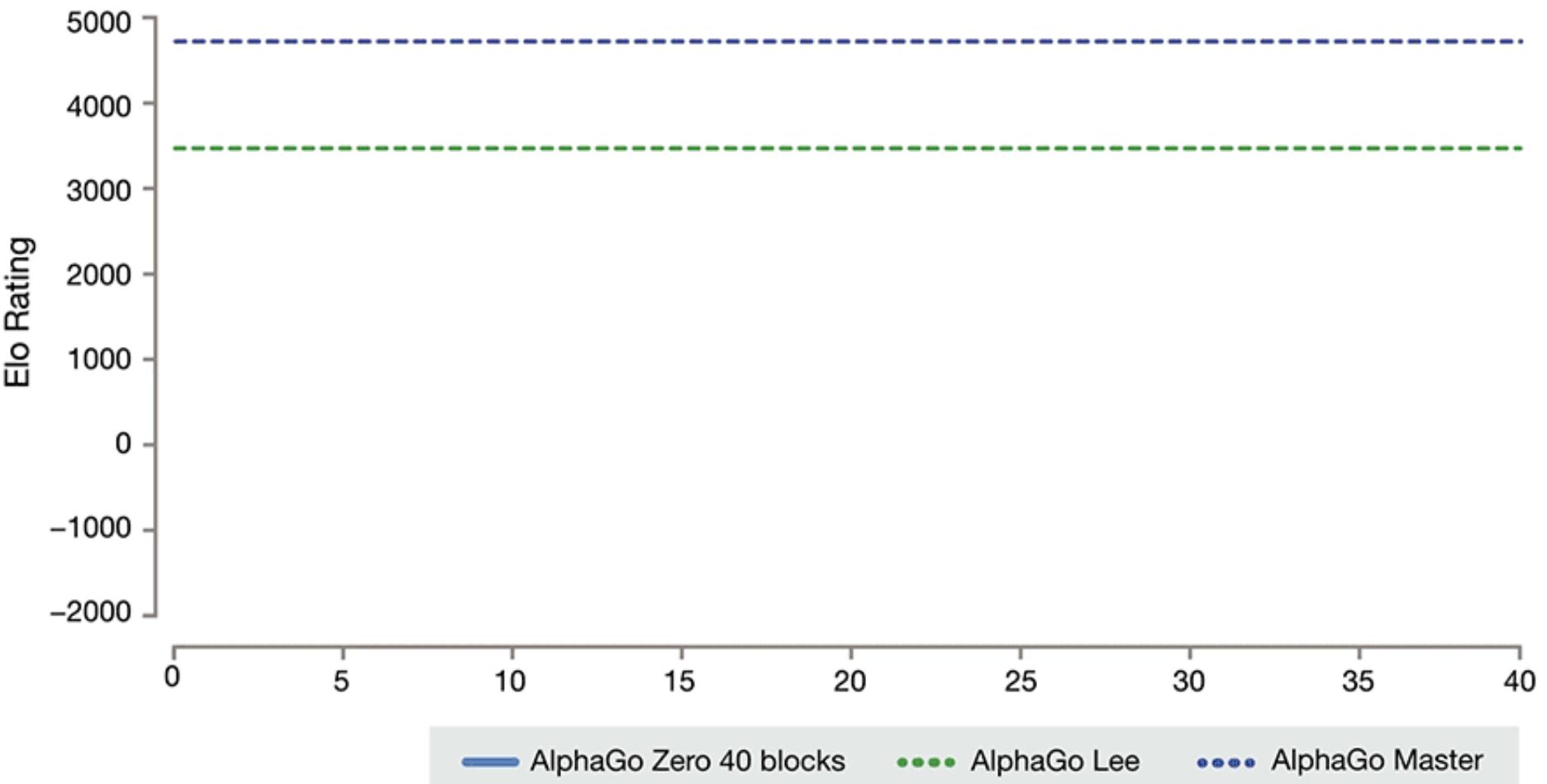
*Result: Deep Blue–Kasparov: 3½–2½*

# Go: AlphaGo vs Lee Sedol

- AlphaGo proiectat de Google DeepMind îl învinge în 2016 campionul mondial la GO



# Go: AlphaGo vs AlphaGo Zero



# Alte aplicații

## Spam filtering



## Machine translation



## Speech recognition



## Advertising and ad placement



## Recommendation systems

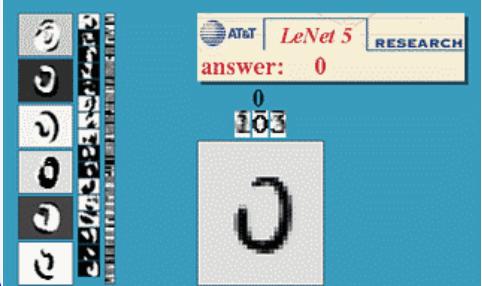


## Driving assistance systems

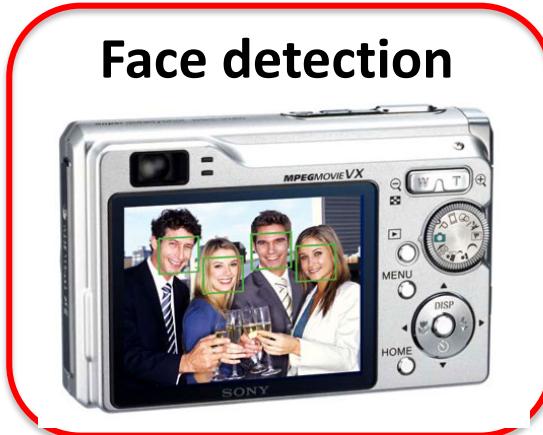


# Alte aplicații

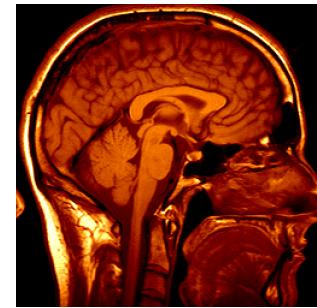
## OCR



## Face detection



## Medical



## Visual search



## XBOX Kinect



## Video referee



# Cuprinsul cursului de azi

1. Învățarea Automată: definiție și terminologie
2. Paradigme de învățare
3. Metrici de măsurare a performanței pentru clasificare
4. Modelul celor mai apropiati k vecini (k-nearest neighbors)

# Învățarea Automată: definiție și terminologie

# Ce este învățarea automată?

**Învățarea automată** – domeniu care studiază modul în care mașinile (calculatoarele) pot fi înzestrate cu abilitatea de a învăța, fără ca aceasta să fie programată în mod explicit.

Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

Un program este înzestrat cu abilitatea de a învăța dacă performanța sa  $P$  pe un task (problemă)  $T$  se îmbunătăște pe măsură ce acumulează mai multă experiență (date)  $E$ .

A computer program is said to learn from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$ .

Tom Mitchell, 1997

**Învățarea** se referă la recunoașterea unor tipare / structuri (patterns) complexe și la luarea deciziilor inteligente bazate pe observațiile din **date**

# Abilitatea de a învăța

- Arthur Samuel (1959) a scris un program pentru a juca dame (probabil primul program bazat pe conceptul de învățare)
- Programul a jucat împotriva lui însuși 10.000 de jocuri
- Programul a fost conceput să găsească ce poziții ale tablei de joc erau bune sau rele în funcție de probabilitatea de a câștiga sau pierde
- În acest caz:

$E = 10000$  de jocuri

$T =$  joacă dame

$P =$  câștigă sau nu



# Când se folosește învățarea automată?

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Cum ați scrie un program pe calculator care vrea să găsească mașinile într-o imagine?



# Când se folosește învățarea automată?

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- Cum ați scrie un program pe calculator care să filtreze mailurile spam?

The screenshot shows a list of flagged messages from a spam filter interface. The messages are listed in two sections: 'This month' and above it.

Category	Message Content
! The CHOICE Home Warranty	Keep Your Home Cool with \$50 Off! – Keep Your Home Co...
! Final, Final (2)	Request To Terminate Your Google™ Account Has Bee...
This month	
! Mr. Tony Adams (2)	Congratulation You Have Won – You have Won \$2.5Million ...
! ZENITH BANK PLC	COMPENSATION UNIT, IN AFFILIATION WITH THE UNITE...
! John Mill	UNITED NATION AND EUROPEAN UNION OFFICIAL WIN...

At the bottom left, there is a green circular icon containing a white letter 'J' and the email address 'urytopk4@crocus.ocn.ne.jp'.

# Când se folosește învățarea automată?

- Se aplică în situații în care este foarte greu (imposibil) să definim un set de reguli de mână / să scriem un program
- probleme pentru care soluțiile tradiționale necesită multă muncă manuală și liste interminabile de reguli, greu de menținut actualizate la schimbările repetitive
  - detectarea de spam, traducere automată
- probleme aproape “imposibil” de programat în abordarea de programare tradițională
  - detectarea de obiecte, recunoașterea vorbirii
- găsirea de corelații în seturi mari de date

# Terminologie în învățarea automată

- vom folosi un **set de date (dataset)** cu informații despre două tipuri de fructe

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

# Terminologie în învățarea automată

- o **etichetă (label, target)** este ceea ce încercăm să prezicem

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

# Terminologie în învățarea automată

- o **etichetă (label, target)** este ceea ce încercăm să prezicem
  - dacă este discretă, se mai numește **clasă**, problemă de *clasificare*
  - dacă este continuă, problemă de *regresie*
  - diferența dintre clasificare și regresie: măsurarea erorii

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

# Terminologie în învățarea automată

- un **exemplu** (**example**, **sample point**) este o instanță particulară a datelor (**data point**)
- poate include sau nu o etichetă (**label**) – date *etichetate* vs. date *neetichetate*

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

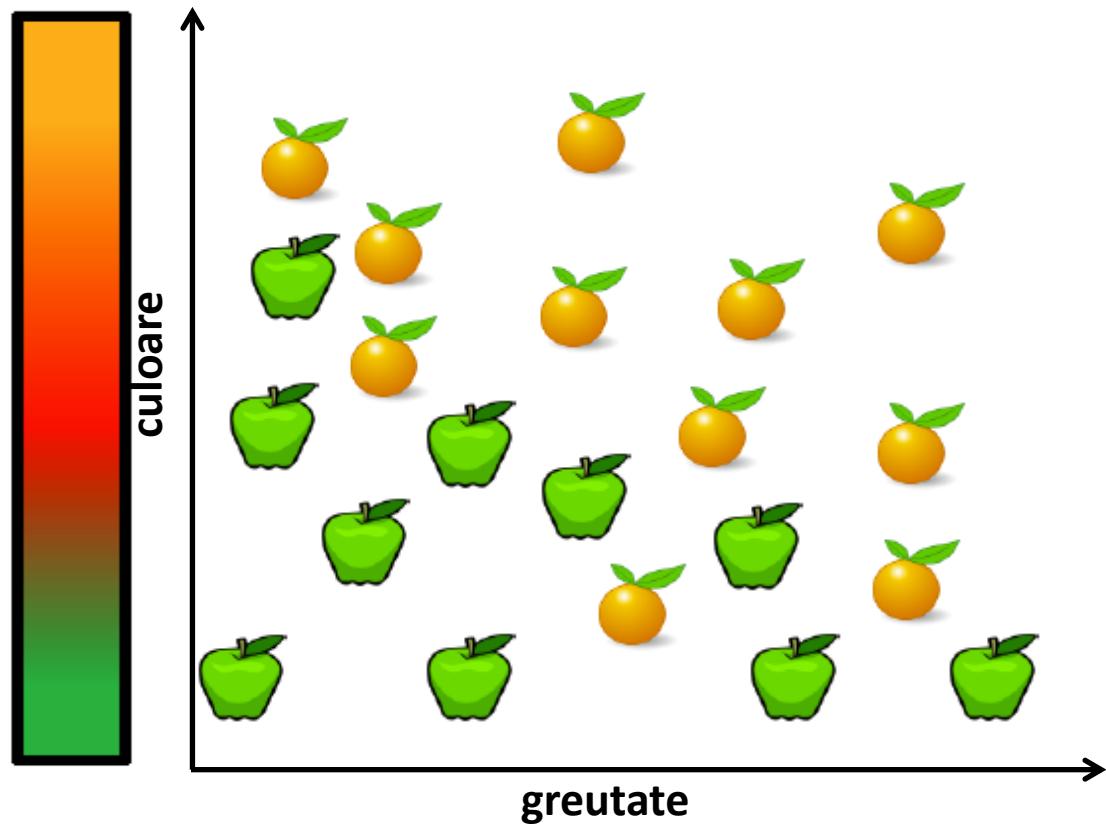
# Terminologie în învățarea automată

- o **caracteristică** (**feature, attribute**) este o proprietate măsurabilă pe baza observațiilor
- toate caracteristicile unui exemplu (direct măsurabile sau obținute) formează un **vector de caracteristici** (**feature vector**) – de obicei au valori numerice

Greutate (g)	Culoare	Textură	pH	Etichetă
84	Verde	Moale	3.5	Măr
121	Portocaliu	Aspră	3.9	Portocală
85	Roșu	Moale	3.3	Măr
101	Portocaliu	Moale	3.7	Portocală
111	Verde	Aspră	3.5	Măr
...	...	...	...	...
117	Roșu	Aspră	3.4	Portocală

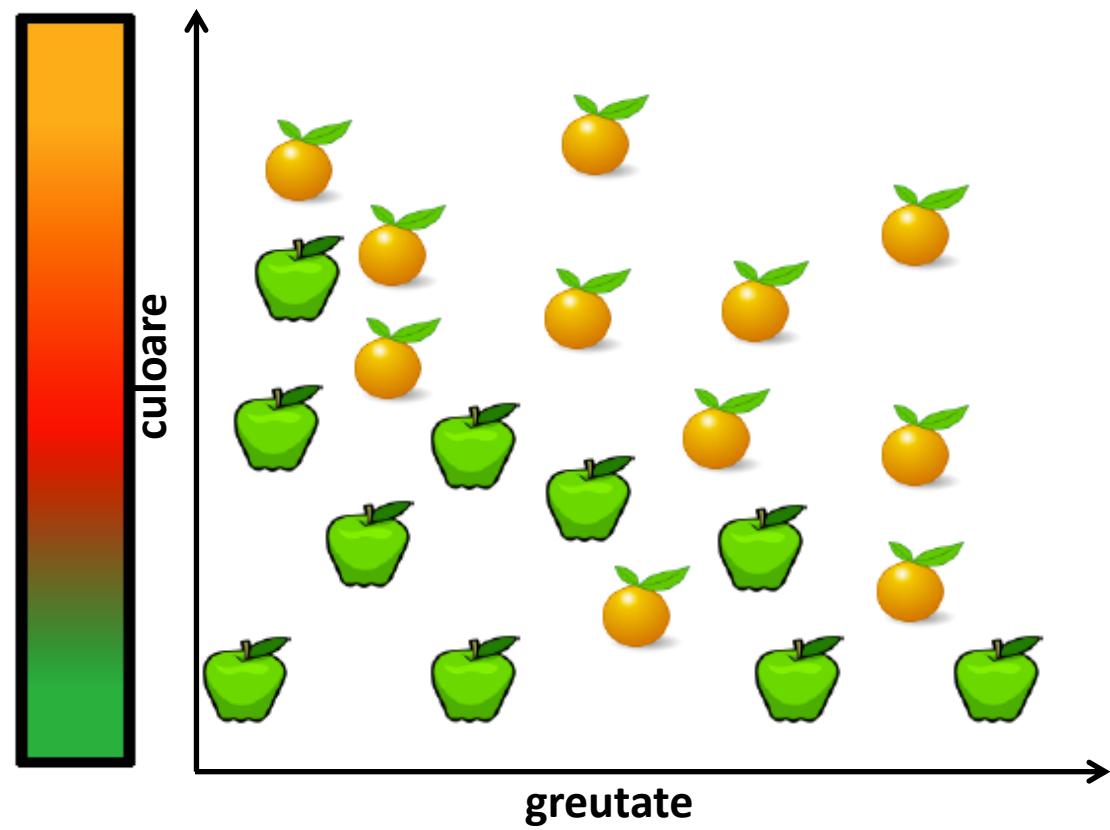
# Terminologie în învățarea automată

- un **model** sau o **ipoteză** este o relație *determinată* dintre caracteristici și etichete



# Terminologie în învățarea automată

- un **model** sau o **ipoteză** este o relație *determinată* dintre caracteristici și etichete
- determinarea relației pe baza unor exemple se numește **antrenarea modelului (training, fitting)**



# Terminologie în învățarea automată

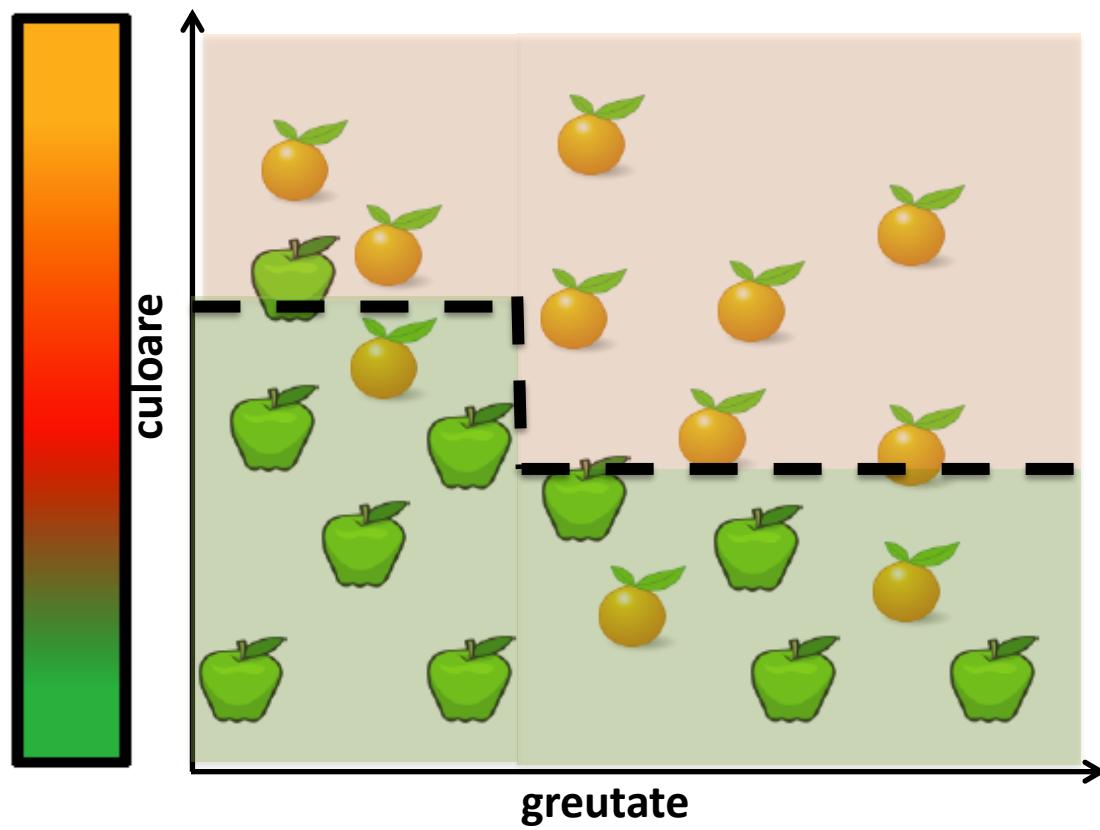
- un **model** sau o **ipoteză** este o relație *determinată* dintre caracteristici și etichete
- determinarea relației pe baza unor exemple se numește **antrenarea modelului (training, fitting)**

Pentru clasificare, fiecărui model îi corespunde o **graniță de separare (decision boundary)**

Etichete prezise

Etichete prezise		
Etichete reale	Green Apple	Orange
Green Apple	9	1
Orange	3	8

Matrice de confuzie



# Terminologie în învățarea automată

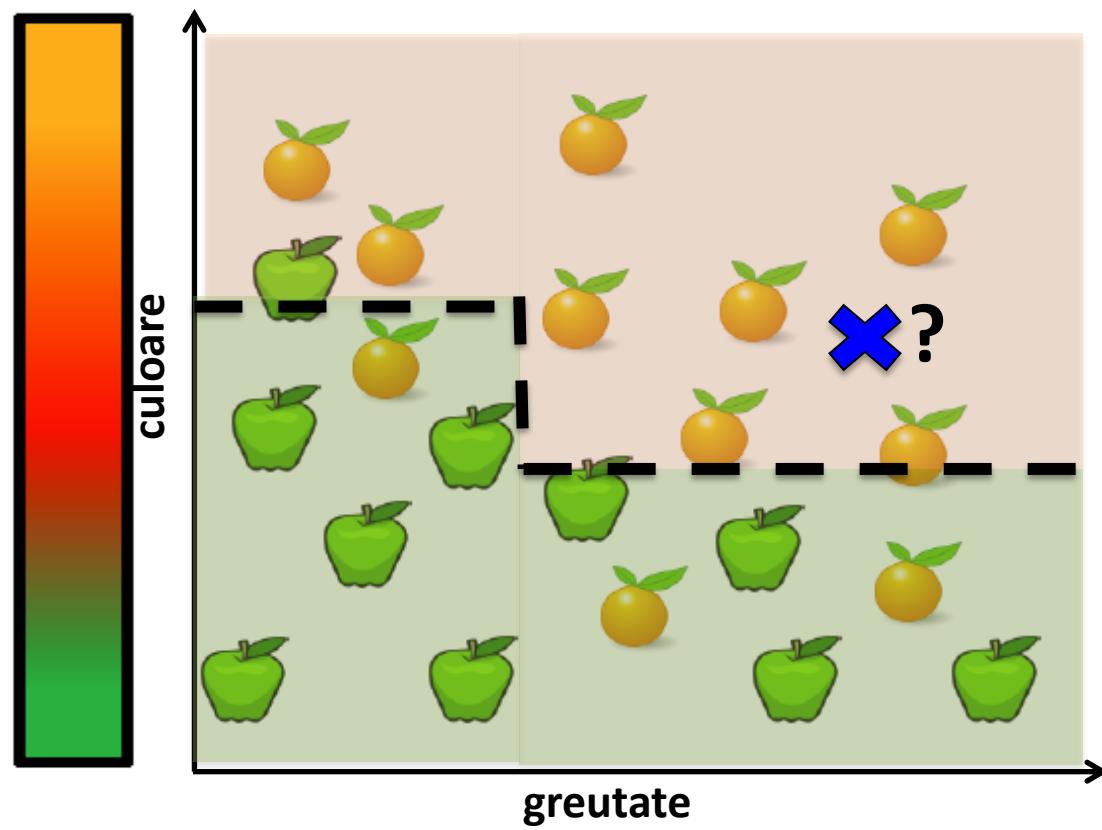
- în timpul **inferenței**, modelul antrenat este folosit la realizare de predicții pentru exemple test, necunoscute

Pentru clasificare, fiecărui model îi corespunde o **graniță de separare** (decision boundary)

Etichete prezise

Etichete prezise		
Etichete reale	green apple	orange
green apple	9	1
orange	3	8

Matrice de confuzie



# Terminologie în învățarea automată

- în timpul **inferenței**, modelul antrenat este folosit la realizare de predicții pentru exemple test, necunoscute
- algoritmii de învățare pot determina ipoteze diferite, cu granițe de separare de forme diferite

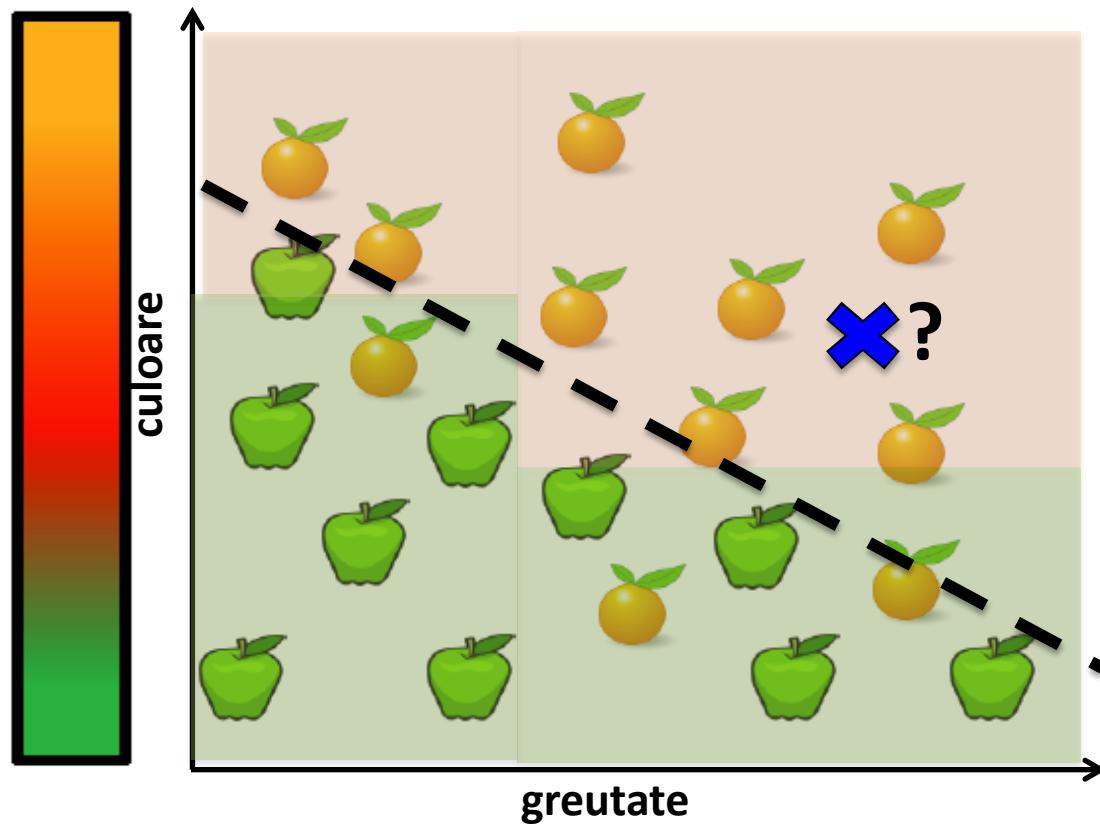
Pentru clasificare, fiecărui model îi corespunde o **graniță de separare** (decision boundary)

Etichete prezise



Etichete reale	apple	orange
apple	10	0
orange	3	8

Matrice de confuzie



# Terminologie în învățarea automată

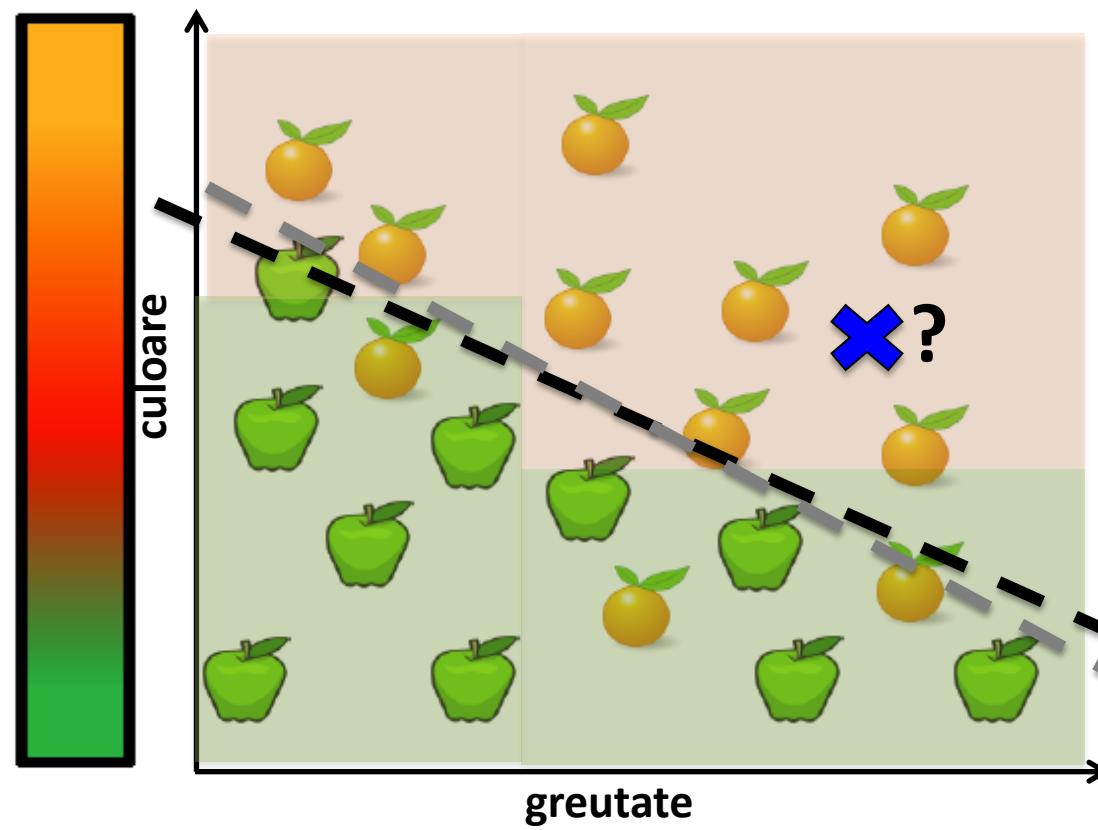
- algoritmii de învățare pot determina ipoteze diferite, cu granițe de separare de forme diferite
- algoritmii pot avea hiperparametri care controlează felul în care decurge învățarea și se reflectă asupra modelului învățat

Pentru clasificare, fiecărui model îi corespunde o **graniță de separare (decision boundary)**

Etichete prezise

Etichete prezise		
Etichete reale	green apple	orange
green apple	10	0
orange	3	8

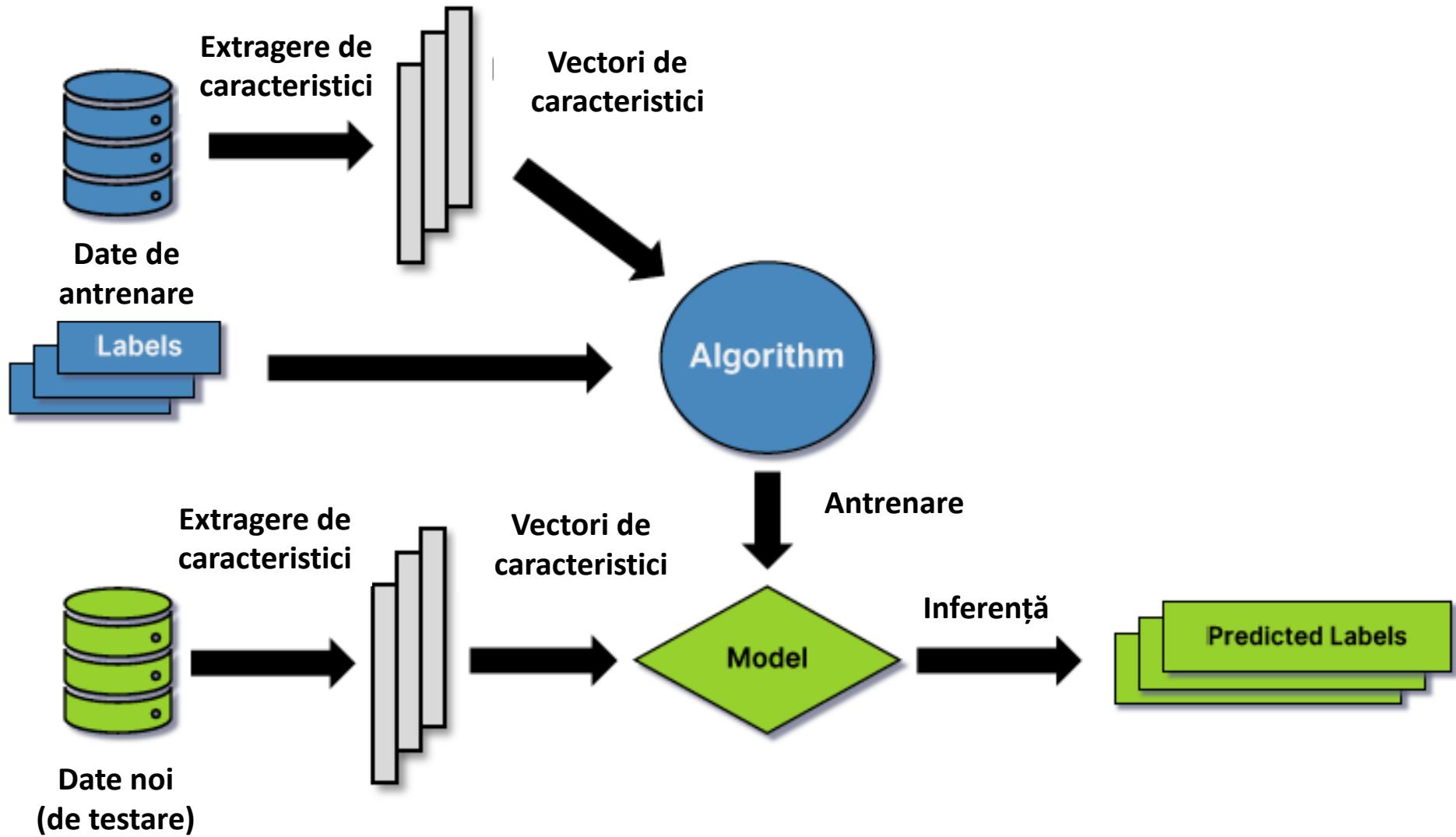
Matrice de confuzie



# Terminologie în învățarea automată

- **etichetă** (label) → ce încercăm să prezicem
- **caracteristică** (feature) → proprietate măsurabilă a unui punct din exemplele din date
- **model** (ipoteză) → relația dintre caracteristici și etichete
- **antrenare** → stabilirea relației pe baza unei multimi de puncte
- **inferență** → realizarea de predicții pe date necunoscute
- **algoritm** → definește cum se face învățarea, poate fi restricționat la o anumită mulțime de ipoteze, fie prin alegerea unui anumit spațiu, fie pe baza hiperparametrilor

# Fluxul tipic în învățarea automată



# Paradigme de învățare

# Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

# Învățarea supervizată

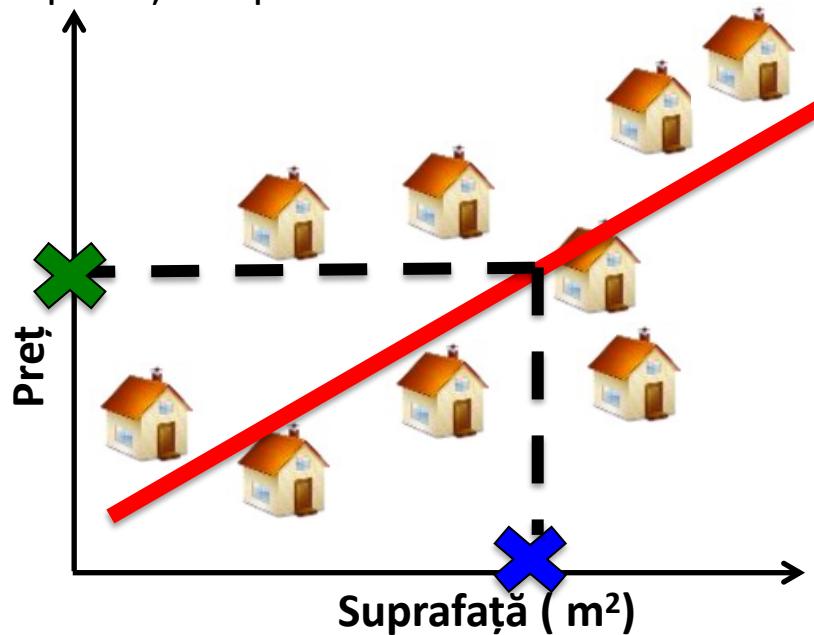
- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății



Regresie liniară

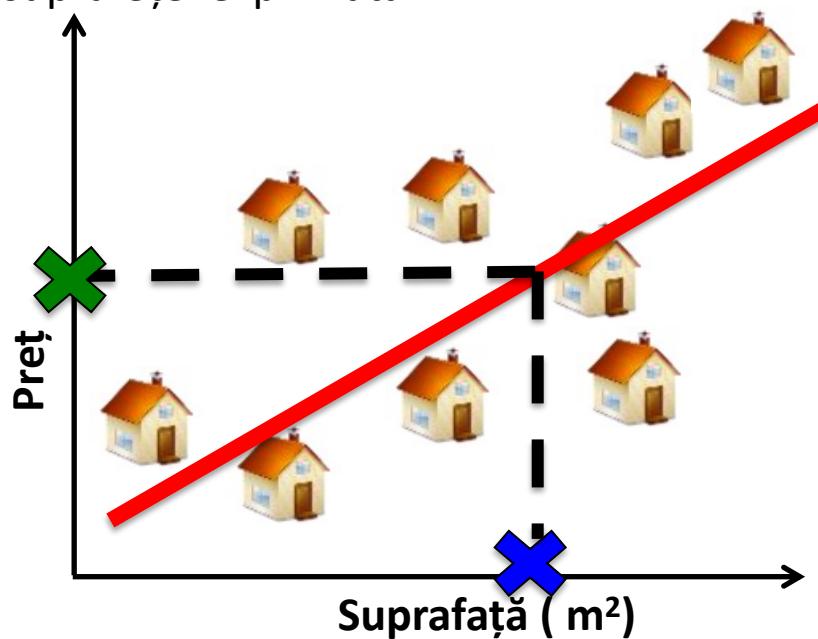
Regresie pe baza kNN

# Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

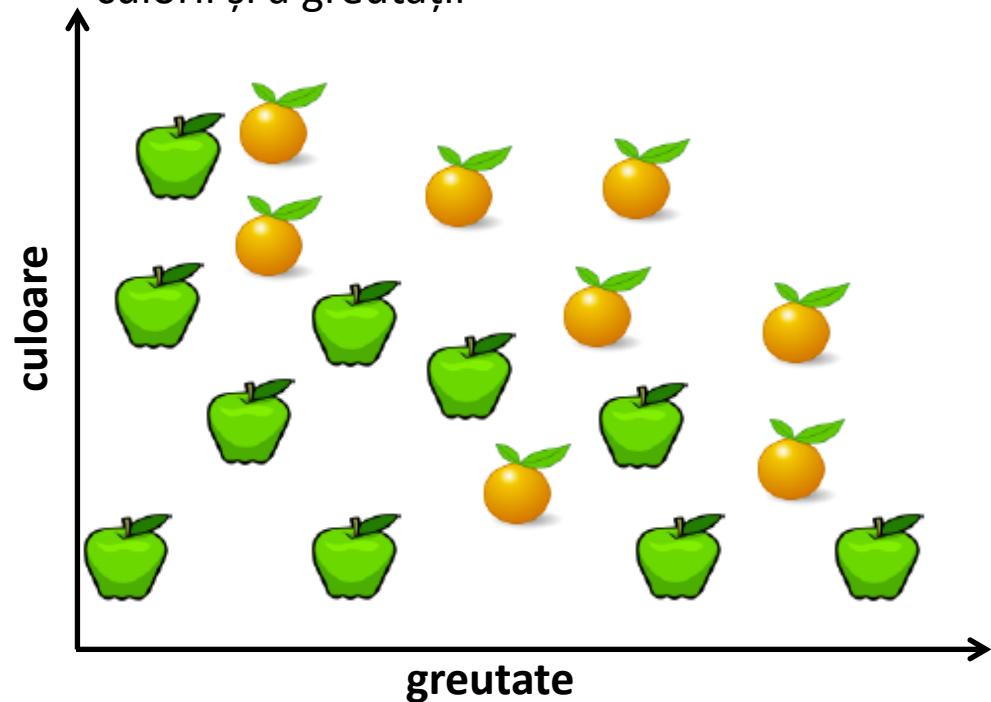


**Regresie liniară**

**Regresie pe baza kNN**

**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății

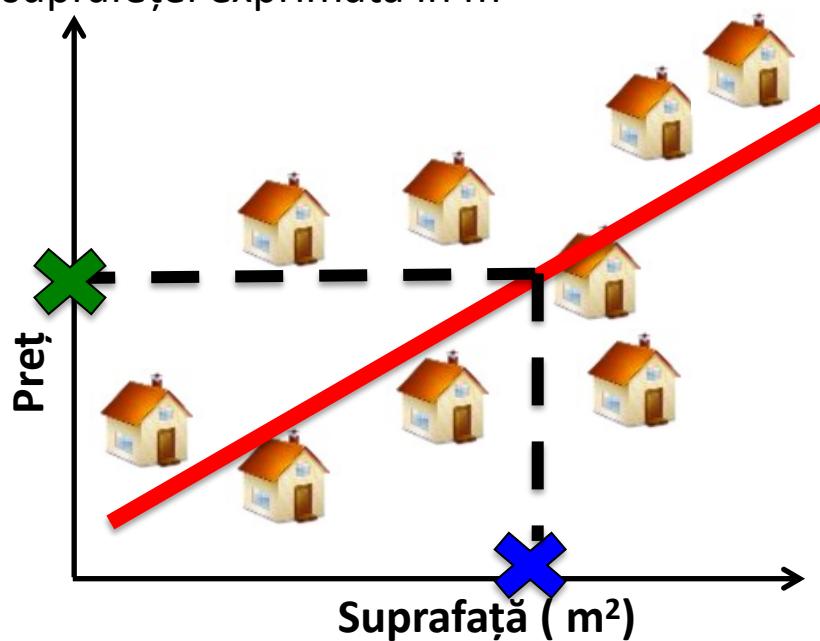


# Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

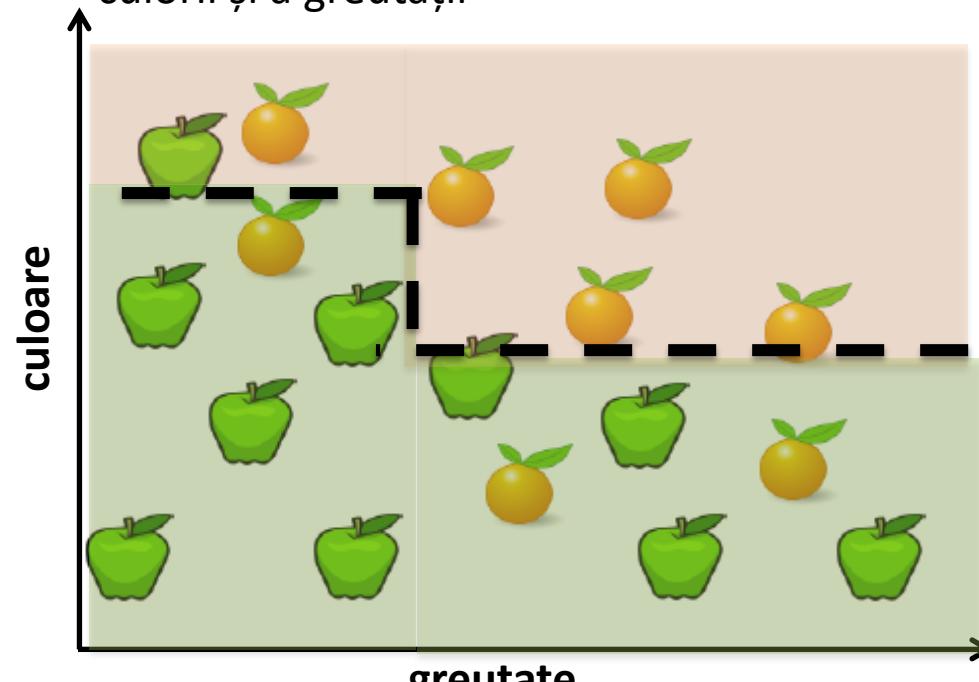


**Regresie liniară**

Regresie pe baza kNN

**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății



Cei mai apropiati k vecini (kNN),  
SVMs, Rețele neuronale

# Învățarea nesupervizată

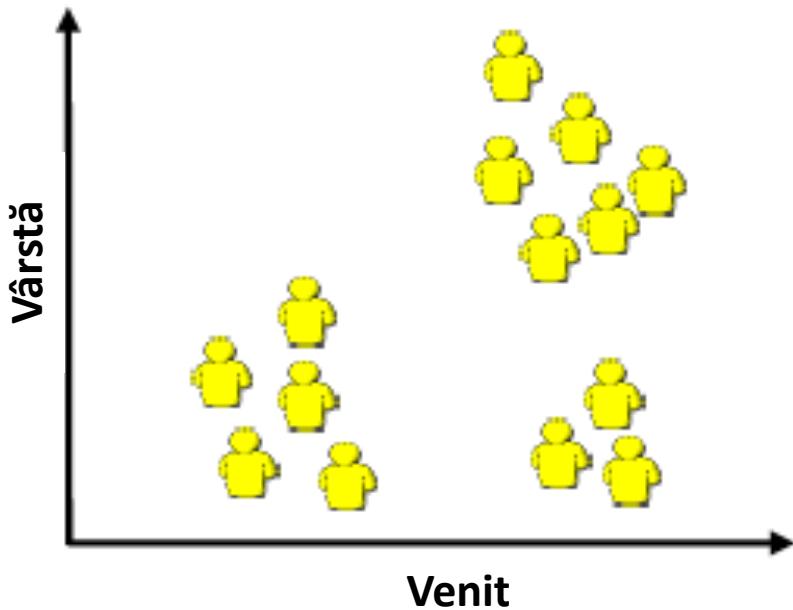
- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele

# Învățarea nesupervizată

- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele

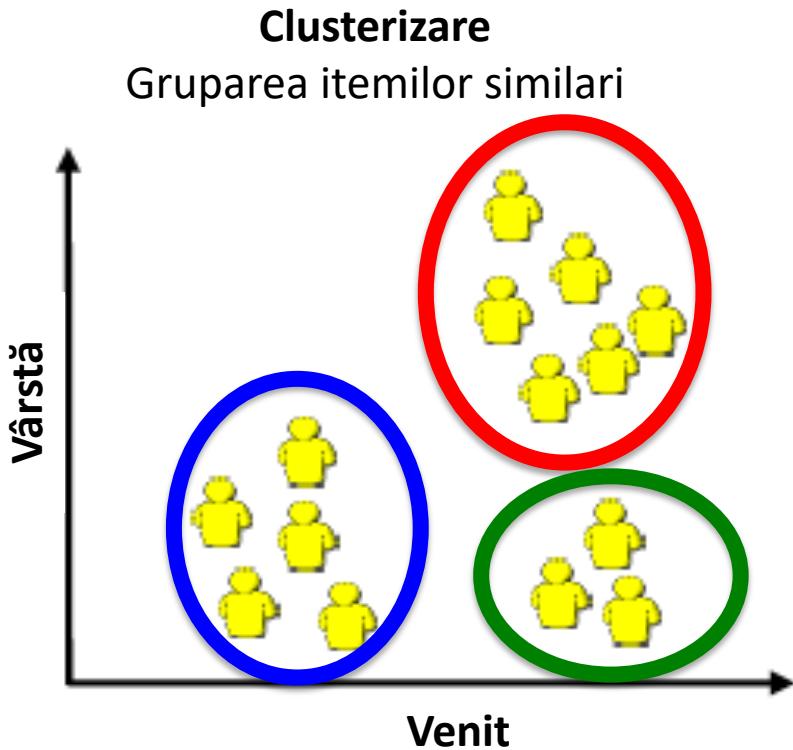
## Clusterizare

Gruparea itemilor similari



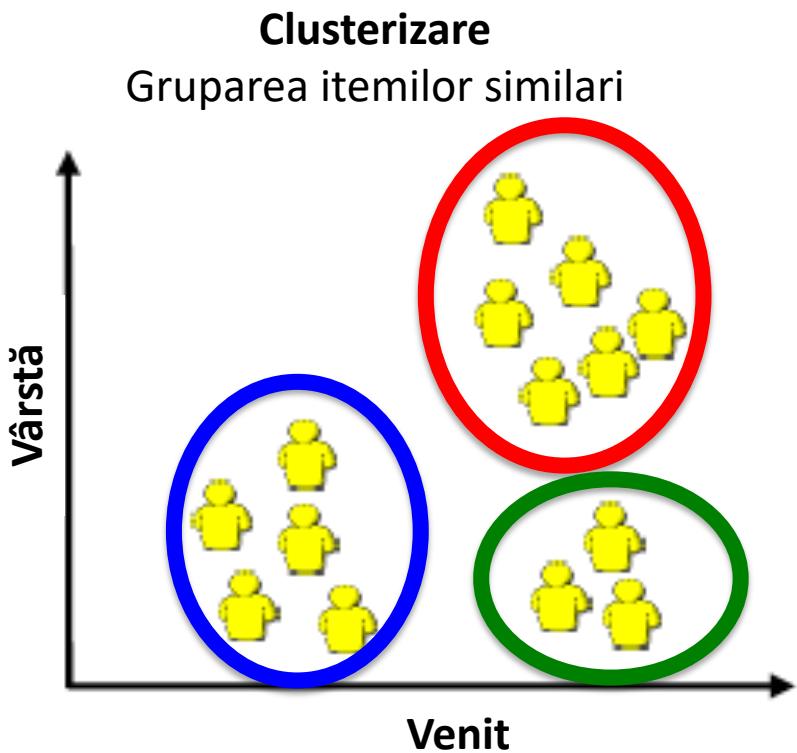
# Învățarea nesupervizată

- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele



# Învățarea nesupervizată

- nu trebuie să prezicem o etichetă, vrem să găsim cum sunt structurate datele

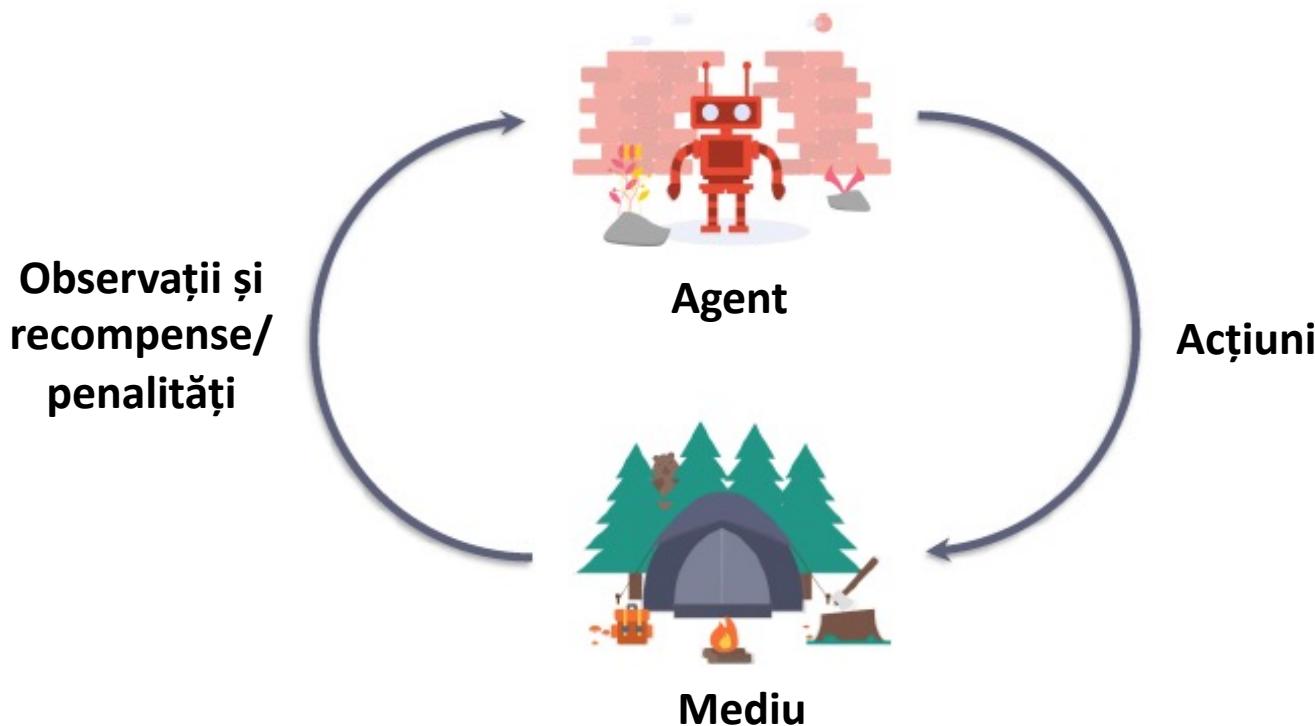


**Căutarea regulilor de asociere**  
Gruparea pattern-urilor frecvente  
și a corelațiilor



# Învățarea pe bază de recompense (reinforcement learning)

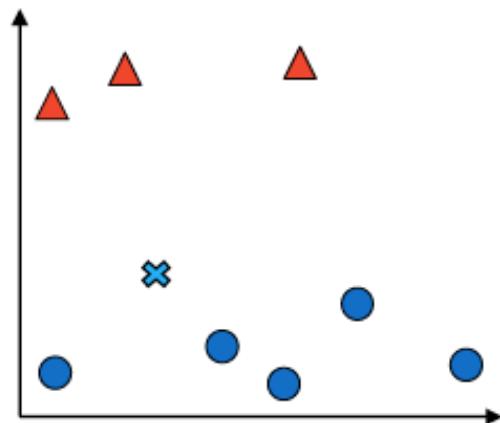
- nu există etichete, ci numai recompense sau penalități pentru realizarea unor acțiuni



# Învățarea semi-supervizată

- vrem să prezicem o etichetă
- avem la dispoziție numai un număr mic de exemple etichetate

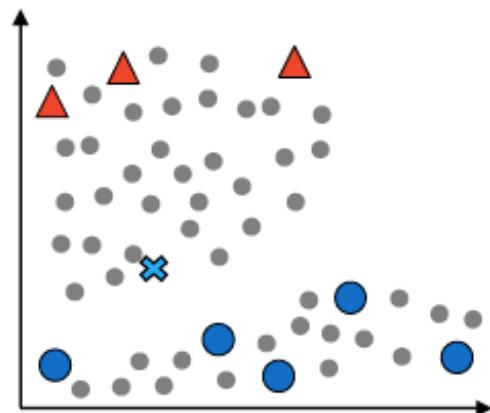
▲ or ● ?



# Învățarea semi-supervizată

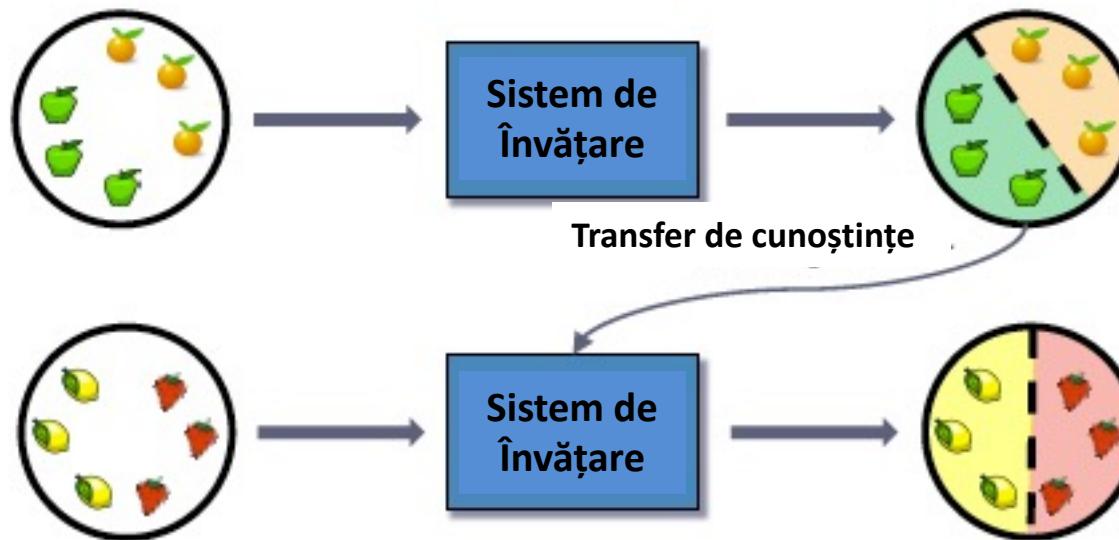
- vrem să prezicem o etichetă
- avem la dispoziție numai un număr mic de exemple etichetate
  - + un număr mare de exemple neetichetate
- datele neetichetate pot îmbunătăți performanțele algoritmilor antrenați în manieră supervizată

▲ or ● ?



# Învățarea prin transfer

- folosim cunoștințele dobândite din rezolvarea unei probleme pentru rezolvarea altei probleme înrudite
- paradigmă folosită în Deep Learning
  - folosirea unui model care a fost pre-antrenat pe un set de date cu multe exemple



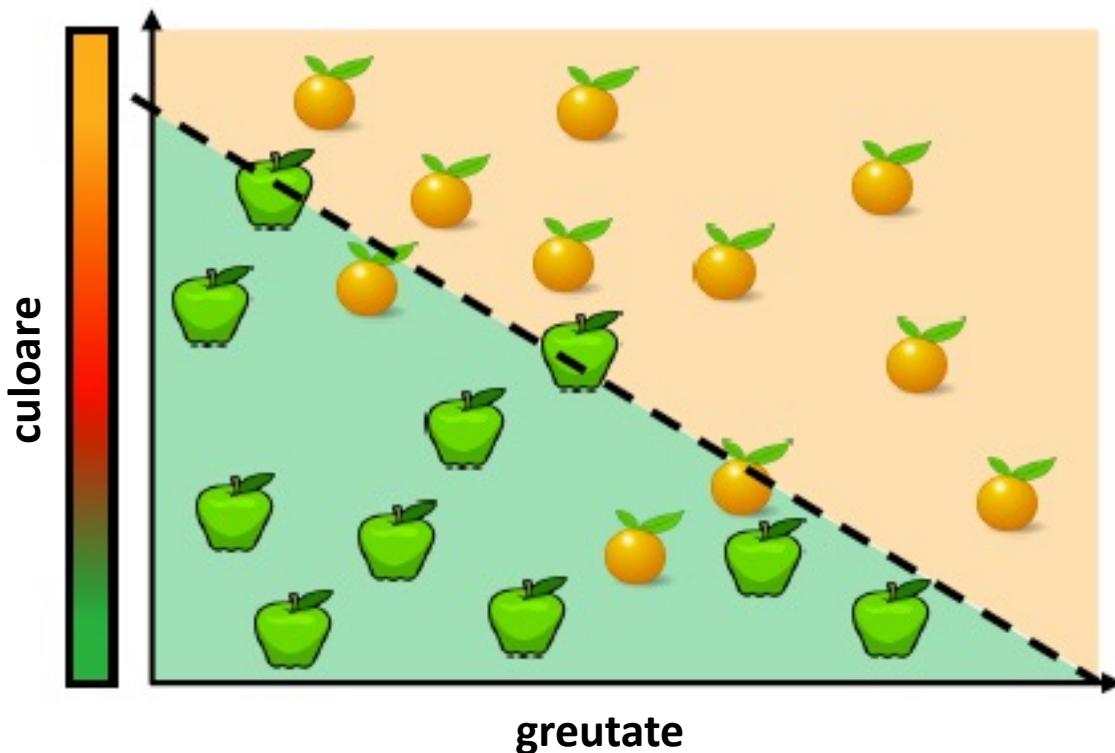
# Paradigme de învățare

- **Învățare Supervizată** → prezicem o etichetă
  - etichetă continuă → **Regresie**
  - etichetă discretă → **Clasificare**
- **Învățarea Nesupervizată** → descoperirea de structuri în date
  - gruparea itemilor similari → **Clusterizare**
  - pattern-uri frecvente → **Căutarea regulilor de asociere**
- **Învățare pe bază de recompense** → fără etichete, ci numai recompense sau penalități pentru realizarea unor acțiuni
- **Învățare semi-supervizată** → câteva date etichetate, multe neetichetate
- **Învățarea prin transfer** → folosirea unui model antrenat pe o problemă pentru învățarea mai rapidă pentru altă problemă

# Metrici de măsurare a performanței pentru clasificare

# Matricea de confuzie

- O matrice de confuzie este un tabel care descrie performanța unui model de clasificare prin precizarea numărului de erori care apar când exemplele sunt clasificate greșit



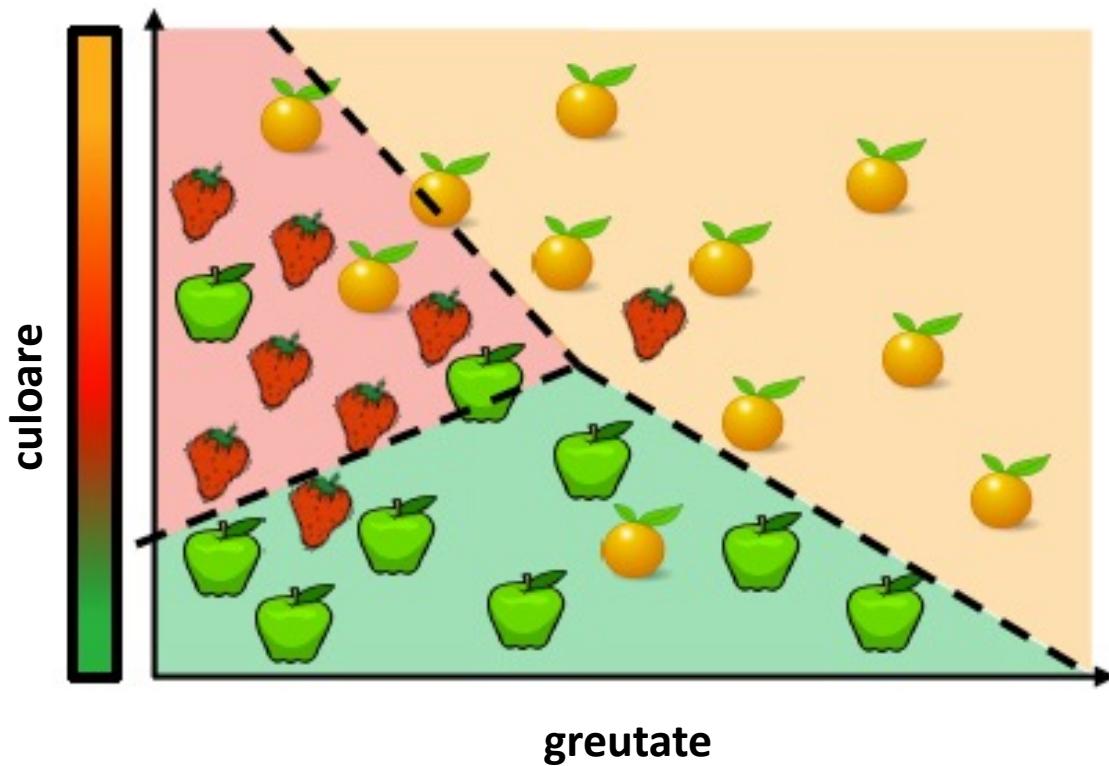
**Etichete prezise**

Etichete reale		
	9	1
	3	8

**Matricea de confuzie**

# Matricea de confuzie

- O matrice de confuzie este un tabel care descrie performanța unui model de clasificare prin precizarea numărului de erori care apar când exemplele sunt clasificate greșit

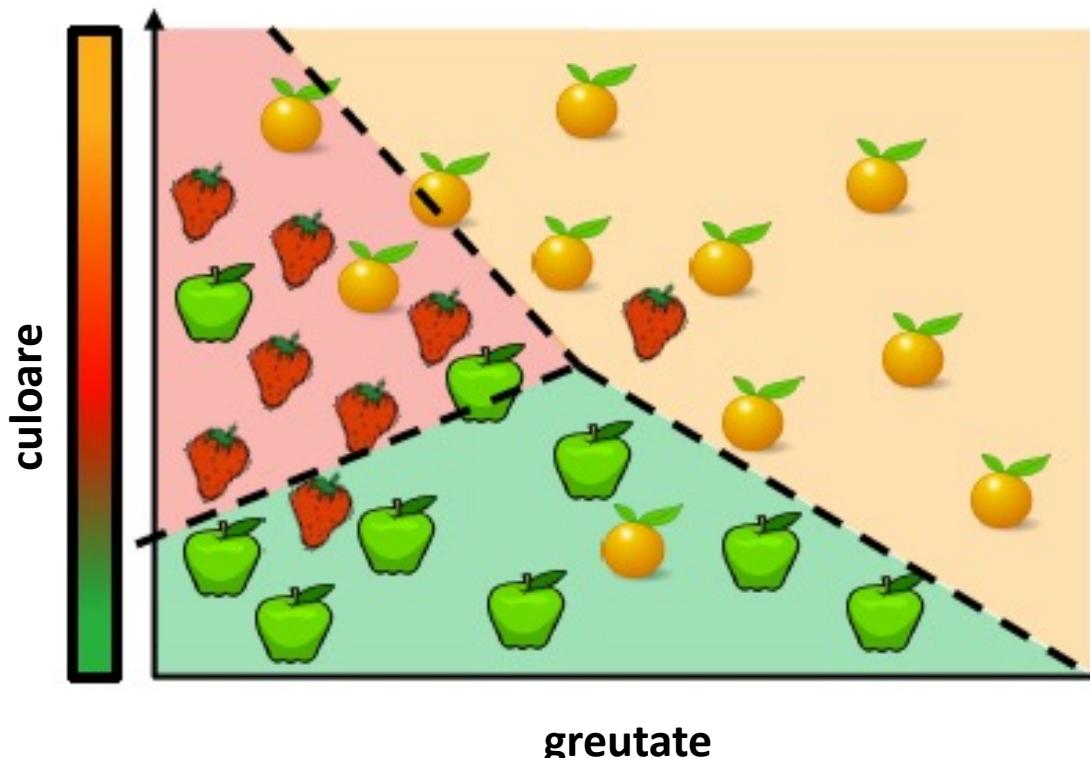


Etichete prezise			
Etichete reale	Apple	Orange	
Apple	7	0	2
Orange	1	8	2
Strawberry	1	1	6

Poate fi folosită pentru  
orice număr de clase

# Acuratețea

- Acuratețea unui model măsoară procentul de exemple din numărul total de exemple clasificate corecte de către model (elemente diagonala principală vs elemente toată matricea)



$$\text{Acuratețea} = 21/28 = 75\%$$

Apples	Oranges	Strawberries	
Apples	7	0	2
Oranges	1	8	2
Strawberries	1	1	6

# Probleme cu acuratețea

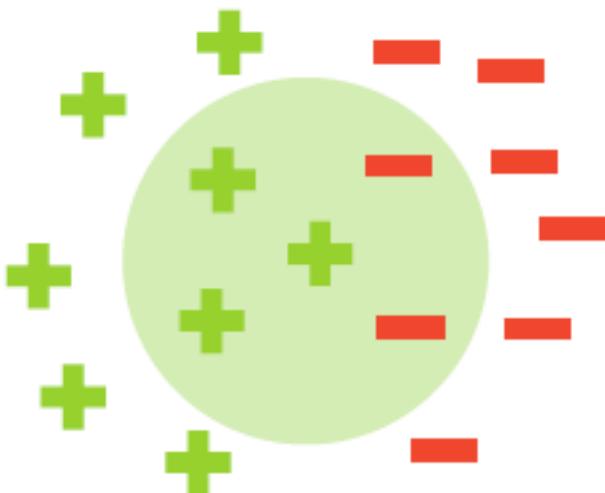
- Presupunem că vrem să antrenăm un clasificator pentru a detecta dacă un pacient are o boală extrem de rară
  - din 1000 de pacienți, numai 20 au o asemenea boală rară (2% grad de incidentă a bolii)
- Considerăm clasificatorul care prezice întotdeauna că pacientul este sănătos
  - are 98% acuratețe!!!
  - care este problema?

# Probleme cu acuratețea

- Acuratețea nu ține cont de câteva aspecte:
  - setul de date este puternic debalansat
    - 98% (pacienți sănătoși) vs. 2% (pacienți bolnavi)
  - cele două clase sunt conceptual diferite
    - o clasă este “pozitivă” – reprezintă clasa de care suntem interesați
    - o clasă este “negativă” – reprezintă starea implicită, nu suntem interesați
  - există două tipuri de erori, care diferă prin gravitatea lor
    - eroare de tipul 1 – “fals pozitiv” – un pacient este clasificat ca fiind bolnav deși este el sănătos
    - eroare de tipul 2 – “fals negativ” – un pacient este clasificat ca fiind sănătos deși el este bolnav
    - în funcție de problemă, una dintre erori poate fi mai gravă decât cealaltă

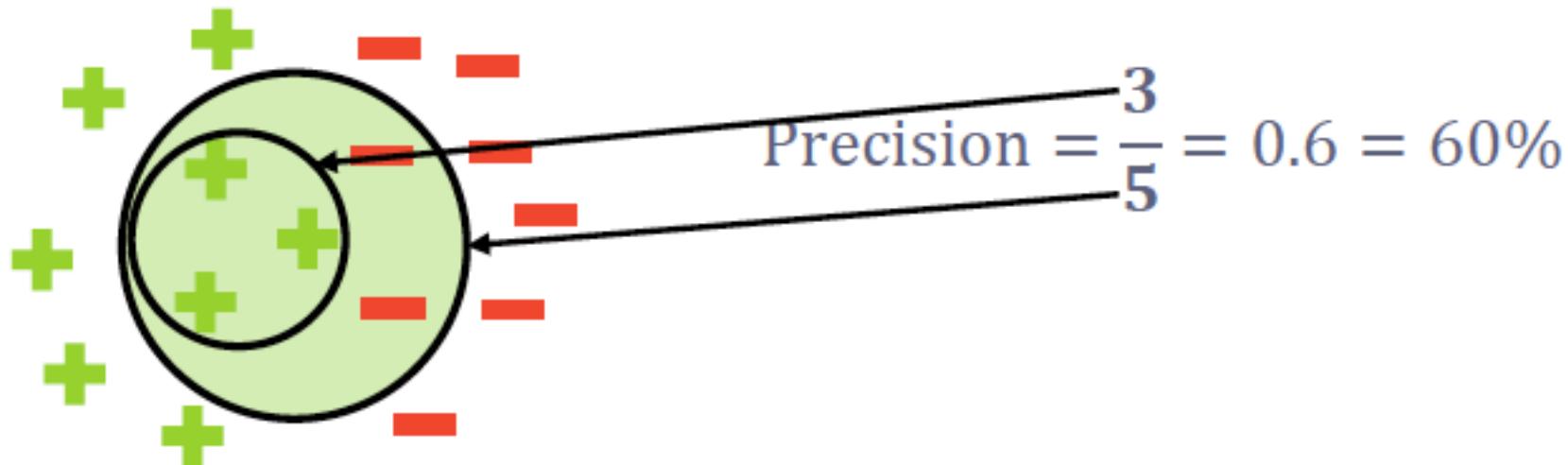
# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din itemi selectați de model (clasa lor prezisă este clasa pozitivă) care sunt relevanți (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din itemii relevanți care sunt selectați



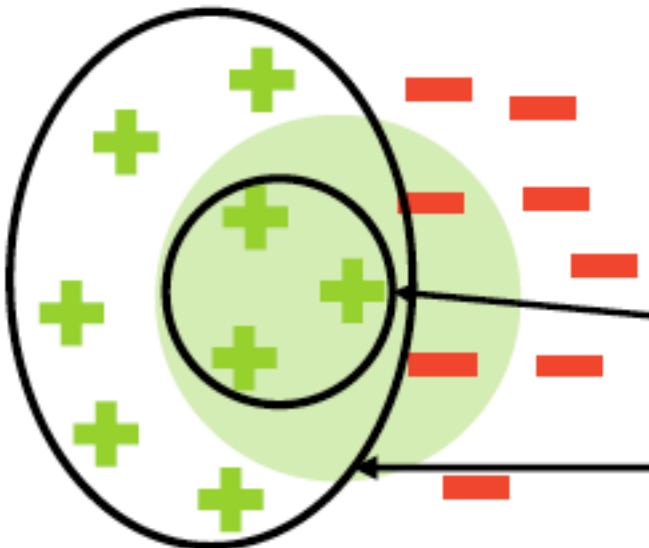
# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din **itemi selectați de model** (clasa lor prezisă este clasa pozitivă) **care sunt relevanți** (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din itemii relevanți care sunt selectați



# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din itemi selectați de model (clasa lor prezisă este clasa pozitivă) care sunt relevanți (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din **itemii relevanți care sunt selectați**



$$\text{Precision} = \frac{3}{5} = 0.6 = 60\%$$

$$\text{Recall} = \frac{3}{8} = 0.375 = 37.5\%$$

# Precizie și Recall

- Pentru problemele de clasificare în care avem o clasă pozitivă (de care suntem interesați), metricile de precizie și recall sunt mult mai informative decât acuratețea.
- **Precizia:** procentul din itemi selectați de model (clasa lor prezisă este clasa pozitivă) care sunt relevanți (fac parte din clasa pozitivă, cea care ne interesează)
- **Recall:** procentul din **itemii relevanți care sunt selectați**
- **Scorul  $F_1$**  : media armonică pentru **Precizie și Recall**

# Acuratețe, Precizie și Recall

Etichete reale

		Etichete prezise	
		+	-
+	+	TP	FN
	-	FP	TN
Matrice de confuzie			

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

TP – true positive

FN – false negative

FP – false positive

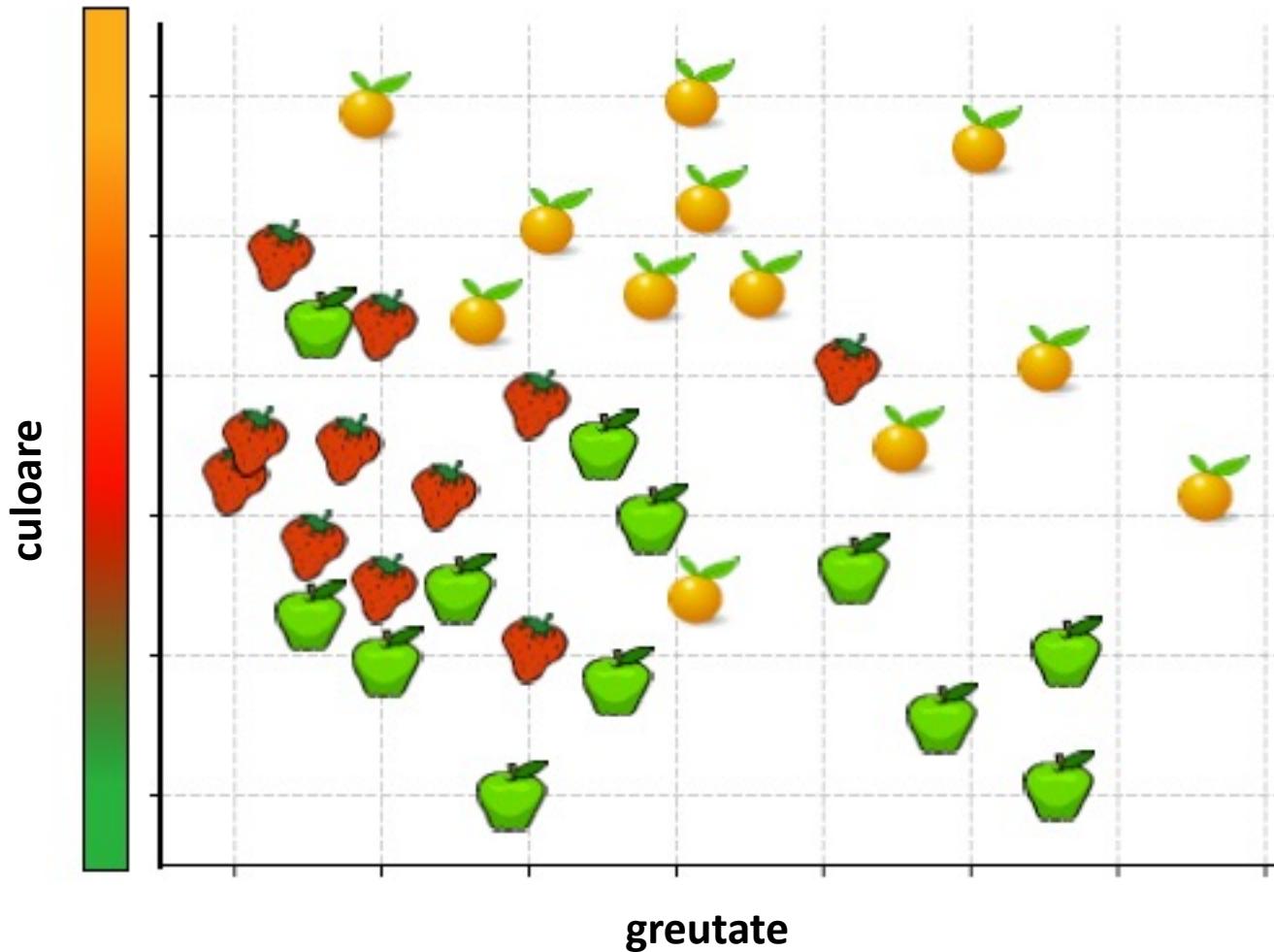
TN – true negative

# Modelul celor mai apropiati k-vecini (k-nearest neighbors)

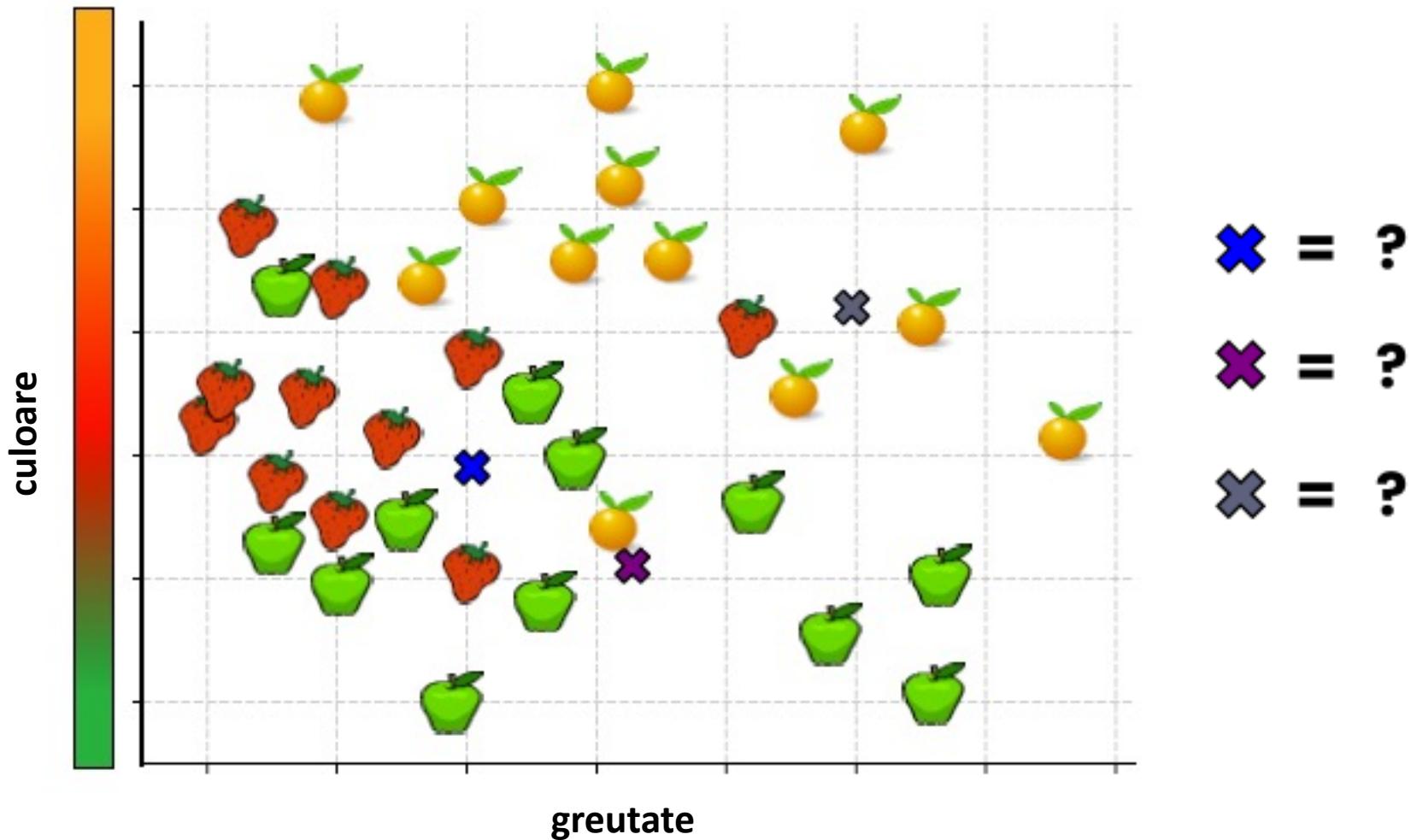
# Modelul celor mai apropiati k-vecini (kNN – k nearest neighbors)

- Modelul kNN prezice eticheta unui exemplu test ca fiind eticheta predominantă ale celor mai apropiate  $k$  exemple de antrenare (cei mai apropiati  $k$  vecini) din spațiul caracteristicilor (*feature space*).
- Învățare supervizată (avem etichete).
- Modelul kNN nu învață explicit un model:
  - în schimb, memorează datele de antrenare și le folosește în realizarea de predicții
- $k$  = numărul de vecini este hiperparametru (stabilit dinainte)

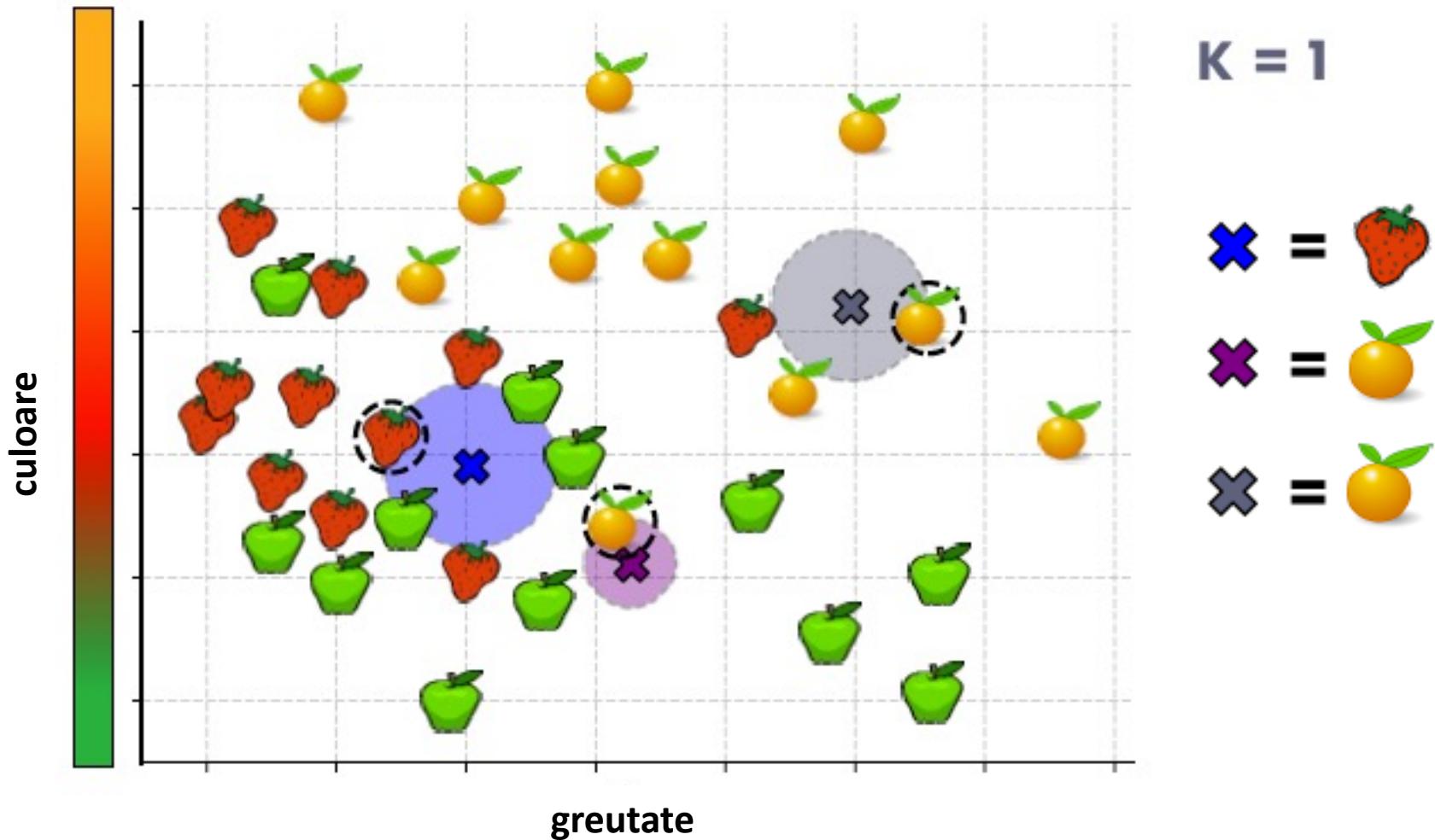
# Modelul celor mai apropiati k-vecini (kNN)



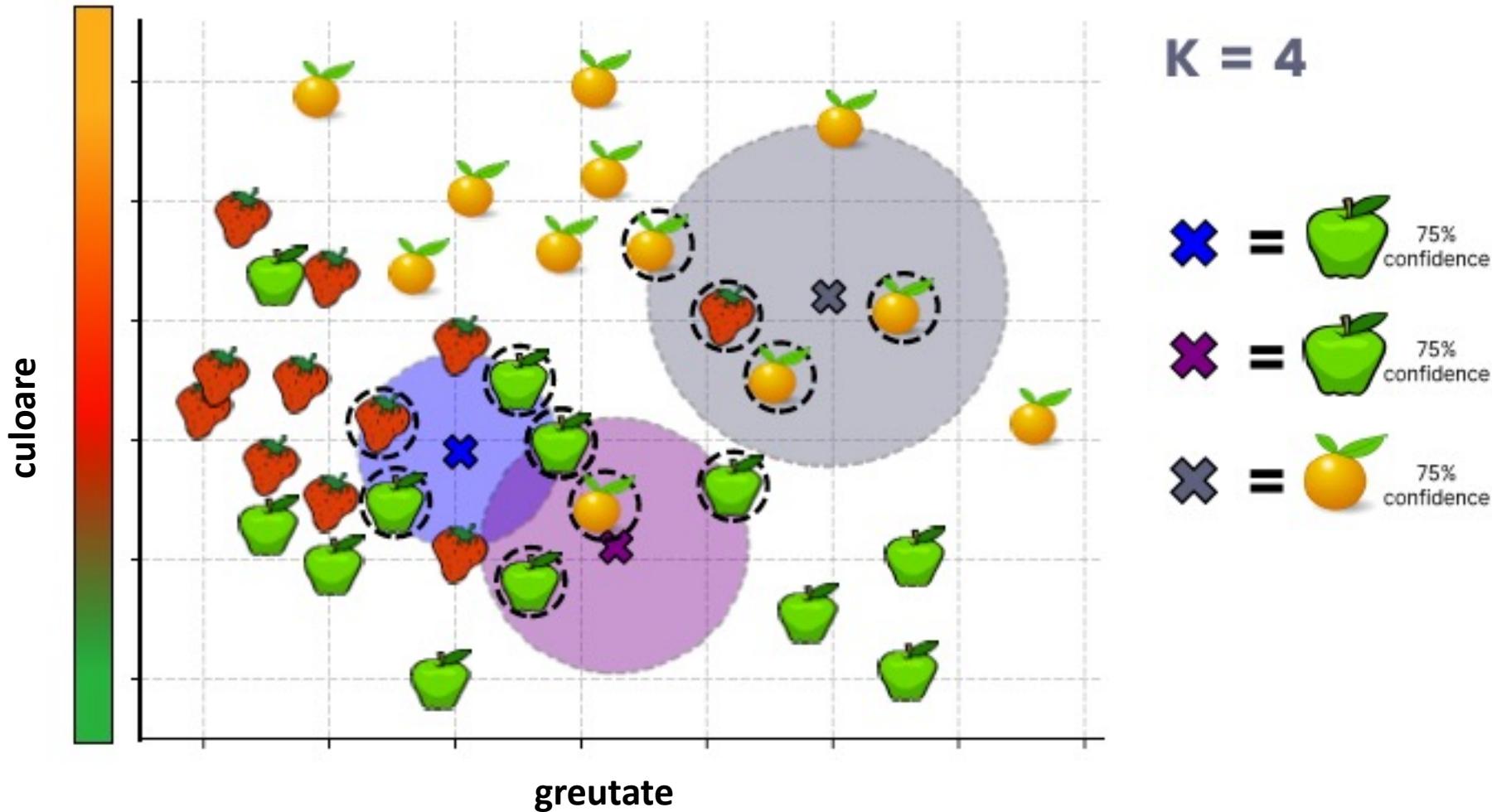
# Modelul celor mai apropiati k-vecini (kNN)



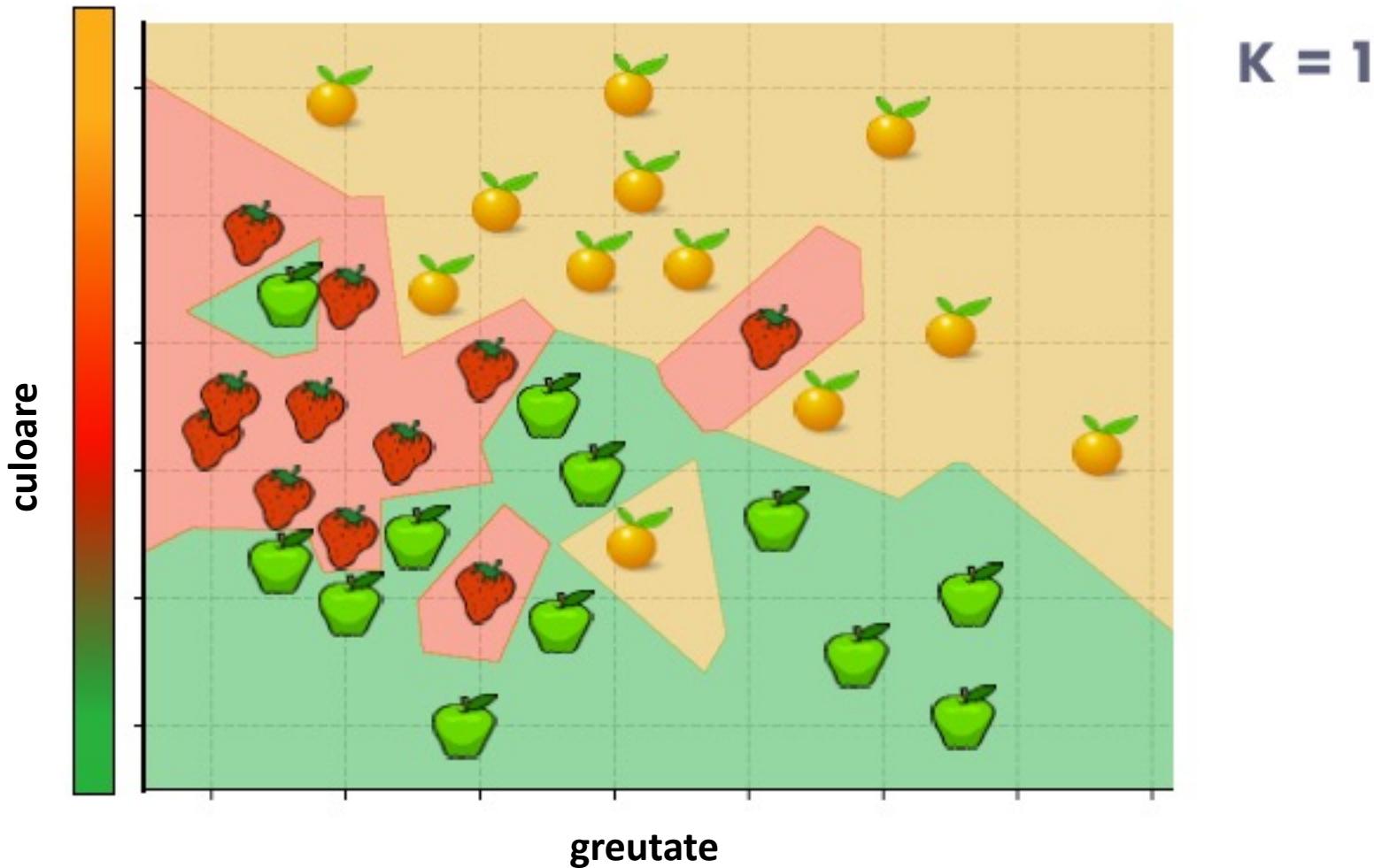
# Modelul celor mai apropiati k-vecini (kNN)



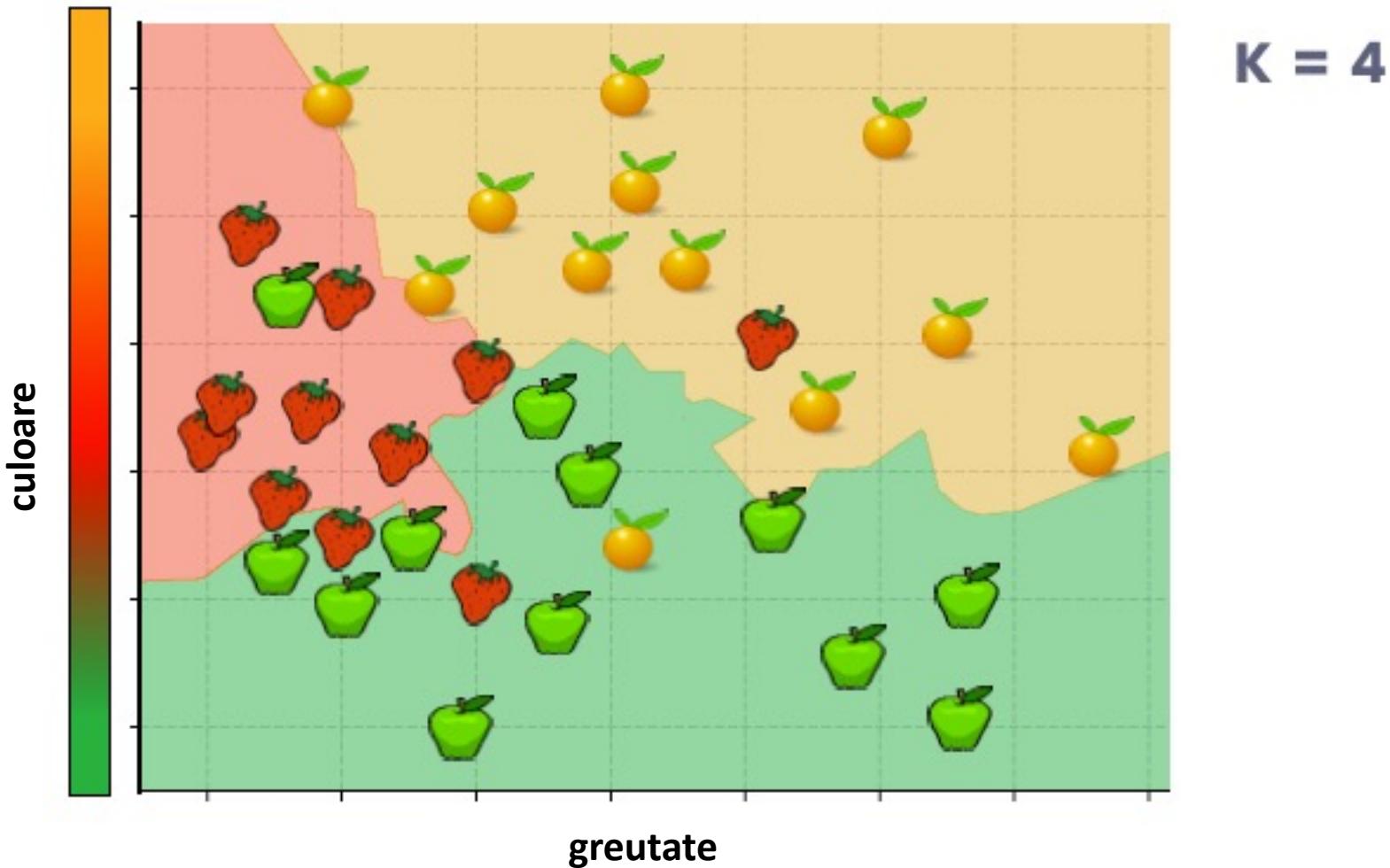
# Modelul celor mai apropiati k-vecini (kNN)



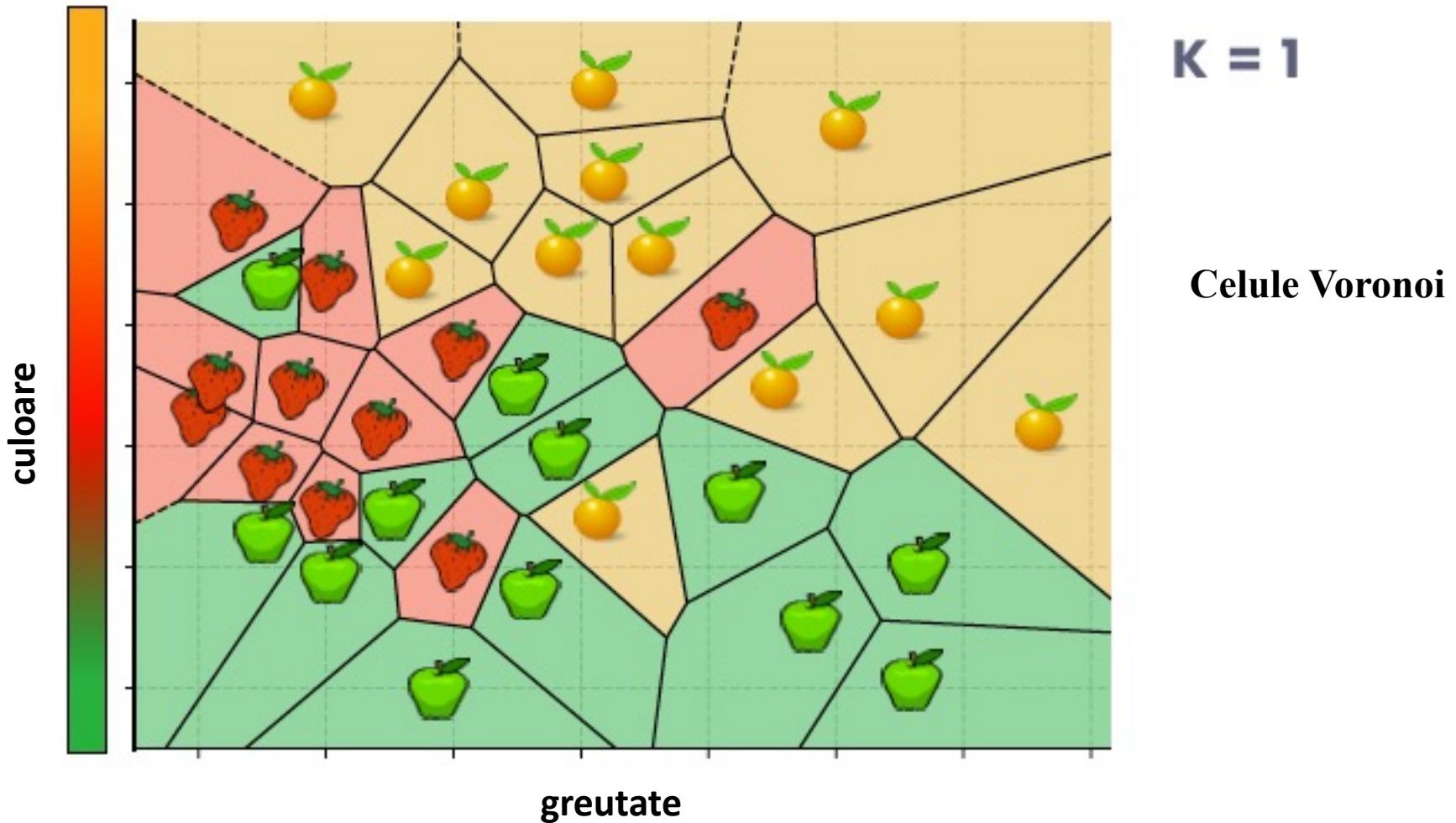
# Graniță de separare



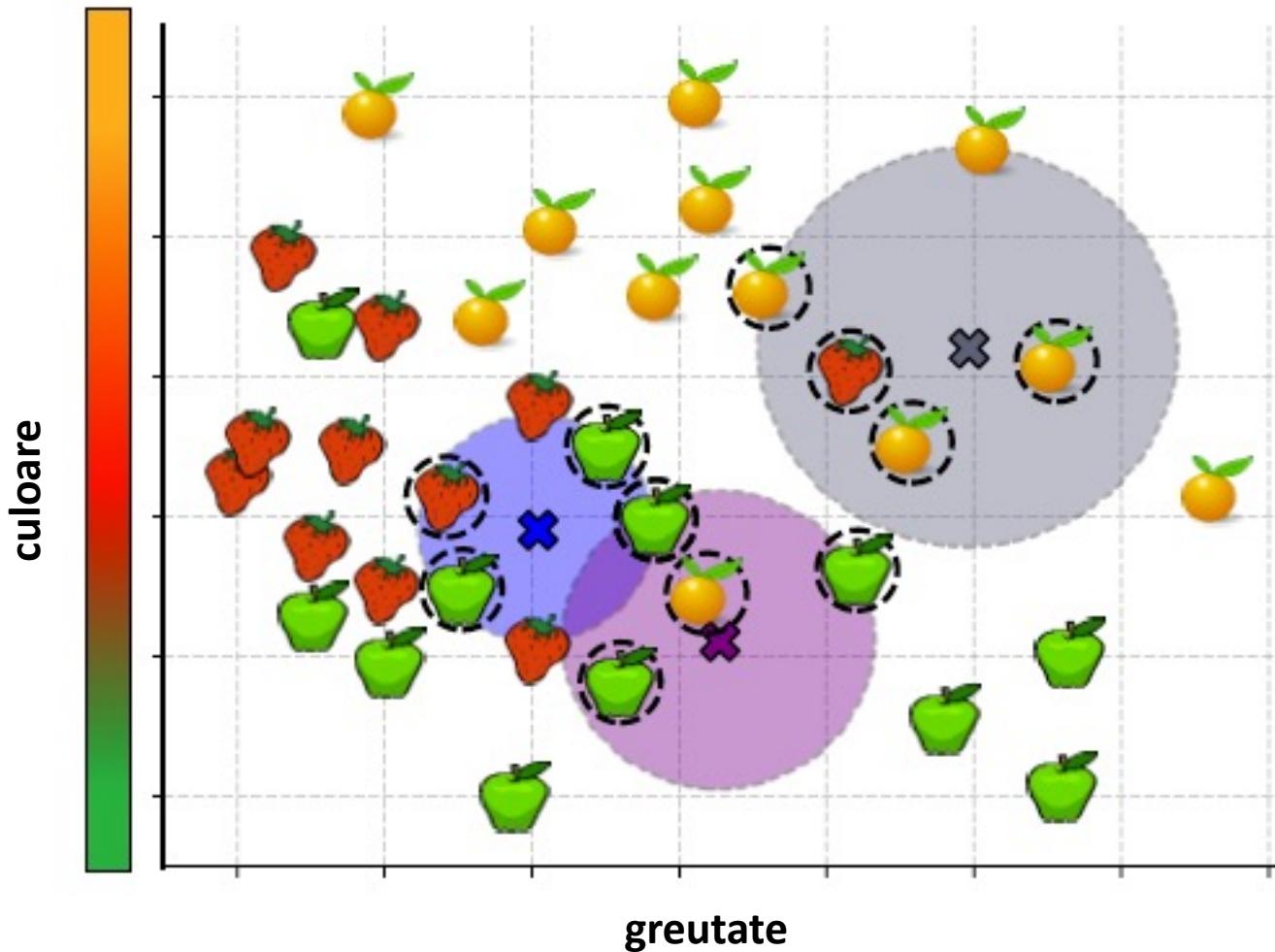
# Graniță de separare



# Graniță de separare



# Distanțe folosite de kNN



**K = 4**

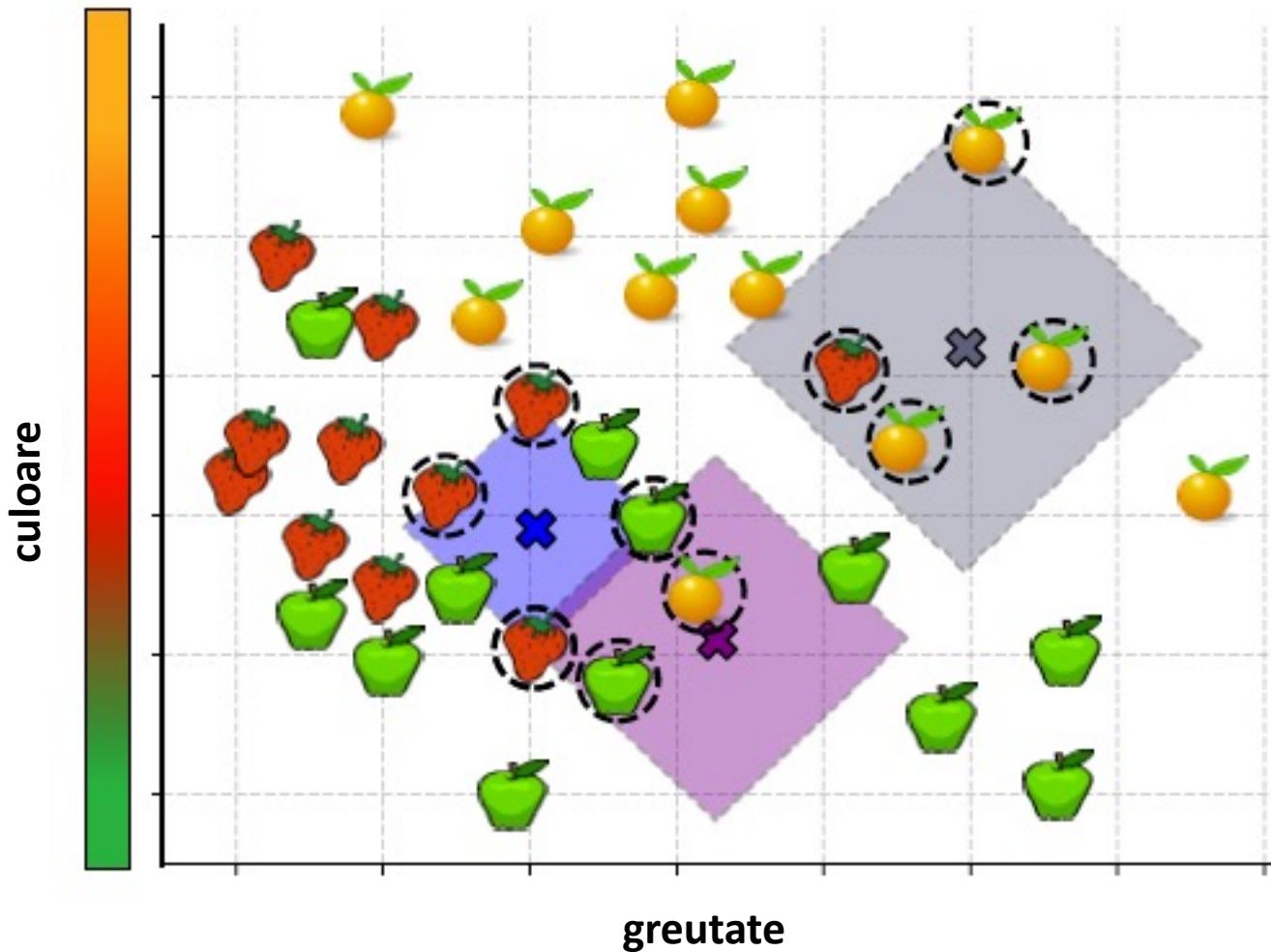
✗ = 75% confidence

✗ = 75% confidence

✗ = 75% confidence

**Distanță  
Euclidiană ( $L_2$ )**

# Distanțe folosite de kNN



**K = 4**

= 75% confidence

= 50% confidence

= 75% confidence

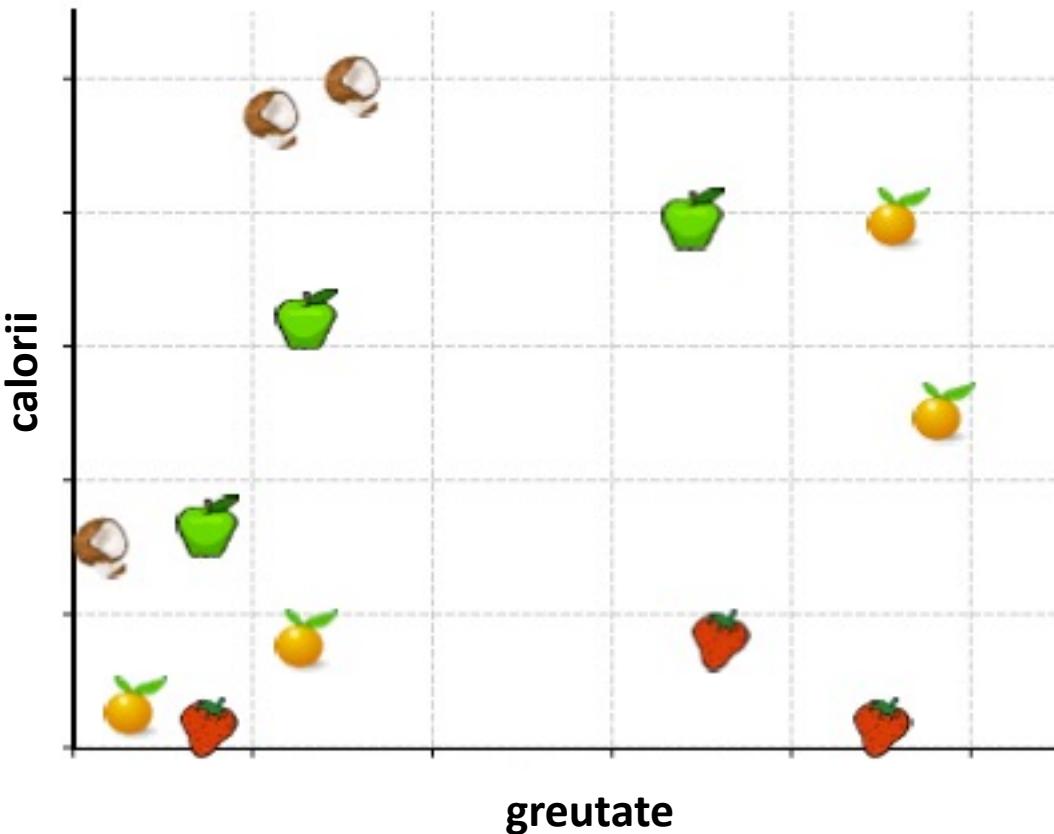
**Distanță  
Manhattan ( $L_1$ )**

# Distanță

- O funcție  $d: X \times X \rightarrow [0, \infty)$  se numește distanță, dacă satisface pentru orice două elemente  $x$  și  $y$  din  $X$  condițiile:
  1.  $d(x,y) \geq 0$
  2.  $d(x,y) = 0 \Leftrightarrow x = y$
  3.  $d(x,y) = d(y,x)$
  4.  $d(x,z) \leq d(x,y) + d(y,z)$
- Performanța modelului kNN depinde foarte mult de tipul distanței folosite în calculul celor mai apropiati vecini
  - distanța trebuie să reflecte natura problemei (ar trebui să ia valori mici pentru itemi cunoscuți ca similari și valori mari pentru itemi cunoscuți ca diferenți)

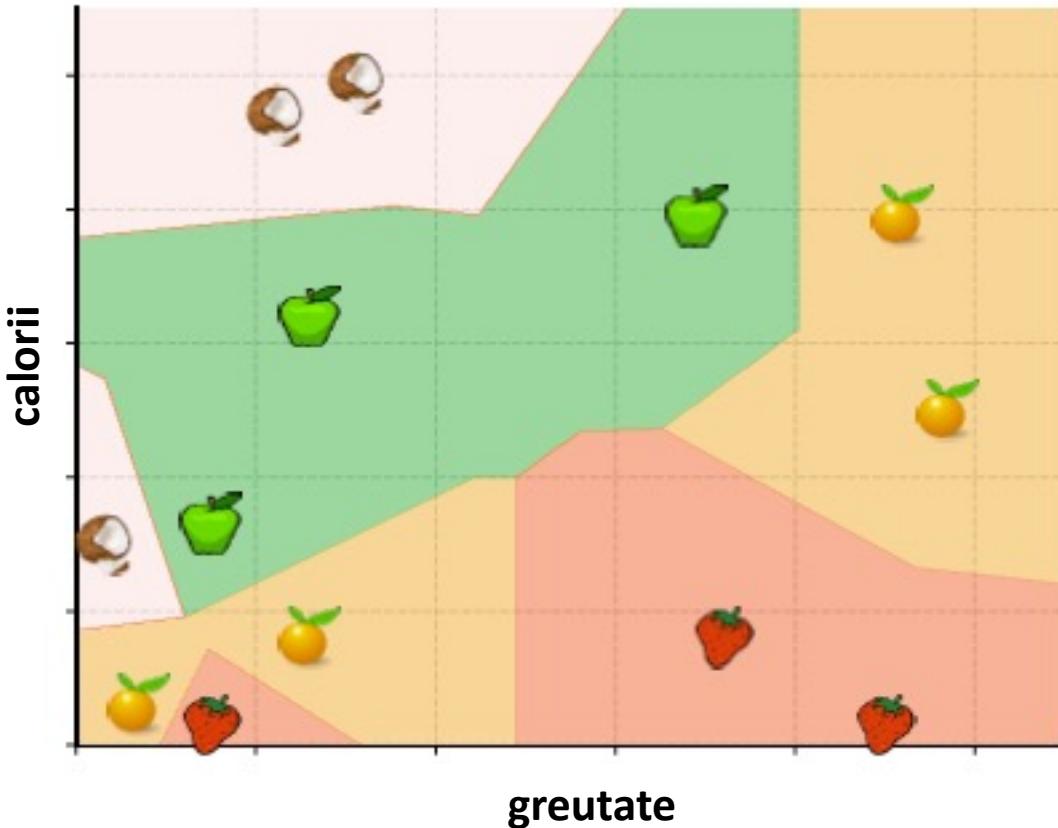
# Distanțe folosite de kNN

- Problemă de clasificare: prezicem tipul de fruct pe baza a două caracteristici, greutatea și cantitatea de calorii conținută



# Distanțe folosite de kNN

- Problemă de clasificare: prezicem tipul de fruct pe baza a două caracteristici, greutatea și cantitatea de calorii conținută

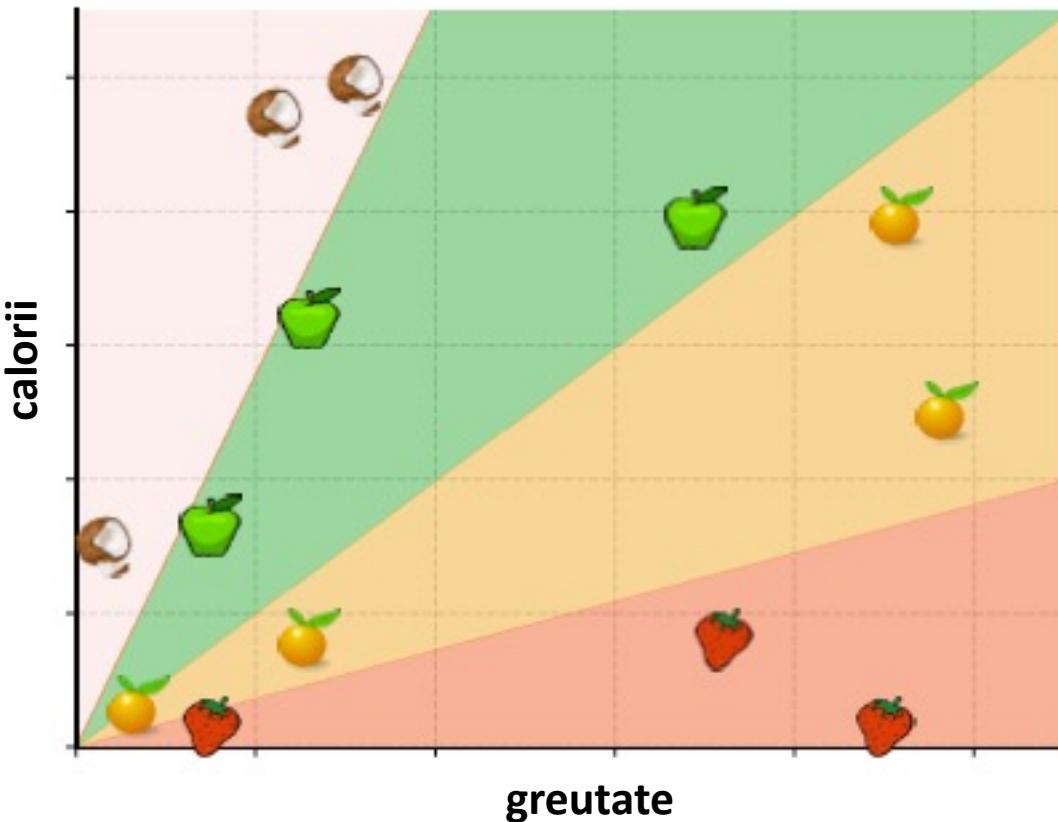


**Distanța  
Euclidiană ( $L_2$ )**

Distanța Euclidină modelează bine relația dintre caracteristici și etichete?

# Distanțe folosite de kNN

- Problemă de clasificare: prezicem tipul de fruct pe baza a două caracteristici, greutatea și cantitatea de calorii conținută



## Distanța Cosinus

Distanța Cosinus măsoară unghiul dintre vectorii de caracteristici.

Itemi similari au rapoarte similare între valorile caracteristicilor.

# Exemple de Distanțe

Euclidean distance:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

$L_2$

Manhattan distance:

$$d(\vec{x}, \vec{y}) = \sum_i |x_i - y_i|$$

$L_1$

Chebyshev distance:

$$d(\vec{x}, \vec{y}) = \max_i (|x_i - y_i|)$$

$L_\infty$

Minkowski distance:

$$d(\vec{x}, \vec{y}) = \left( \sum_i |x_i - y_i|^p \right)^{1/p}$$

$L_p$

Cosine distance:

$$d(\vec{x}, \vec{y}) = 1 - \cos \theta = 1 - \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

# Distanța Euclidiană vs Manhattan

Euclidean distance:

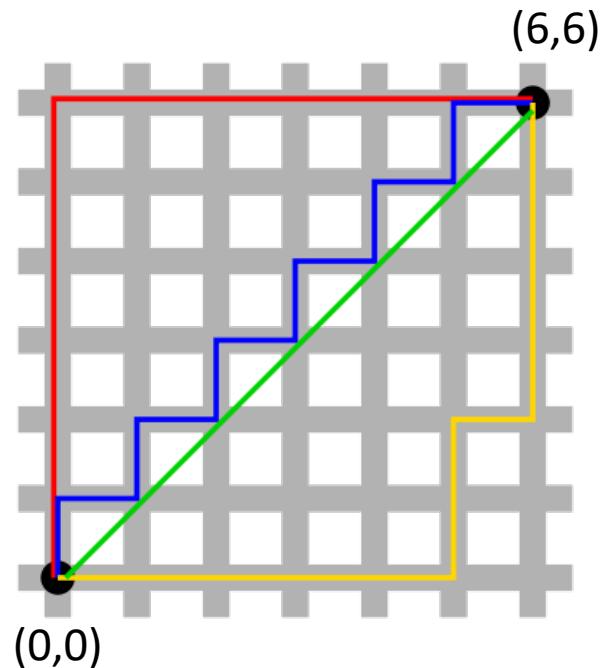
$$d(\vec{x}, \vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

distanța Euclidiană = lungimea drumului cel mai scurt = **lungimea drumului verde** =  $6\sqrt{2}$

Manhattan distance:

$$d(\vec{x}, \vec{y}) = \sum_i |x_i - y_i|$$

distanța Manhattan = suma absolută a diferențelor coordonateleor carteziene = **lungimea drumului roșu** = **lungimea drumului albastru** = **lungimea drumului galben** = 12



# Cuvinte cheie

Învățare automată

Etichetă

Caracteristică

Vector de caracteristici

Antrenare

Inferență

Model

Ipoteză

Algoritm

Supervizare

Clasificare

Regresie

Nesupervizare

Semi-supervizare

Acuratețe

Precizie

Recall

Matrice de confuzie

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 3

# Schimbare de orar

De săptămâna viitoare facem permutarea între ASC și IA

Cursul de Inteligență Artificială va fi joi, în intervalul 10-12,  
cursul de Arhitectura Sistemelor de Calcul va fi joi, în  
intervalul 12-14.

# Recapitulare – cursul trecut

1. Învățarea Automată: definiție și terminologie
2. Paradigme de învățare
3. Metrici de măsurare a performanței pentru clasificare

# Terminologie în învățarea automată

- **etichetă** (label) → ce încercăm să prezicem
- **caracteristică** (feature) → proprietate măsurabilă a unui punct din exemplele din date
- **model** (ipoteză) → relația dintre caracteristici și etichete
- **antrenare** → stabilirea relației pe baza unei multimi de puncte
- **inferență** → realizarea de predicții pe date necunoscute
- **algoritm** → definește cum se face învățarea, poate fi restricționat la o anumită mulțime de ipoteze, fie prin alegerea unui anumit spațiu, fie pe baza hiperparametrilor

# Paradigme de învățare

- **Învățare Supervizată** → prezicem o etichetă
  - etichetă continuă → **Regresie**
  - etichetă discretă → **Clasificare**
- **Învățarea Nesupervizată** → descoperirea de structuri în date
  - gruparea itemilor similari → **Clusterizare**
  - pattern-uri frecvente → **Căutarea regulilor de asociere**
- **Învățare pe bază de recompense** → fără etichete, ci numai recompense sau penalități pentru realizarea unor acțiuni
- **Învățare semi-supervizată** → câteva date etichetate, multe neetichetate
- **Învățarea prin transfer** → folosirea unui model antrenat pe o problemă pentru învățarea mai rapidă pentru altă problemă

# Acuratețe, Precizie și Recall

Etichete reale

		Etichete prezise	
		+	-
+	+	TP	FN
	-	FP	TN
Matrice de confuzie			

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

TP – true positive

FN – false negative

FP – false positive

TN – true negative

# Cuvinte cheie

Învățare automată

Etichetă

Caracteristică

Vector de caracteristici

Antrenare

Inferență

Model

Ipoteză

Algoritm

Supervizare

Clasificare

Regresie

Nesupervizare

Semi-supervizare

Acuratețe

Precizie

Recall

Matrice de confuzie

# Cuprinsul cursului de azi

1. Modelul celor mai apropiati k vecini (k-nearest neighbors) – continuare
2. Normalizarea caracteristicilor
3. Clasificatorul naïve Bayes
4. Evaluarea performanței unui model

# Modelul celor mai apropiati k-vecini (k-nearest neighbors)

# Modelul celor mai apropiati k-vecini (kNN)

- Modelul kNN prezice eticheta unui exemplu test ca fiind eticheta predominantă ale celor mai apropiate  $k$  exemple de antrenare (cei mai apropiati  $k$  vecini) din spațiul caracteristicilor (*feature space*).
- Învățare supervizată (avem etichete).
- Modelul kNN nu învață explicit un model:
  - în schimb, memorează datele de antrenare și le folosește în realizarea de predicții
- $k$  = numărul de vecini este hiperparametru

# Laborator

- kNN aplicat pe o parte din setul de date MNIST (cifre scrise de mâňă). Setul original conține 50.000 de exemple de antrenare, 10.000 de exemple de testare.

# Vizualizare date de antrenare

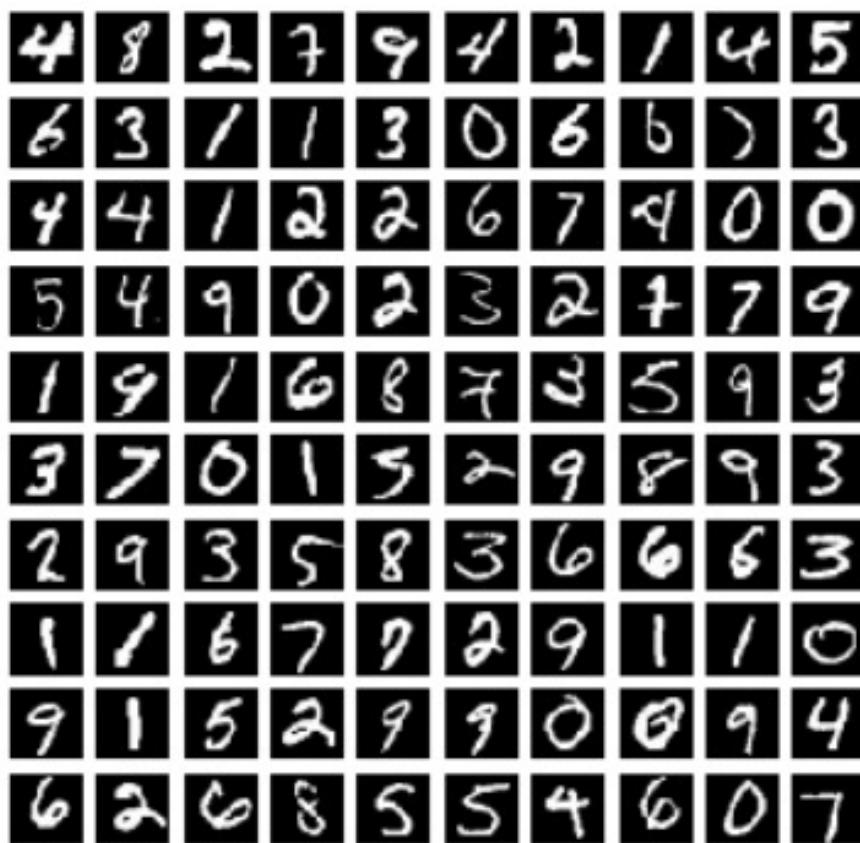
- kNN aplicat pe o parte din setul de date MNIST (cifre scrise de mâňă). Setul original conține 50.000 de exemple de antrenare, 10.000 de exemple de testare.

```
1 #plot the first 100 training images with their labels in a 10 x 10 subplot
2 nbImages = 10
3 plt.figure(figsize=(5,5))
4 for i in range(nbImages**2):
5     plt.subplot(nbImages,nbImages,i+1)
6     plt.axis('off')
7     plt.imshow(np.reshape(train_images[i,:],(28,28)),cmap = "gray")
8 plt.show()
9 labels_nbImages = train_labels[:nbImages**2]
10 print(np.reshape(labels_nbImages,(nbImages,nbImages)))
```

# Vizualizare date de antrenare

- kNN aplicat pe o parte din setul de date MNIST (cifre scrise de mâňă). Setul original conține 50.000 de exemple de antrenare, 10.000 de exemple de testare.

Primele 100 exemple de antrenare



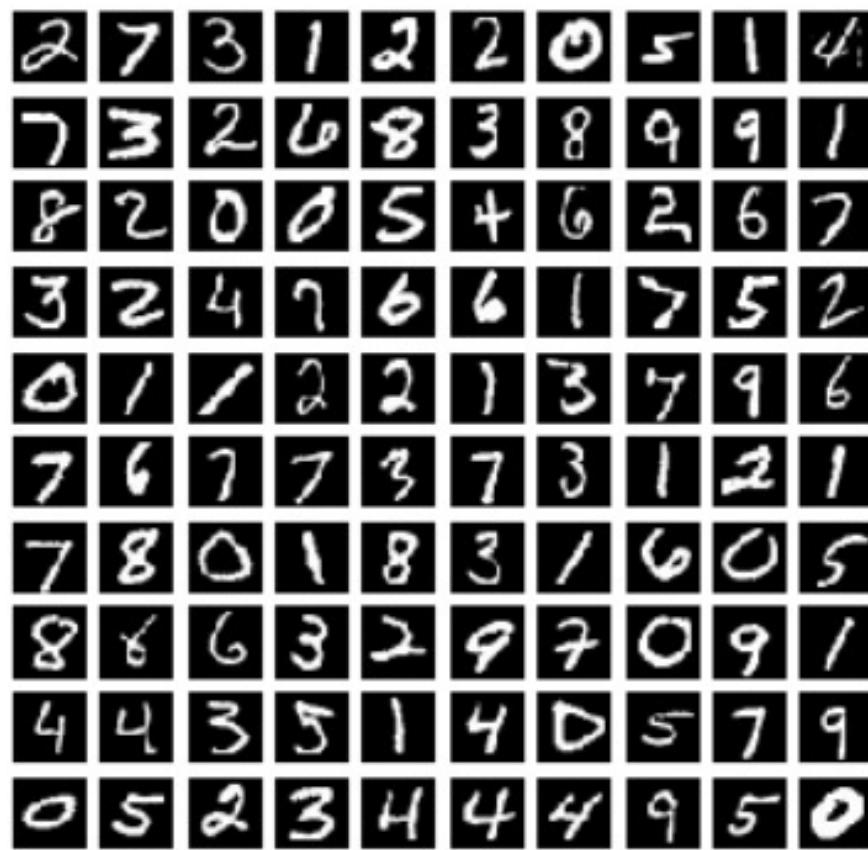
Etichetele corespunzătoare primelor 100 exemple de antrenare

```
[ [4 8 2 7 9 4 2 1 4 5]
  [6 3 1 1 3 0 6 6 7 3]
  [4 4 1 2 2 6 7 4 0 0]
  [5 4 9 0 2 3 2 1 7 9]
  [1 9 1 6 8 7 3 5 9 3]
  [3 7 0 1 5 2 9 8 9 3]
  [2 9 3 5 8 3 6 6 6 3]
  [1 1 6 7 9 2 9 1 1 0]
  [9 1 5 2 9 9 0 0 9 4]
  [6 2 6 8 5 5 4 6 0 7] ]
```

# Vizualizare date de testare

- kNN aplicat pe o parte din setul de date MNIST (cifre scrise de mâna). Setul original conține 50.000 de exemple de antrenare, 10.000 de exemple de testare.

Primele 100 exemple de testare



Etichetele corespunzătoare primelor  
100 exemple de testare

```
[ [2 7 3 1 2 2 0 5 1 4]
  [7 3 2 6 8 3 8 9 9 1]
  [8 2 0 0 5 4 6 2 6 7]
  [3 2 4 7 6 6 1 7 5 2]
  [0 1 1 2 2 1 3 7 9 6]
  [7 6 7 7 3 7 3 1 2 1]
  [7 8 0 1 8 3 1 6 0 5]
  [8 6 6 3 2 9 7 0 9 1]
  [4 4 3 5 1 4 0 5 7 9]
  [0 5 2 3 4 4 4 9 5 0] ]
```

# Vizualizare 7-NN

- Aplicăm 7-NN pentru primul exemplu de testare

**Exemplu testare**



# Vizualizare 7-NN

- Aplicăm 7-NN pentru primul exemplu de testare

**Exemplu testare**



**Primul vecin** →



Cel mai apropiat exemplu din setul de antrenare considerând distanța Euclidiană calculată pe cele 784 de componente

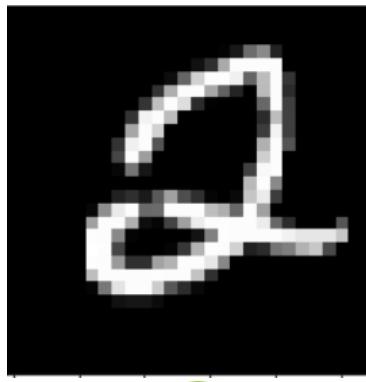
# Vizualizare 7-NN

- Aplicăm 7-NN pentru primul exemplu de testare

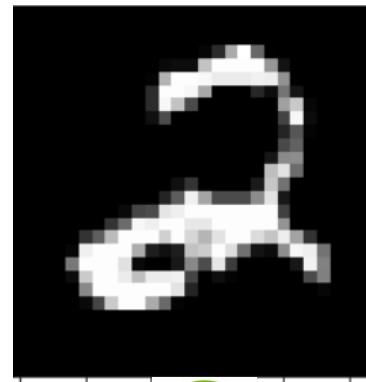
Exemplu testare



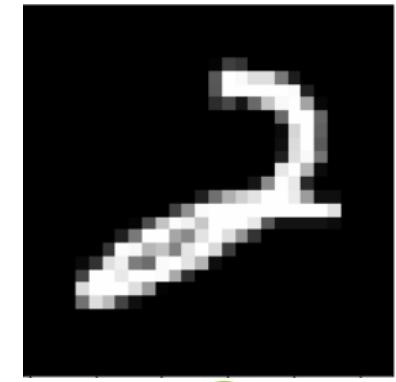
Primul vecin



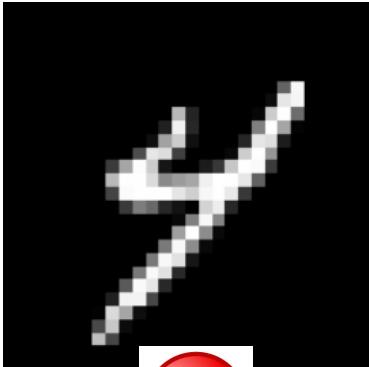
Al doilea vecin



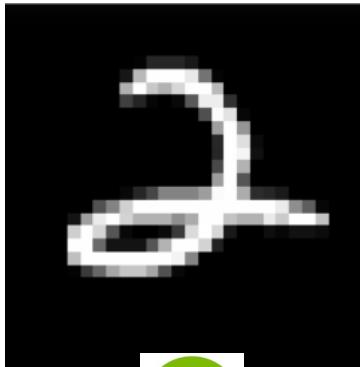
Al treilea vecin



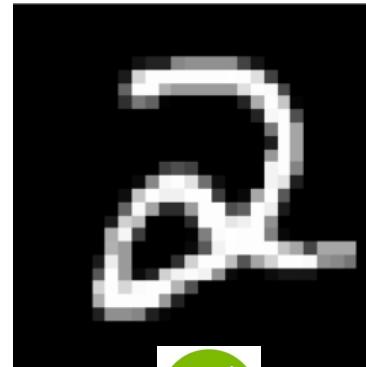
Al patrulea vecin



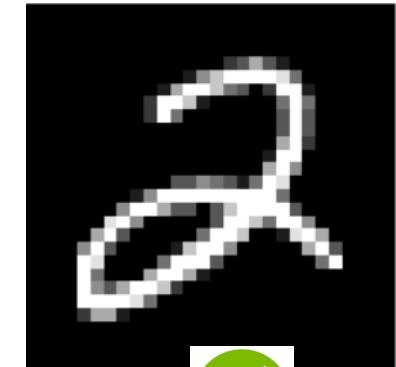
Al cincilea vecin



Al șaselea vecin



Al șaptelea vecin



# Laborator 3 – Matrice confuzie

		Eticheta prezisa									
		0	1	2	3	4	5	6	7	8	9
Eticheta reala	0	[ [ 51	0	0	0	0	1	1	0	0	0 ]
	1	[ 0	52	0	0	0	0	0	0	0	0 ]
	2	[ 1	6	47	1	0	0	1	2	0	0 ]
	3	[ 0	0	0	51	0	1	0	0	0	1 ]
	4	[ 0	0	0	0	44	0	0	0	0	2 ]
	5	[ 2	1	1	6	0	40	1	0	0	1 ]
	6	[ 0	0	0	0	0	1	47	0	0	0 ]
	7	[ 1	2	0	0	1	0	0	46	0	0 ]
	8	[ 1	0	2	2	1	1	1	1	36	1 ]
	9	[ 0	0	1	1	3	1	0	1	0	35 ] ]

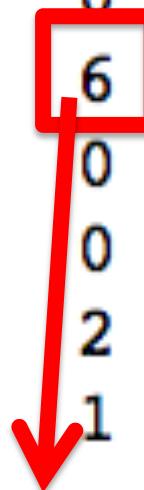
-nu e simetrică

-numărul de exemple clasificate corect – pe diagonala principală

-numărul de exemple clasificate greșit – restul matricei

# Laborator 3 – Matrice confuzie

	0	1	2	3	4	5	6	7	8	9
0	[ [ 51	0	0	0	0	1	1	0	0	0 ]
1	[ 0	52	0	0	0	0	0	0	0	0 ]
2	[ 1	6	47	1	0	0	1	2	0	0 ]
3	[ 0	0	0	51	0	1	0	0	0	1 ]
4	[ 0	0	0	0	44	0	0	0	0	2 ]
5	[ 2	1	1	6	0	40	1	0	0	1 ]
6	[ 0	0	0	0	0	1	47	0	0	0 ]
7	[ 1	2	0	0	1	0	0	46	0	0 ]
8	[ 1	0	2	2	1	1	1	1	36	1 ]
9	[ 0	0	1	1	3	1	0	1	0	35 ] ]



Numărul de exemple de testare cu cifra  
5 care au fost clasificate drept cifra 3

# Laborator 3 – Exemplu de testare cu cifra 5 clasificate drept cifra 3

Exemplu test



Vecinul 1



Vecinul 2



Vecinul 3



Exemplu test



Vecinul 1



Vecinul 2



Vecinul 3



Exemplu test



Vecinul 1



Vecinul 2



Vecinul 3



# Laborator 3 – Exemplu de testare cu cifra 5 clasificate drept cifra 3

Exemplu test



Vecinul 1



Vecinul 2



Vecinul 3



Exemplu test



Vecinul 1



Vecinul 2



Vecinul 3



Exemplu test



Vecinul 1



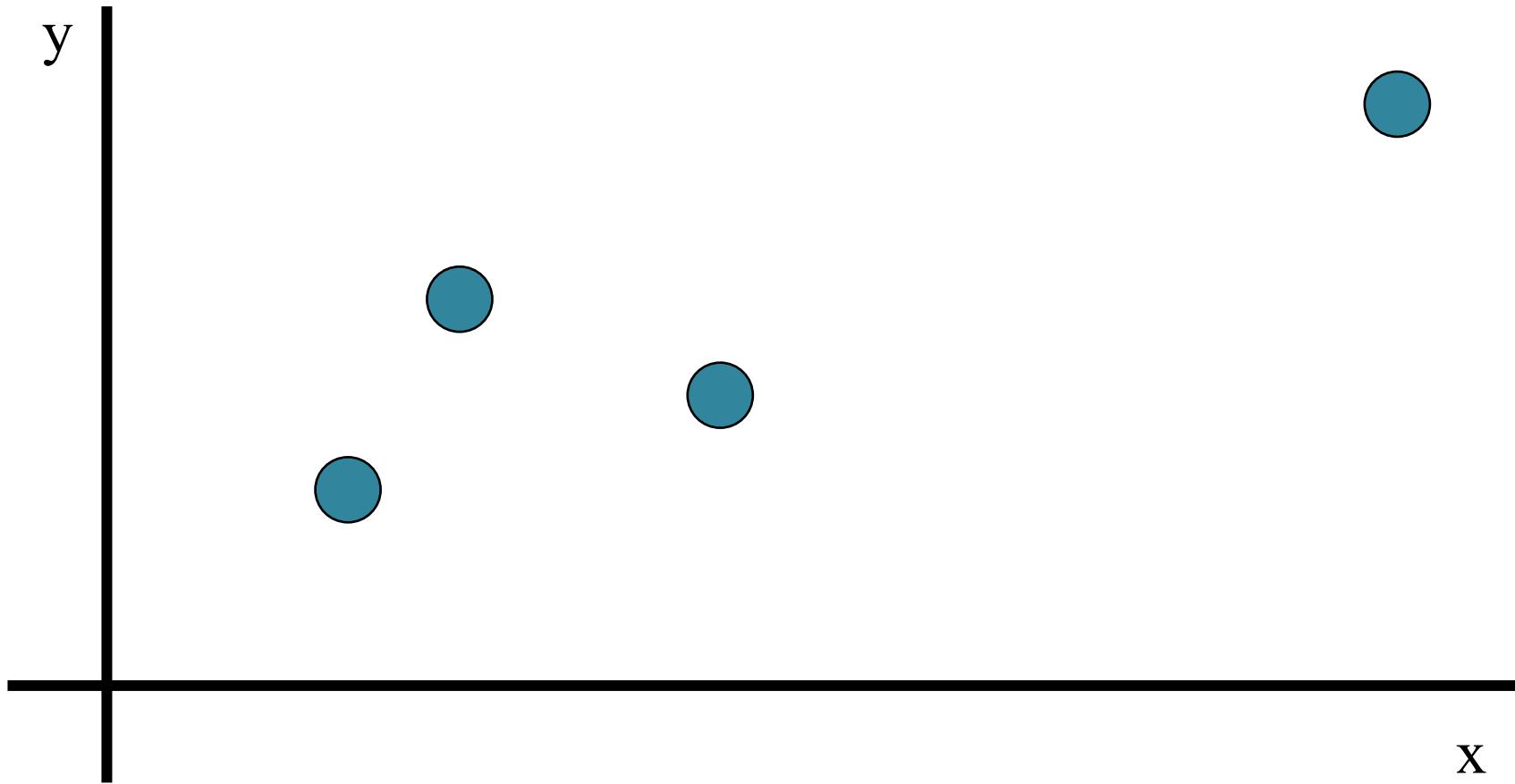
Vecinul 2



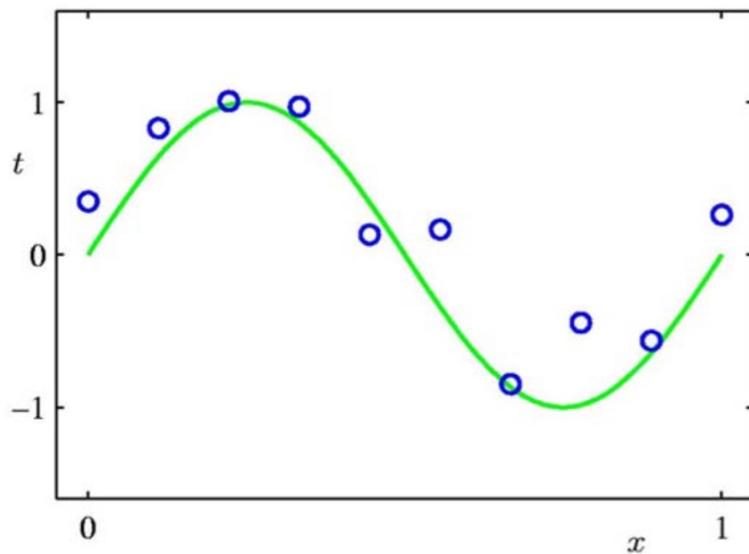
Vecinul 3



# Modelul celor mai apropiati k-vecini pentru probleme de regresie



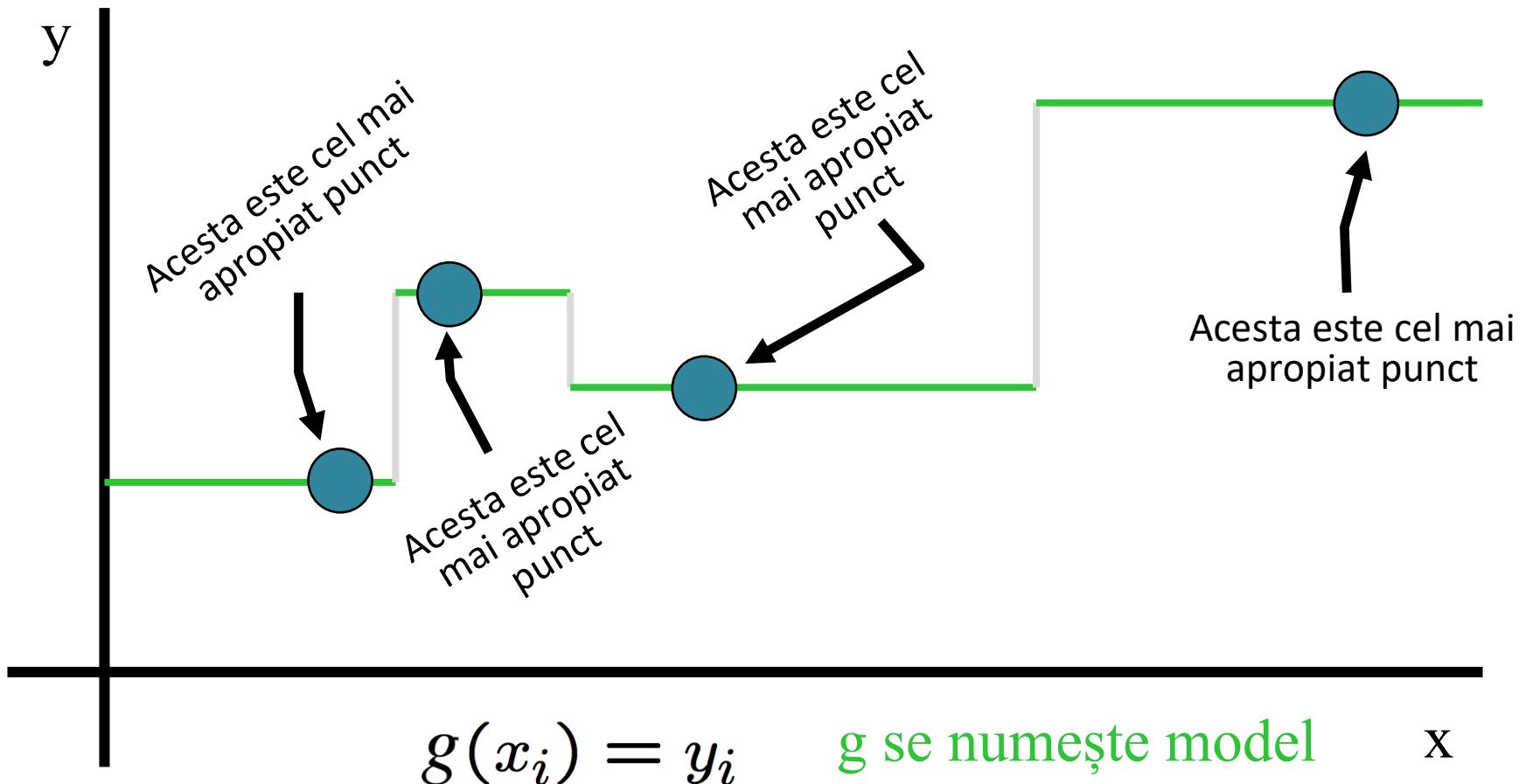
# Regresie din exemple etichetate



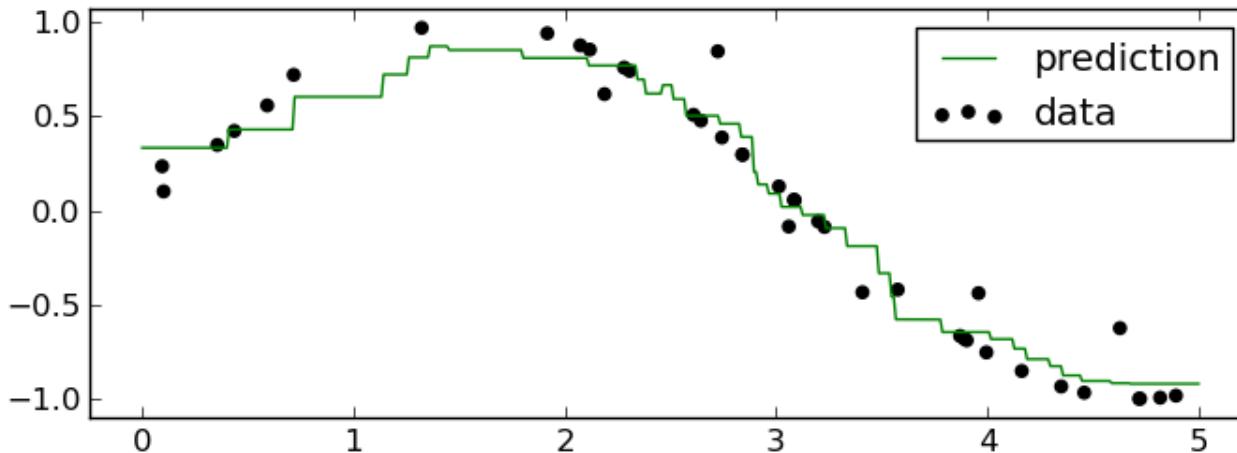
- Presupunem că avem un set de  $N$  exemple de antrenare:  
 $(x_1, \dots, x_N)$  și  $(y_1, \dots, y_N)$ ,  $x_i, y_i \in \mathbb{R}$
- Problema regresiei constă în estimarea funcției  $g(x)$  a.î.:  
$$g(x_i) = y_i \quad g \text{ se numește model}$$

# Modelul celor mai apropiati k-vecini pentru probleme de regresie

**K = 1**



# Modelul celor mai apropiati k-vecini pentru probleme de regresie



Algoritmul de regresie bazat pe cei mai apropiati k-vecini:

- 1) Pentru fiecare exemplu de test  $x$ , găsim cei mai apropiati  $k$  vecini și etichetele lor
- 2) Predictia este media etichetelor celor  $k$  vecini

$$f(x) = \frac{1}{K} \sum_{i=1}^K y_i$$

# Modelul celor mai apropiati k-vecini (kNN)

## Avantaje:

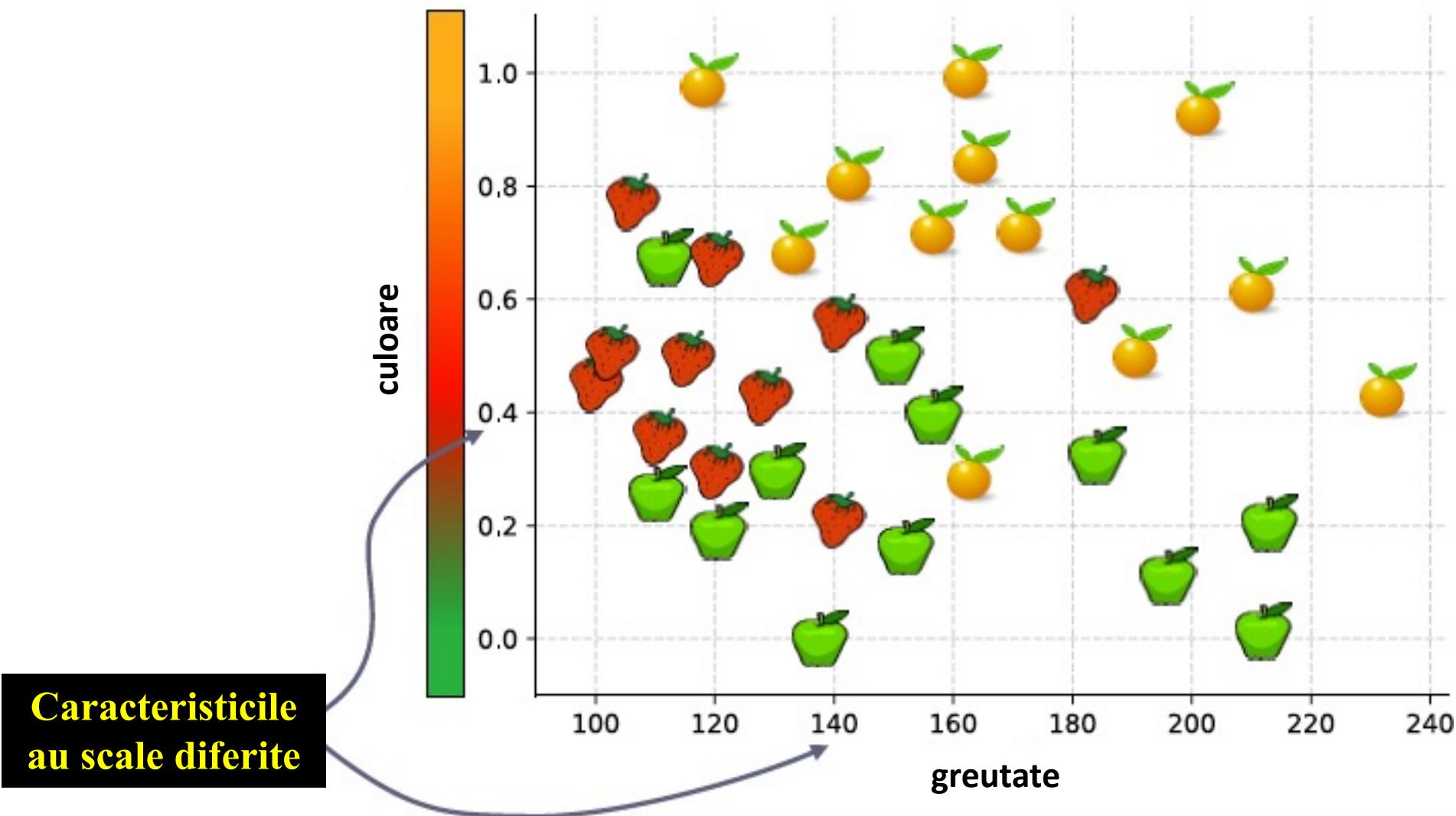
- intuitiv, simplu de înțeles, ușor de implementat
- nu pornește de la anumite presupuneri, poate fi aplicat oricând
- nu are parametri
- poate fi folosit în clasificare binară sau cu mai multe clase dar și în regresie

## Dezavantaje:

- timpul de rulare este proporțional cu datele stocate în memorie
- cum alegem hiperparametrul  $K$  (= numărul de vecini) ?
- nu învață nimic (memorează datele), spațiu de stocare mare
- ce distanță alegem? caracteristicile trebuie normalize (să aibă aceeași scală)
- blestemul dimensionalității (ce se întâmplă în multe dimensiuni)
- sensitiv la date debalansate, punctele outlier, caracteristici lipsă

# Normalizarea caracteristicilor

# Normalizarea caracteristicilor

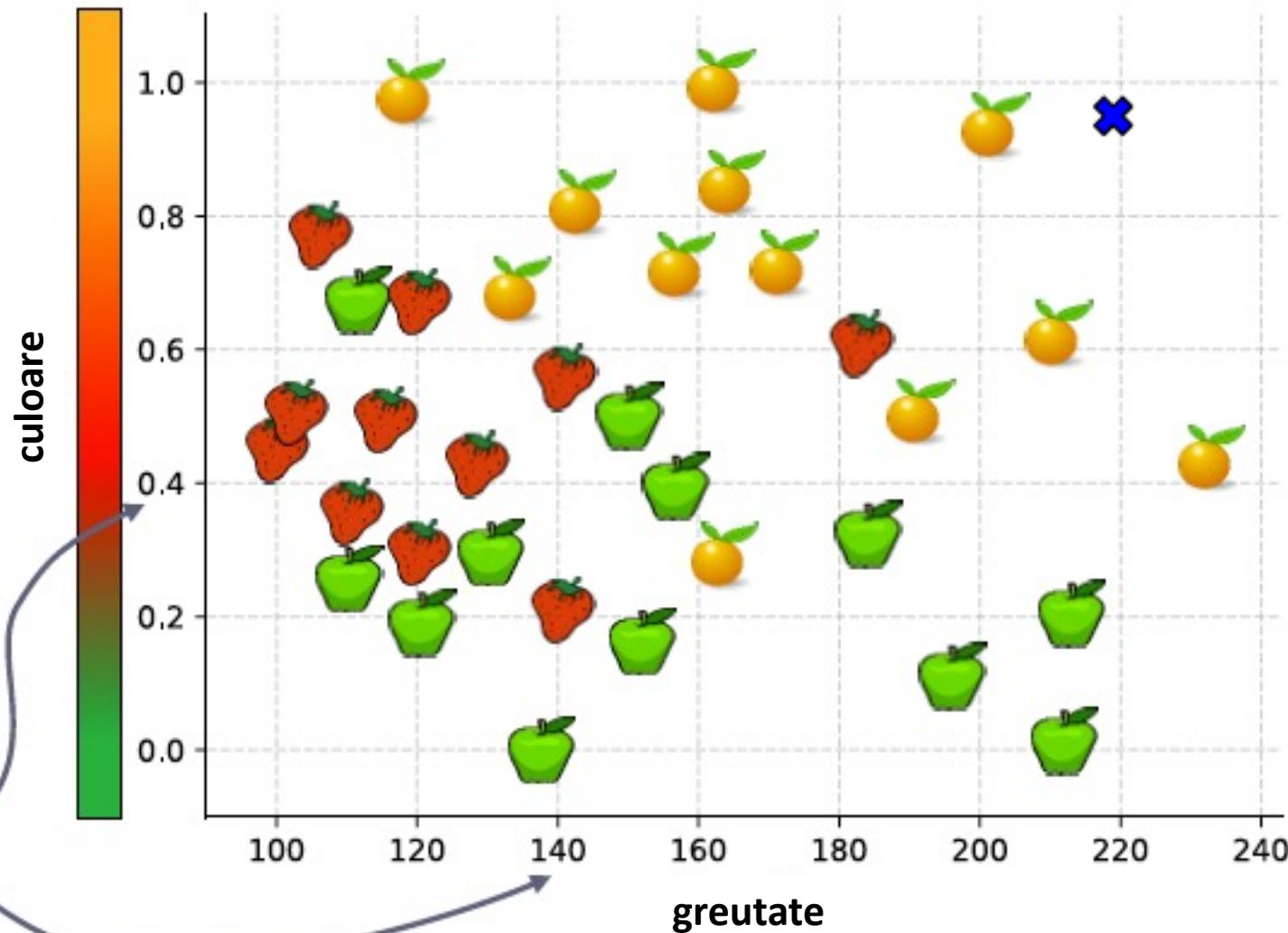


# Normalizarea caracteristicilor

$K = 1$

$\times = ?$

Caracteristicile  
au scale diferite



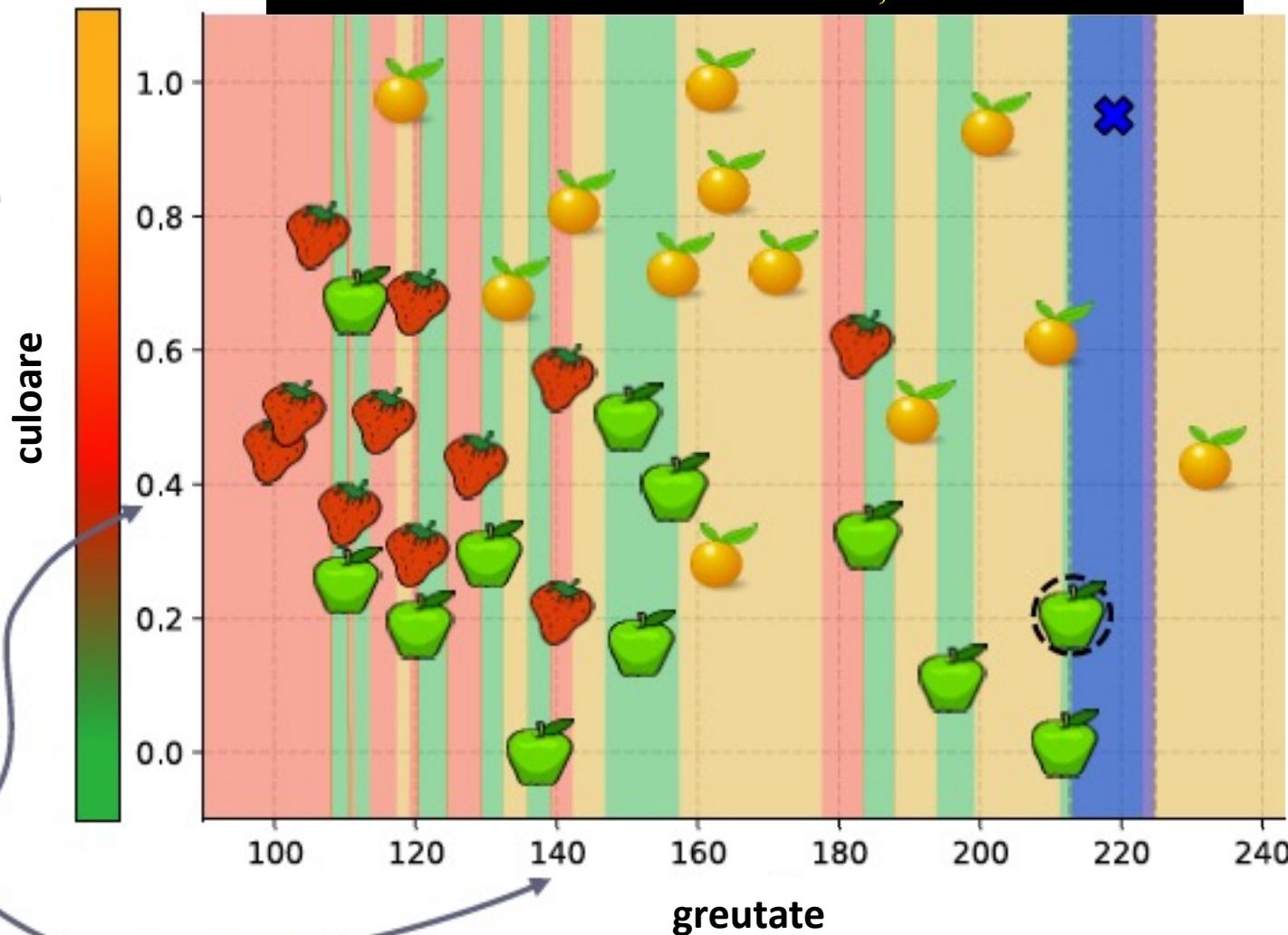
# Normalizarea caracteristicilor

$K = 1$

$\times = \text{apple}$

Caracteristicile  
au scale diferite

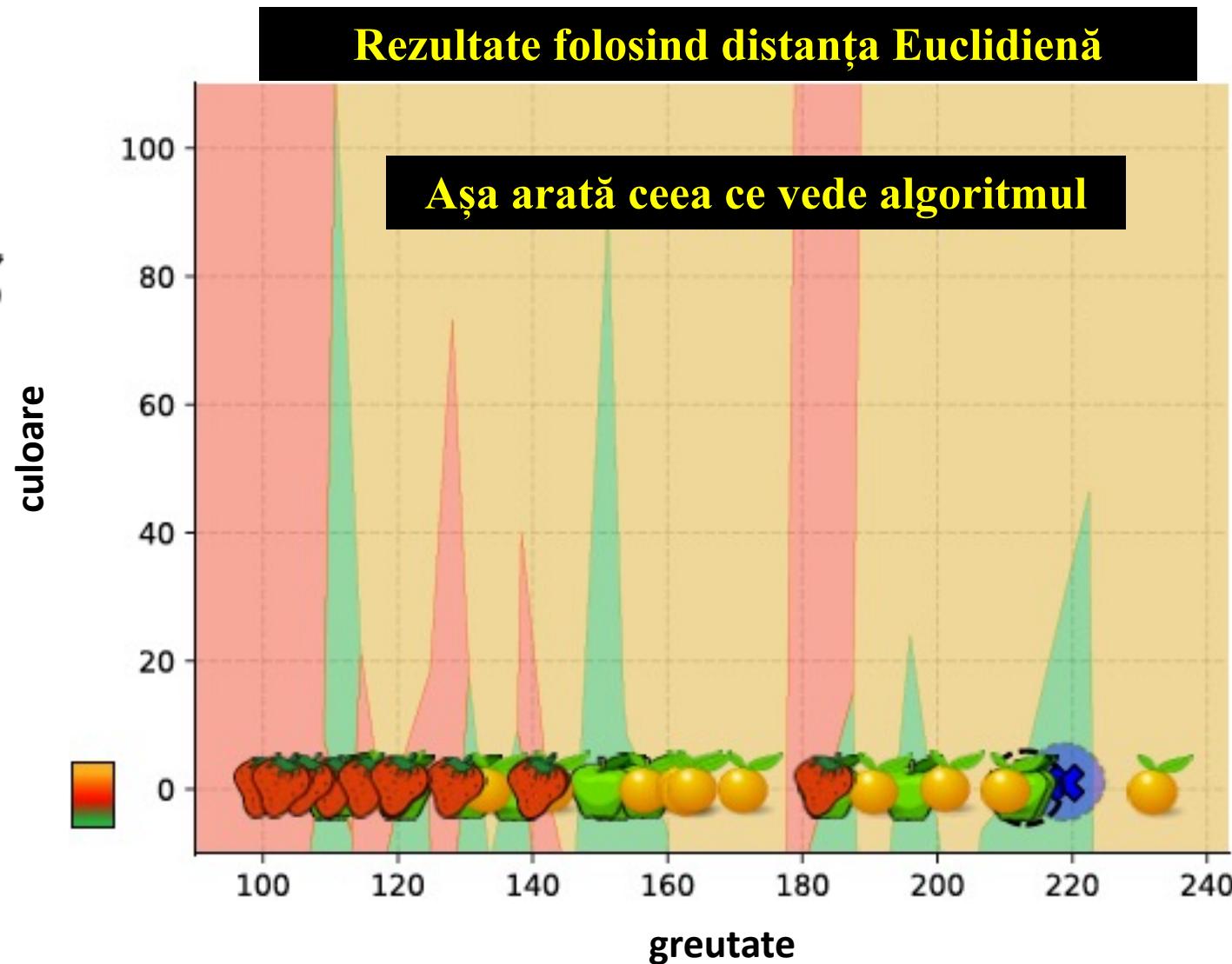
Rezultate folosind distanță Euclidiană



# Normalizarea caracteristicilor

$K = 1$

$\times = \text{apple}$

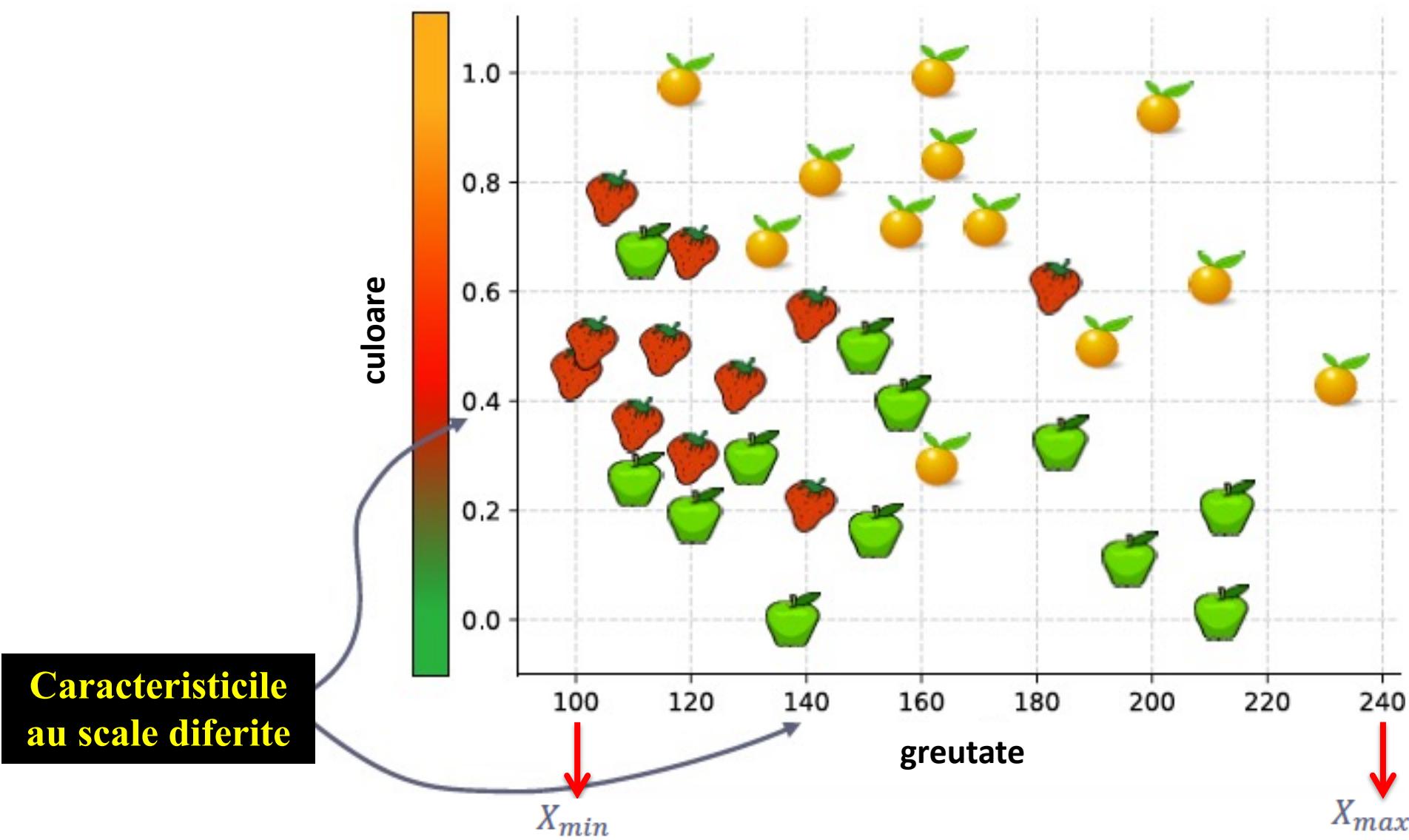


# Normalizarea caracteristicilor

- când caracteristicile au scale (mărimi) diferite, unele caracteristici domină pe celelalte în calculul distanței.
- vrem să avem toate caracteristicile în același interval:
  - *scalare de tip min-max* – transformă fiecare caracteristică în parte în intervalul [0, 1]:
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$
  - *standardizare* – transformă fiecare caracteristică în parte astfel încât au proprietăți ale distribuției normale standardizate (medie = 0 și deviație standard = 1):
$$X_{norm} = \frac{X - \mu_X}{\sigma_X}$$

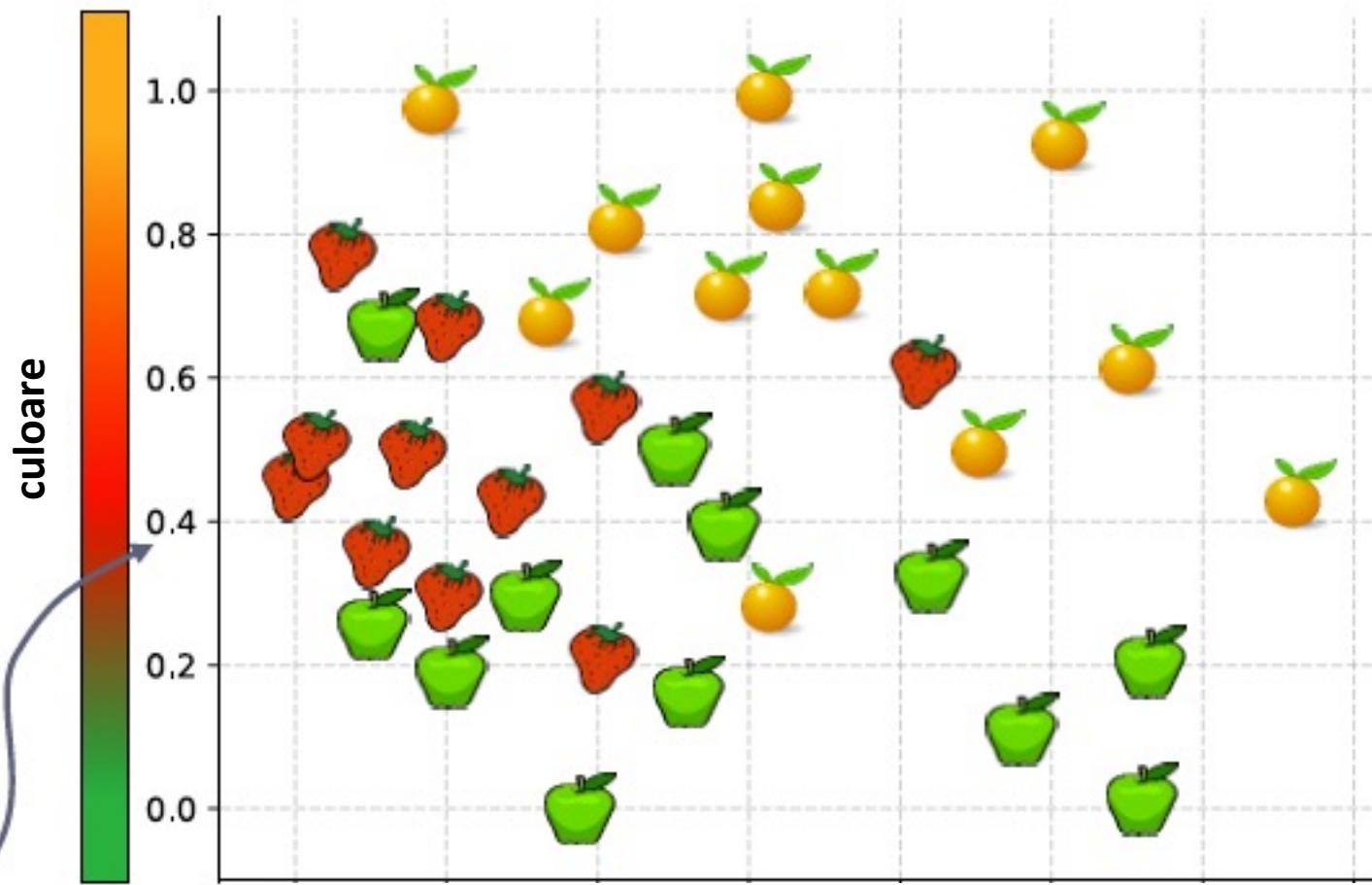
# Normalizarea caracteristicilor

## Scalare de tip min-max



# Normalizarea caracteristicilor

## Scalare de tip min-max



Caracteristicile  
au aceeași scală

$X_{min}$

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$X_{max}$

# Clasificatorul naïve Bayes

# Probabilități - recapitulare

- Avem două zaruri, fiecare având 6 fețe.  
Amestecăm zarurile și le aruncăm.
- Definim evenimentele următoare:

A: zarul 1 arată fața “3”,

B: zarul 2 arată fața “1”,

C: suma fețelor celor două zaruri este 8.

Calculați probabilitățile următoare:

$$1) P(A) = ? \quad P(A) = 1/6$$

$$P(B) = 1/6$$

$$P(C) = 5/36$$

$$P(A|B) = 1/6$$

$$P(C|A) = 1/6$$

$$P(A, B) = 1/36$$

$$P(A, C) = 1/36$$



# Probabilități - recapitulare

- Avem două zaruri, fiecare având 6 fețe.

Amestecăm zarurile și le aruncăm.

- Definim evenimentele următoare:

A: zarul 1 arată fața “3”,

B: zarul 2 arată fața “1”,

C: suma fețelor celor două zaruri este 8.

Calculați probabilitățile următoare:

1)  $P(A) = ?$        $P(A) = 1/6$

2)  $P(B) = ?$        $P(B) = 1/6$

3)  $P(C) = ?$        $P(C) = 5/36$

4)  $P(A | B) = ?$        $P(A|B) = 1/6$

5)  $P(C | A) = ?$        $P(C|A) = 1/6$

6)  $P(A, B) = ?$        $P(A, B) = 1/36$

7)  $P(A, C) = ?$        $P(A, C) = 1/36$

8) Este  $P(A, C) = P(A) * P(C)$ ?    NU

$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A, B) = P(A \cap B) = P(A | B) \times P(B)$$

Ω - spațiu total de evenimente – modelează fenomenul

Ω = {(1,1), (1,2), ..., (6,6)} – conține 36 de evenimente elementare, fiecare are probabilitatea 1/36

A = {(3,1), (3,2), (3,3), (3,4), (3,5), (3,6)}  $P(A) = 1/6$

B = {(1,1), (2,1), (3,1), (4,1), (5,1), (6,1)}  $P(B) = 1/6$

C = {(2,6), (3,5), (4,4), (5,3), (6,2)}  $P(C) = 5/36$

# Teorema lui Bayes

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

actualizează probabilitatea ipotezei A pe baza evidenței B

**Exemplu:** O boală rară are incidentă de 5% asupra populației. Un test medical care se face pentru depistarea bolii are acuratețe 90% (dă răspunsul corect în 90% din cazuri). Dacă o persoană face testul și ieșe pozitiv, care este probabilitatea ca ea să fie bolnavă?

# Teorema lui Bayes

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

**Exemplu:** O boală rară are incidență de 5% asupra populației. Un test medical care se face pentru depistarea bolii are acuratețe 90% (dă răspunsul corect în 90% din cazuri). Dacă o persoană face testul și ieșe pozitiv care este probabilitatea ca ea să fie bolnavă?

$\Omega$  - spațiu total de evenimente – modelează fenomenul

$\Omega = \{(c,t)\}$  – perechi de forma (clasă, test), unde clasa poate fi {bolnav, sanatos} iar testul poate fi {pozitiv, negativ}

A = persoana este bolnava,  $A = \{(bolnav, pozitiv), (bolnav, negativ)\}$

$$P(A) = 0.05, P(\bar{A}) = 0.95$$

B = testul ieșe pozitiv,  $B = \{(bolnav, pozitiv), (negativ, pozitiv)\}$

$$P(B|A) = 0.9$$

$$P(\bar{B}|\bar{A}) = 0.9$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.9 * 0.05}{P(B)}$$

# Teorema lui Bayes

$$P(B) = P(B \cap \Omega) = P(B \cap (A \cup \bar{A})) = P((B \cap A) \cup (B \cap \bar{A})) = P(B \cap A) + P(B \cap \bar{A})$$
$$P(B) = P(B|A)P(A) + P(B|\bar{A})P(\bar{A})$$

A = persoana este bolnava, A = {(bolnav, pozitiv), (bolnav, negativ)}  
P(A) = 0.05, P(\bar{A}) = 0.95

B = testul iese pozitiv, B = {(bolnav, pozitiv), (negativ, pozitiv)}  
P(B|A) = 0.9  
P(\bar{B}|\bar{A}) = 0.9, P(B|\bar{A}) = 0.1

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.9 * 0.05}{0.9 * 0.05 + 0.1 * 0.95} = \frac{0.045}{0.045 + 0.095} = \frac{0.045}{0.13} = 34.61\%$$

# Teorema lui Bayes

A = persoana este bolnava, A = {(bolnav, pozitiv), (bolnav, negativ)}

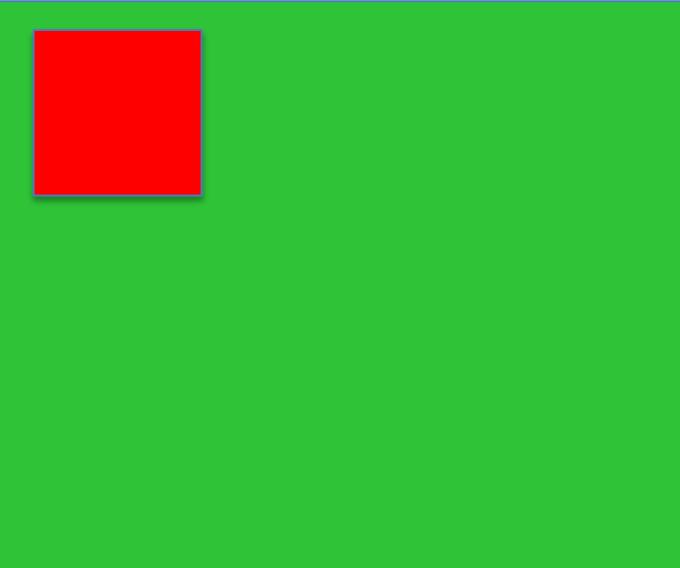
P(A) = 0.05, P( $\bar{A}$ ) = 0.95

B = testul iese pozitiv, B = {(bolnav, pozitiv), (negativ, pozitiv)}

P(B|A) = 0.9

P( $\bar{B} | \bar{A}$ ) = 0.9, P(B| $\bar{A}$ ) = 0.1

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.9 * 0.05}{0.9 * 0.05 + 0.1 * 0.95} = \frac{0.045}{0.045 + 0.095} = \frac{0.045}{0.13} = 34.61\%$$



Populatie sanatoasa = 95%

Populatia bolnava = 5%

# Teorema lui Bayes

A = persoana este bolnava, A = {(bolnav, pozitiv), (bolnav, negativ)}

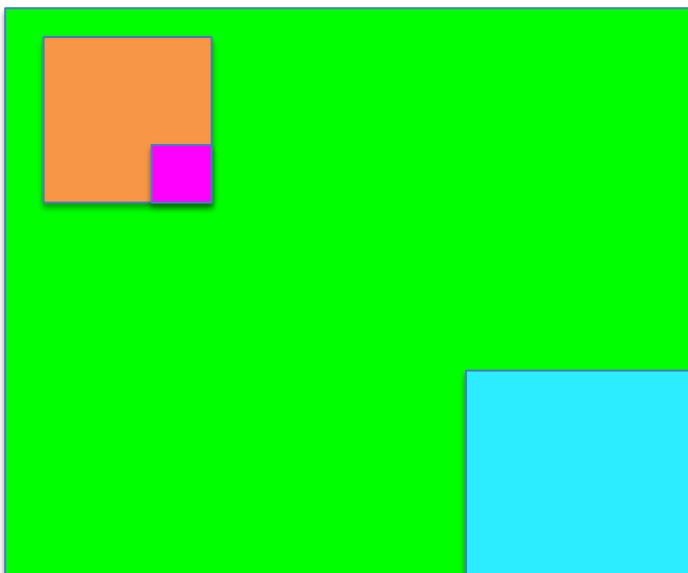
P(A) = 0.05, P( $\bar{A}$ ) = 0.95

B = testul iese pozitiv, B = {(bolnav, pozitiv), (negativ, pozitiv)}

P(B|A) = 0.9

P( $\bar{B} | \bar{A}$ ) = 0.9, P(B| $\bar{A}$ ) = 0.1

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.9 * 0.05}{0.9 * 0.05 + 0.1 * 0.95} = \frac{0.045}{0.045 + 0.095} = \frac{0.045}{0.13} = 34.61\%$$



Populatie sanatoasa, test negativ = 95% \* 0.9 = 0.855

Populatie sanatoasa, test pozitiv = 95% \* 0.1 = 0.095

Populatia bolnava, test pozitiv = 5% \* 0.9 = 0.045

Populatia bolnava, test negativ = 5% \* 0.1 = 0.005

Daca testul iese pozitiv, avem ca persoana provine fie din **regiunea portocalie (bolnav)** fie din **regiunea cyan (sanatos)**: probabilitatea ca persoana sa fie bolnava este  $0.045/(0.045+0.095) = 0.3461$

# Clasificare

- Definirea problemei:
  - date fiind caracteristicile măsurate  $X_1, X_2, \dots, X_n$
  - realizați o predicție a etichetei  $c$  (clasa)

# Clasificare - exemplu

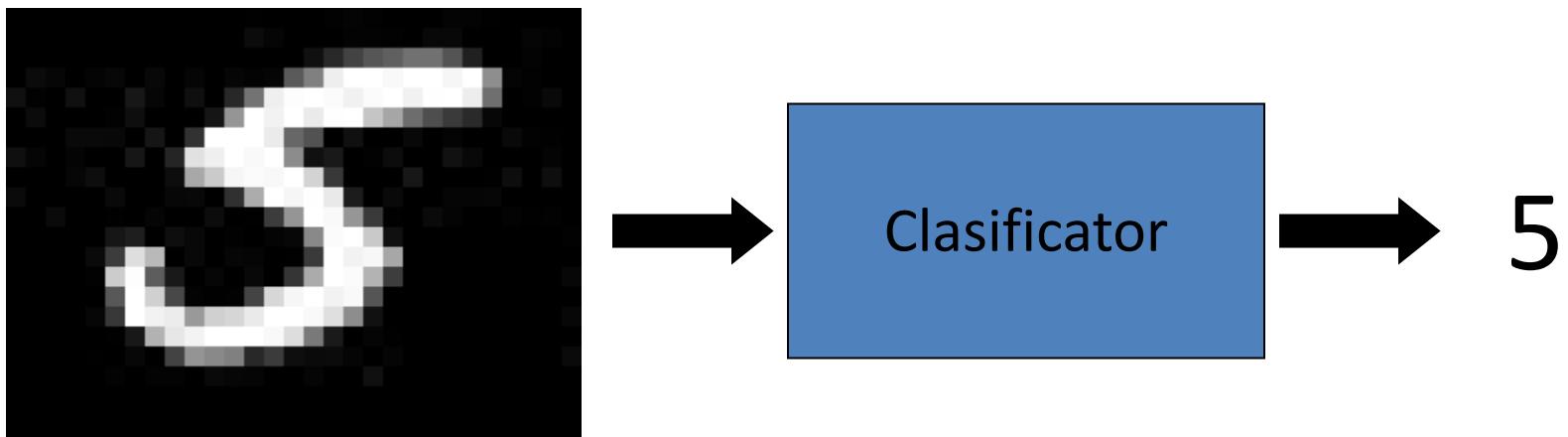
- Definirea problemei:
  - date fiind caracteristicile măsurate  $X_1, X_2, \dots, X_n$
  - realizați o predicție a etichetei  $c$  (clasa)
- Clasificare de gen :
  - $X_1, X_2, \dots, X_n$  sunt caracteristici măsurate la oameni: înălțimea, greutatea, etc.
  - $c \in \{B, F\}$  (clasifica dacă o persoană este bărbat sau femeie)

# Clasificare - exemplu

- Definirea problemei:
  - date fiind caracteristicile măsurate  $X_1, X_2, \dots, X_n$
  - realizați o predicție a etichetei  $c$
- Clasificare de solvabilitate pentru acordarea de credit :
  - $X_1, X_2, \dots, X_n$  sunt caracteristici măsurate la oameni de către bănci pentru analiza acordării unui credit: vârstă, sex, venituri, ocupație, stare civilă, număr copii, etc.
  - $c \in \{S, I\}$  (clasifica dacă o persoană este solvabilă sau nu, dacă poate returna creditul la timp)

# Clasificare - exemplu

- Definirea problemei:
  - date fiind caracteristicile măsurate  $X_1, X_2, \dots, X_n$
  - realizați o predicție a etichetei  $c$
- Clasificarea cifrelor scrise de mână (laborator 3+4):
  - $X_1, X_2, \dots, X_n$  sunt intensitățile pixelilor dintr-o imagine grayscale
  - $X_1, X_2, \dots, X_n \in \{0, 1, 2, \dots, 255\}$  – intensități
  - $c \in \{0, 1, 2, \dots, 9\}$  (clasifică ce cifră este imaginea)



# Regula lui Bayes

- Clasificarea cifrelor scrise de mână:
  - $X_1, X_2, \dots, X_n$  sunt intensitățile pixelilor dintr-o imagine grayscale
  - $X_1, X_2, \dots, X_n \in \{0, 1, 2, \dots, 255\}$  – intensități
  - $c \in \{0, 1, 2, \dots, 9\}$  (clasifică ce cifră este imaginea)
- Considerăm  $X$  imaginea cu  $n$  pixeli de intensitate  $X_1, X_2, \dots, X_n$ :  $X = (X_1, X_2, \dots, X_n)$
- Regula lui Bayes:  $P(c | X) = \frac{P(X | c) \times P(c)}{P(X)}$

$\Omega$  - spațiu total de evenimente – modelează fenomenul

$\Omega = \{(c, X)\}$  – perechi de forma (clasă, imagine)

# Regula lui Bayes

$X =$



$$P(c=0|X) = 0.5$$

$$P(c=6|X) = 0.2$$

$$P(c=8|X) = 0.3$$

$X =$



$$P(c=1|X) = 0.05$$

$$P(c=3|X) = 0.8$$

$$P(c=7|X) = 0.15$$

$X =$



$$P(c=4|X) = 0.35$$

$$P(c=9|X) = 0.65$$

# Imaginea X (din setul MNIST) văzută de algoritm

# Regula lui Bayes

- Considerăm X imaginea cu pixelii de intensitate  $X_1, X_2, \dots, X_n$ :  $X = (X_1, X_2, \dots, X_n)$

- Regula lui Bayes:

$$P(c | X) = \frac{P(X | c) \times P(c)}{P(X)}$$

Probabilitatea să observăm imaginea X condiționată de faptul că imaginea conține o cifră din clasa c

Probabilitatea a-priori ca o imagine să fie din clasa c (să conțină cifră din clasa c)

Probabilitatea să avem cifra din clasa c dându-se imaginea X

Probabilitatea să observăm imaginea X

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

# Regula de clasificare

$$P(c = 0 / X) = \frac{P(X / c = 0) \times P(c = 0)}{P(X)}$$

$$P(c = 1 / X) = \frac{P(X / c = 1) \times P(c = 1)}{P(X)}$$

.....

$$P(c = 9 / X) = \frac{P(X / c = 9) \times P(c = 9)}{P(X)}$$

Presupunem că am calculat cele 10 probabilități a-posteriori:  
 $P(c=0|X)$ ,  $P(c=1|X)$ , ...  $P(c=9|X)$

Cum arată regula de clasificare?

# Regula de clasificare

$$P(c = 0 / X) = \frac{P(X / c = 0) \times P(c = 0)}{P(X)}$$

$$P(c = 1 / X) = \frac{P(X / c = 1) \times P(c = 1)}{P(X)}$$

.....

$$P(c = 9 / X) = \frac{P(X / c = 9) \times P(c = 9)}{P(X)}$$

Presupunem că am calculat cele 10 probabilități a-posteriori:  
 $P(c=0|X)$ ,  $P(c=1|X)$ , ...  $P(c=9|X)$

Regula de clasificare:  $c^* = \arg \max_{i=0,1,\dots,9} P(c = i / X)$

**alege clasa care maximizează probabilitatea a-posteriori**

# Regula de clasificare

$$P(c = 0 / X) = \frac{P(X / c = 0) \times P(c = 0)}{P(X)}$$

$$P(c = 1 / X) = \frac{P(X / c = 1) \times P(c = 1)}{P(X)}$$

.....

$$P(c = 9 / X) = \frac{P(X / c = 9) \times P(c = 9)}{P(X)}$$

Numitorul este același.  
Pot renunța la el, nu mă interesează valoarea exactă a probabilității ci care probabilitate e mai mare.

Presupunem că am calculat cele 10 probabilități a-posteriori:  
 $P(c=0|X), P(c=1|X), \dots P(c=9|X)$

Regula de clasificare:  $c^* = \arg \max_{i=0,1,\dots,9} P(c = i / X)$

**alege clasa care maximizează probabilitatea a-posteriori**

# Regula de clasificare

$\propto$  - direct proporțional

$$P(c = 0 / X) \propto P(X / c = 0) \times P(c = 0)$$

$$P(c = 1 / X) \propto P(X / c = 1) \times P(c = 1)$$

.....

$$P(c = 9 / X) \propto P(X / c = 9) \times P(c = 9)$$

Presupunem că am calculat cele 10 probabilități:

$$\text{Regula de clasificare: } c^* = \operatorname{argmax}_{i=0,1,\dots,9} P(X / c = i) \times P(c = i)$$

**alege clasa care maximizează numărătorul**

# Calculul probabilității a-priori

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

Probabilitatea a-priori

- calculăm probabilitatea a-priori numărând câte exemple din mulțimea de antrenare au clasa  $I$
- lucrați la laborator cu setul de date MNIST, mulțimea de antrenare are 1000 de exemple etichetate

```
1 #compute p_C
2 p_C = np.zeros(10, 'uint8')
3 for label in train_labels:
4     p_C[label] = p_C[label]+1
5 print(p_C/sum(p_C))
```

[0.091 0.12 0.096 0.099 0.105 0.083 0.098 0.104 0.101 0.103]

↓

P(c=0)

↓

P(c=9)

# Calculul probabilității likelihood

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

Probabilitatea  
likelihood

- $P(X|c = i) = P(X_1=x_1, X_2=x_2, \dots, X_{784}=x_{784} | c = i)$

Pentru exemplul de mai devreme vrem să calculăm probabilitatea **comună**:

- $P(X|c = i) = P(X_1=0, X_2=0, \dots, X_{784}=253 | c = i)$

Calculul acestei probabilități este practic imposibil, nu avem aşa de multe date de antrenare (avem numai 1000 cu toate clasele, în jur de 100 per clasă) care să cuprindă toate n-uplurile posibile

# Calculul probabilității likelihood folosind clasificatorul naïve Bayes

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

↑  
Probabilitatea  
likelihood

- **clasificatorul naïve Bayes consideră caracteristicile independente (nu e întotdeauna adevărat acest lucru)**

$$P(X | c = i) = \prod_{j=1}^{n=784} P(X_j = x_j | c = i)$$

- $P(X|c = i) = P(X_1=x_1, X_2=x_2, \dots, X_{784}=x_{784}|c = i) =$  (folosind presupunerea de independentă între caracteristici) =  
 $= P(X_1=x_1|c=i) \times P(X_2=x_2|c=i) \times \dots P(X_{784}=x_{784}|c = i)$

# Regula de clasificare pentru clasificatorul naïve Bayes

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

Regula de clasificare:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} P(X | c = i) \times P(c = i)$$

Regula de clasificare pentru naïve Bayes:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \prod_{j=1}^{n=784} P(X_j = x_j | c = i) \right) \times P(c = i)$$

# Regula de clasificare pentru clasificatorul naïve Bayes

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

Regula de clasificare:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} P(X | c = i) \times P(c = i)$$

Regula de clasificare pentru naïve Bayes:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \prod_{j=1}^{n=784} P(X_j = x_j | c = i) \right) \times P(c = i)$$

Cum o calculez?

# Calculul probabilității individuale $P(X_j = x_j | c=i)$

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \prod_{j=1}^n P(X_j = x_j | c = i) \right) \times P(c = i)$$

Cum o calculez?

$$P(X_j = 0 | c) = ?$$

$$P(X_j = 1 | c) = ?$$

$$P(X_j = 2 | c) = ?$$

...

$$P(X_j = 255 | c) = ?$$

Nu avem date să estimăm corect asemenea probabilități, avem în jur de 100 de puncte  $X_j$  în mulțimea de antrenare.

[ 0.091 0.12 0.096 0.099 0.105 0.083 0.098 0.104 0.101 0.103 ]

↓  
 $P(c=0)$

↓  
 $P(c=9)$

# Calculul probabilității individuale $P(X_j = x_j | c=i)$

**Exemplu: consider  $j = 370$ ,  $c = 0$   
(poziția 370 în vectorul de 784 de componente, cifra 0)**

```
1 #calculeaza p(Xj = xj/c)
2 c = 0
3 index = np.ravel(np.where(train_labels == 0))
4 print(index)
5 j = 370
6 features = train_images[index,j]
7 print(features.shape)
8 print(features.size)
9 print(features)
```

```
[ 15  28  29  33  52  79  86  87  98 113 122 124 141 148 173 176 186 226
 230 232 237 244 247 255 258 265 279 283 287 289 296 315 320 322 332 333
 335 367 378 383 388 397 400 411 446 461 466 479 482 493 498 509 517 531
 544 552 567 569 573 611 640 643 645 660 677 684 689 715 728 732 760 789
 800 806 825 828 835 839 875 880 881 882 899 915 919 932 943 963 972 974
 981]
(91,)
91
[ 25.   0. 120.   0.   86. 253. 243. 218.   0.   0. 253.   0. 252. 229.
   0.   0. 252.   0.   0.   0.   0.   0. 250.   0. 242.   0. 253.
 254.   0. 254. 252.   0.   0.   0. 55.   0.   0.   0. 250.   0.   0.
   0.   0.   0. 254. 11. 255.   0.   0.   0. 254. 252. 253.   0.
 255.   0.   0.   0.   0. 103. 32. 63. 61. 37.   0.   0. 166.
   0.   0.   0. 84. 253. 25. 253.   0.   0.   0. 182. 95. 156. 225.
 191.   0.   0. 112.   0.   0.   0.]
```

Indecșii celor 91 de imagini de antrenare ce conțin cifra 0

Cele 91 de valori ale pixelului 370 pentru exemplele de antrenare cu cifra 0

Calculul probabilității individuale  $P(X_j = x_j | c=i)$

**Exemplu: consider  $j = 370$ ,  $c = 0$   
(poziția 370 în vectorul de 784 de componente, cifra 0)**

```
[ 25.    0.  120.    0.   86.  253.  243.  218.    0.    0.  253.    0.  252.  229.  
     0.    0.  252.    0.    0.    0.    0.    0.  250.    0.  242.    0.  253.  
 254.    0.  254.  252.    0.    0.    0.  55.    0.    0.    0.  250.    0.    0.  
     0.    0.    0.    0.  254.    11.  255.    0.    0.    0.  254.  252.  253.    0.  
 255.    0.    0.    0.    0.    0.  103.    32.    63.    61.    37.    0.    0.  166.  
     0.    0.    0.    84.  253.    25.  253.    0.    0.    0.  182.    95.  156.  225.  
191.    0.    0.  112.    0.    0.    0.]
```

Cele 91 de valori ale pixelului 370 pentru exemplele de antrenare cu cifra 0

```
[50.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  2.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  1.  0.  0.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  1.  0.  0.  
 0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  
 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  
 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  0.  
 0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  2.  0.  
4.  6.  4.  2.]
```

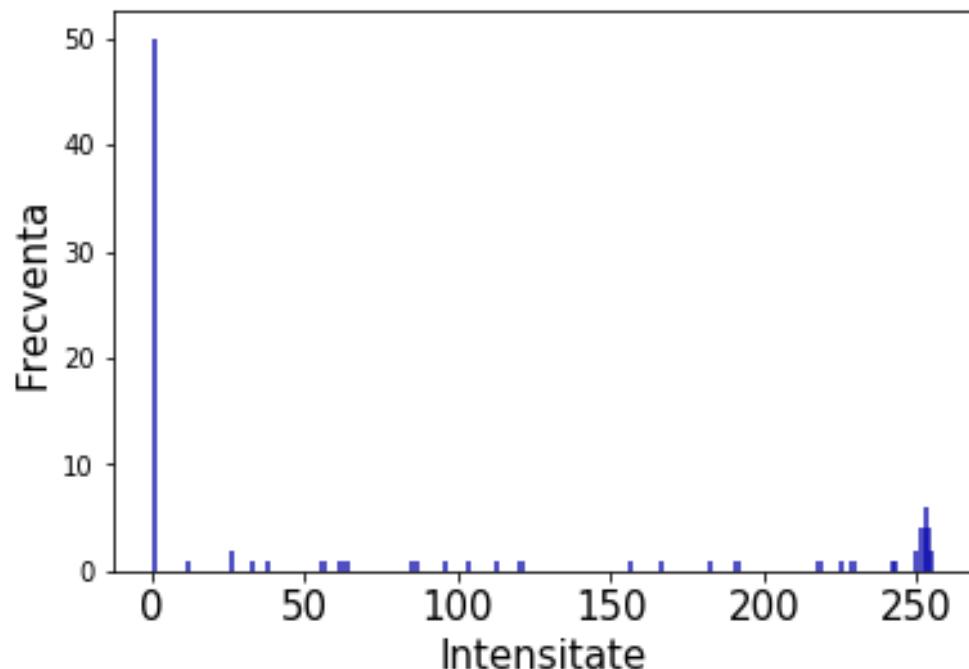
Vector de frecvență pentru fiecare intensitate

# Calculul probabilității individuale $P(X_j = x_j | c=i)$

**Exemplu: consider  $j = 370$ ,  $c = 0$   
(poziția 370 în vectorul de 784 de componente, cifra 0)**

```
[ 25.   0. 120.   0.   86. 253. 243. 218.   0.   0. 253.   0. 252. 229.  
  0.   0. 252.   0.   0.   0.   0.   0. 250.   0. 242.   0. 253.  
254.   0. 254. 252.   0.   0.   0. 55.   0.   0.   0. 250.   0.   0.  
  0.   0.   0. 254. 11. 255.   0.   0.   0. 254. 252. 253.   0.  
255.   0.   0.   0.   0. 103. 32. 63. 61. 37.   0.   0. 166.  
  0.   0.   0. 84. 253. 25. 253.   0.   0.   0. 182. 95. 156. 225.  
191.   0.   0. 112.   0.   0.   0.]
```

Cele 91 de valori ale pixelului 370 pentru exemplele de antrenare cu cifra 0



Vector de frecvență pentru fiecare intensitate

# Calculul probabilității individuale $P(X_j = x_j | c=i)$

Nu avem date să estimăm corect asemenea probabilități, avem în jur de 100 de puncte  $X_j$  în mulțimea de antrenare.

O posibilă soluție:

Împart intervalul de valori posibile [0,255] în p părți egale și estimez care este probabilitatea ca o valoare să apară într-un astfel de interval.

Exemplu: consider  $p = 4$ , obțin 4 intervale:

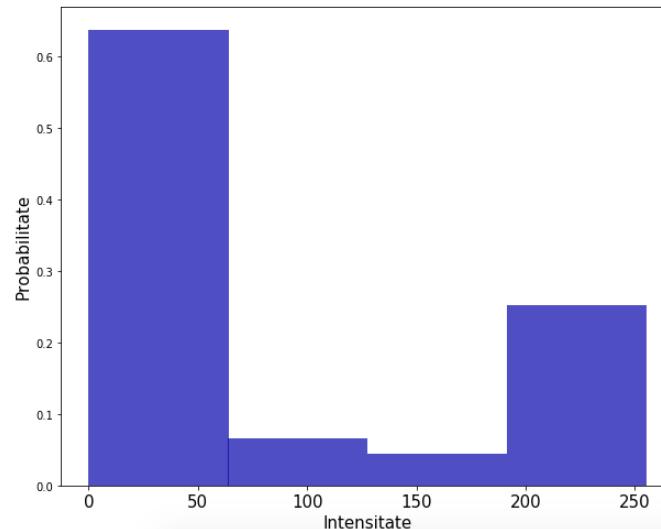
[0, 63], [64, 127], [128, 191], [192,255]

Aproximez:

$P(X_j = 0|c)$  cu  $P(X_j \in [0,63]|c)$

$P(X_j = 100|c)$  cu  $P(X_j \in [64,127]|c)$

Probabilitatea va arăta astfel:

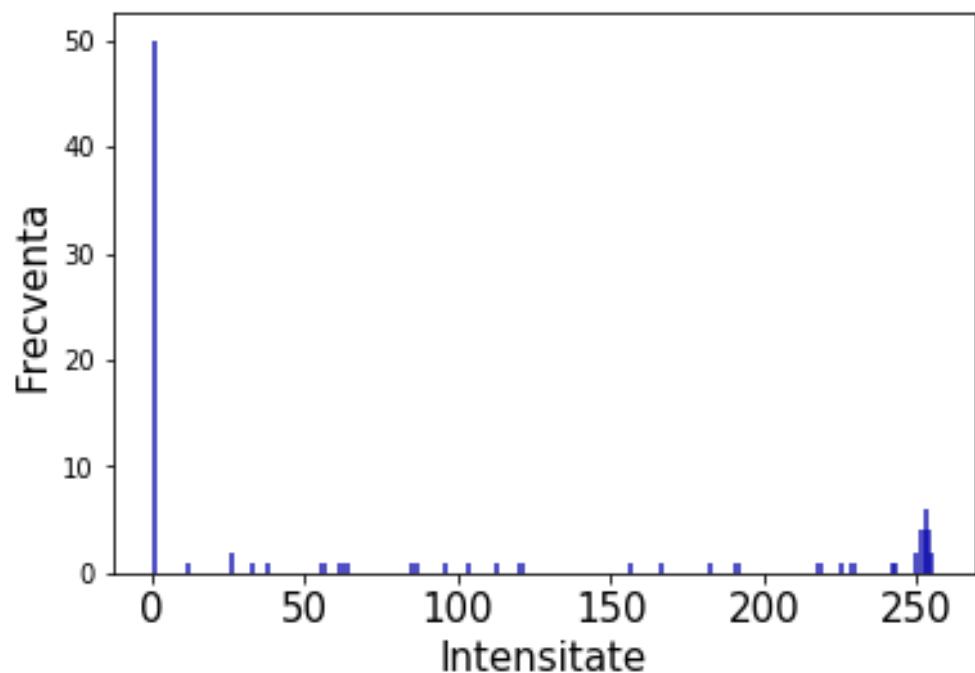


# Calculul probabilității individuale $P(X_j = x_j | c=i)$

Nu avem date să estimăm corect asemenea probabilități, avem în jur de 100 de puncte  $X_j$  în mulțimea de antrenare.

**Soluție alternativă:**

**Încerc să fitez o distribuție parametrică (distribuție normală, etc.) care să se potrivească cu datele.**



# Stabilitate numerică

Regula de clasificare pentru naïve Bayes:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \prod_{j=1}^{n=784} P(X_j = x_j \mid c = i) \right) \times P(c = i)$$


înmulțesc  $n=784$  numere subunitare (sunt probabilități), uneori foarte aproape de 0.

Aplic funcția logaritm (e monoton crescătoare), păstrează ierarhia:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \log \left( \prod_{j=1}^{n=784} P(X_j = x_j \mid c = i) \right) \right) \times P(c = i))$$
$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \sum_{j=1}^{n=784} \log(P(X_j = x_j \mid c = i)) + \log(P(c = i)) \right)$$

# Laborator – Matrice confuzie

	0	1	2	3	4	5	6	7	8	9
0	[ [ 48	0	1	0	0	0	3	0	1	0 ]
1	[ 0	46	1	0	0	1	0	0	4	0 ]
2	[ 1	0	52	3	1	0	0	1	0	0 ]
3	[ 0	0	4	43	1	1	0	0	1	3 ]
4	[ 0	0	0	0	40	1	0	0	2	3 ]
5	[ 1	0	2	11	1	31	0	0	6	0 ]
6	[ 1	0	2	0	2	1	42	0	0	0 ]
7	[ 0	0	1	0	2	0	0	38	1	8 ]
8	[ 0	0	2	4	2	3	1	1	32	1 ]
9	[ 0	0	1	1	12	0	0	1	1	26 ] ]

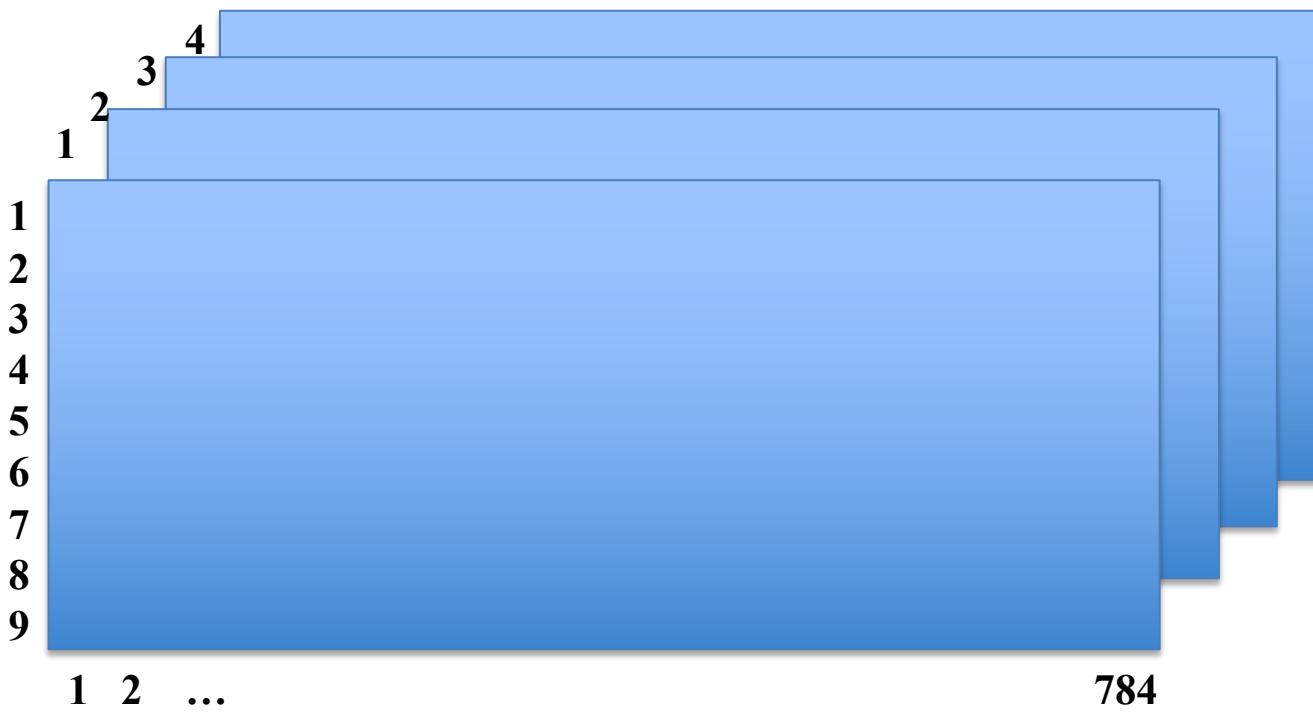
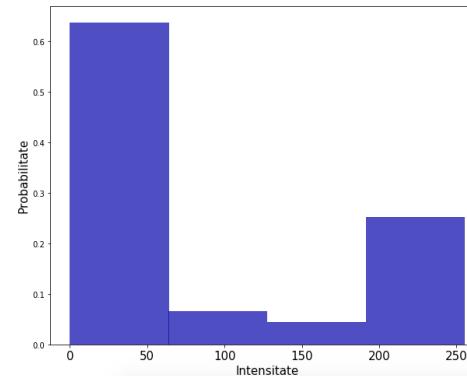
Numărul de exemple de testare cu cifra  
9 care au fost clasificate drept cifra 4

# Model parametric

Care sunt parametri învățați de către model?

**Modelul învăță din datele de antrenare pentru fiecare clasă i (linie) și componentă j (coloană)**

$P(X_j = x_j | c=i)$  sub forma unui vector de probabilități de dimensiune 4



Numărul de parametri:  
 $784 \times 10 \times 4 = 31360$

De fapt întrucât fiecare vector de dimensiune 4 are suma 1:  
 $784 \times 10 \times 3 = 24520$

**Hiperparametrul p = numărul de părți, p = 4**

# Clasificatorul Bayes pe cazul general

- X vector de n caracteristici  $X = (X_1, X_2, \dots, X_n)$
- c clase (admis/respins, bărbat/femeie,litere, cifre, etc.)
- Regula lui Bayes:

$$P(c | X) = \frac{P(X | c) \times P(c)}{P(X)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

- **clasificatorul naïve Bayes consideră caracteristicile independente (nu e întotdeauna adevărat acest lucru)**

$$P(X | c = i) = \prod_{j=1}^n P(X_j = x_j | c = i)$$

# Clasificatorul Bayes pe cazul general

- $X$  vector de  $n$  caracteristici  $X = (X_1, X_2, \dots, X_n)$
- $c$  clase (admis/respins, litere, cifre, etc.)
- Regula de clasificare pentru Naïve Bayes:

$$c^* = \operatorname{argmax}_i \left( \prod_{j=1}^n P(X_j = x_j \mid c = i) \right) \times P(c = i)$$

trebuie estimată din date

- stabilitate numerică:

$$c^* = \operatorname{argmax}_i \left( \log \left( \prod_{j=1}^n P(X_j = x_j \mid c = i) \right) \times P(c = i) \right)$$

$$c^* = \operatorname{argmax}_i \left( \sum_{j=1}^n \log(P(X_j = x_j \mid c = i)) + \log(P(c = i)) \right)$$

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 4

# Recapitulare – cursul trecut

1. Modelul celor mai apropiati k vecini (k-nearest neighbors) – continuare
2. Normalizarea caracteristicilor
3. Clasificatorul naïve Bayes

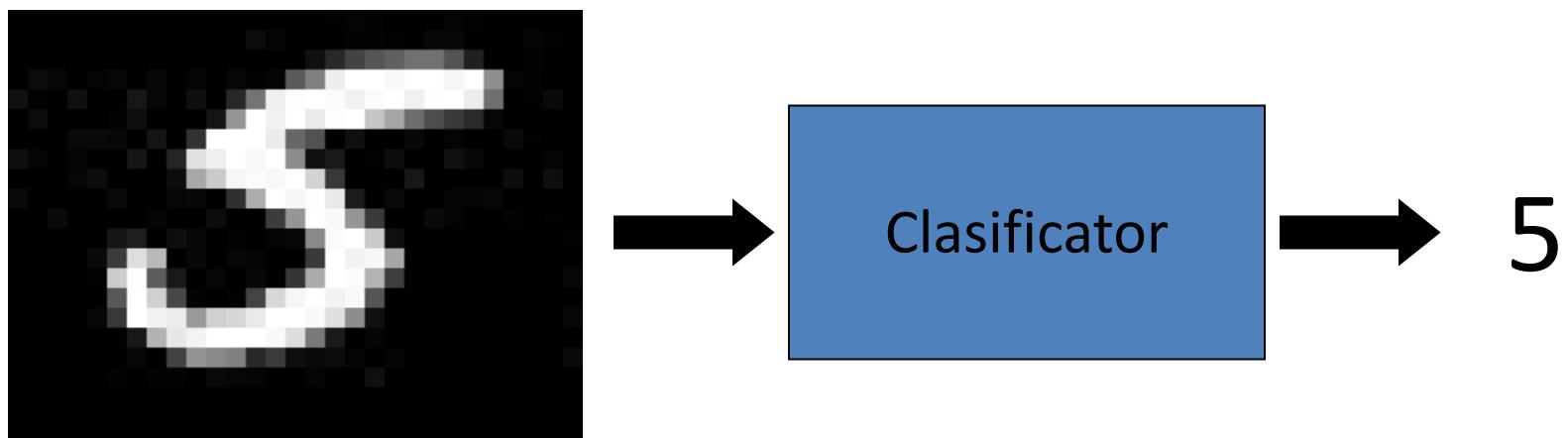
# Cuprinsul cursului de azi

1. Clasificatorul naïve Bayes
2. Evaluarea performanței unui model
3. Strategii de împărțire a datelor
4. Proiect
  - concurs pe platforma Kaggle
  - demo
  - laborator săptămâna 5
5. Mașini cu vectori suport (SVMs – Support Vector Machines)

# Clasificatorul naïve Bayes

# Clasificare - exemplu

- Definirea problemei:
  - date fiind caracteristicile măsurate  $X_1, X_2, \dots, X_n$
  - realizați o predicție a etichetei  $c$
- Clasificarea cifrelor scrise de mână (laborator):
  - $X_1, X_2, \dots, X_n$  sunt intensitățile pixelilor dintr-o imagine grayscale
  - $X_1, X_2, \dots, X_n \in \{0, 1, 2, \dots, 255\}$  – intensități
  - $c \in \{0, 1, 2, \dots, 9\}$  (clasifică ce cifră este imaginea)



# Regula lui Bayes

- Clasificarea cifrelor scrise de mână:
  - $X_1, X_2, \dots, X_n$  sunt intensitățile pixelilor dintr-o imagine grayscale
  - $X_1, X_2, \dots, X_n \in \{0, 1, 2, \dots, 255\}$  – intensități
  - $c \in \{0, 1, 2, \dots, 9\}$  (clasifică ce cifră este imaginea)
- Considerăm  $X$  imaginea cu  $n$  pixeli de intensitate  $X_1, X_2, \dots, X_n$ :  $X = (X_1, X_2, \dots, X_n)$
- Regula lui Bayes:  $P(c | X) = \frac{P(X | c) \times P(c)}{P(X)}$

$\Omega$  - spațiu total de evenimente – modelează fenomenul

$\Omega = \{(c, X)\}$  – perechi de forma (clasă, imagine)

# Regula lui Bayes

- Considerăm X imaginea cu pixelii de intensitate  $X_1, X_2, \dots, X_n$ :  $X = (X_1, X_2, \dots, X_n)$

- Regula lui Bayes:

$$P(c | X) = \frac{P(X | c) \times P(c)}{P(X)}$$

Probabilitatea să observăm imaginea X condiționată de faptul că imaginea conține o cifră din clasa c

Probabilitatea a-priori ca o imagine să fie din clasa c (să conțină cifră din clasa c)

Probabilitatea să avem cifra din clasa c dându-se imaginea X

Probabilitatea să observăm imaginea X

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

# Regula de clasificare

$$P(c = 0 / X) = \frac{P(X / c = 0) \times P(c = 0)}{P(X)}$$

$$P(c = 1 / X) = \frac{P(X / c = 1) \times P(c = 1)}{P(X)}$$

.....

$$P(c = 9 / X) = \frac{P(X / c = 9) \times P(c = 9)}{P(X)}$$

Presupunem că am calculat cele 10 probabilități a-posteriori:  
 $P(c=0|X)$ ,  $P(c=1|X)$ , ...  $P(c=9|X)$

Regula de clasificare:  $c^* = \arg \max_{i=0,1,\dots,9} P(c = i / X)$

**alege clasa care maximizează probabilitatea a-posteriori**

# Regula de clasificare

$\propto$  - direct proporțional

$$P(c = 0 / X) \propto P(X / c = 0) \times P(c = 0)$$

$$P(c = 1 / X) \propto P(X / c = 1) \times P(c = 1)$$

.....

$$P(c = 9 / X) \propto P(X / c = 9) \times P(c = 9)$$

Presupunem că am calculat cele 10 probabilități:

$$\text{Regula de clasificare: } c^* = \operatorname{argmax}_{i=0,1,\dots,9} P(X / c = i) \times P(c = i)$$

**alege clasa care maximizează numărătorul**

# Calculul probabilității likelihood folosind clasificatorul naïve Bayes

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

↑  
Probabilitatea  
likelihood

- **clasificatorul naïve Bayes consideră caracteristicile independente (nu e întotdeauna adevărat acest lucru)**

$$P(X | c = i) = \prod_{j=1}^{n=784} P(X_j = x_j | c = i)$$

- $P(X|c = i) = P(X_1=x_1, X_2=x_2, \dots, X_{784}=x_{784}|c = i) =$  (folosind presupunerea de independentă între caracteristici) =  
 $= P(X_1=x_1|c=i) \times P(X_2=x_2|c=i) \times \dots P(X_{784}=x_{784}|c = i)$

# Calculul probabilității individuale $P(X_j = x_j | c=i)$

Nu avem date să estimăm corect asemenea probabilități, avem în jur de 100 de puncte  $X_j$  în mulțimea de antrenare.

O posibilă soluție:

Împart intervalul de valori posibile [0,255] în p părți egale și estimez care este probabilitatea ca o valoare să apară într-un astfel de interval.

Exemplu: consider  $p = 4$ , obțin 4 intervale:

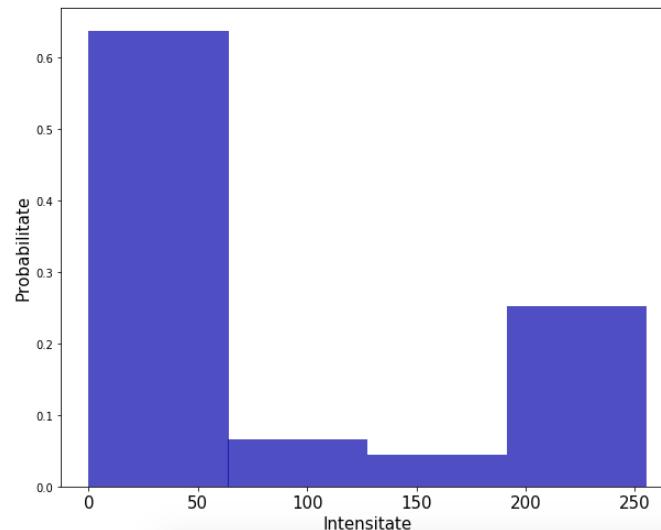
[0, 63], [64, 127], [128, 191], [192,255]

Aproximez:

$P(X_j = 0|c)$  cu  $P(X_j \in [0,63]|c)$

$P(X_j = 100|c)$  cu  $P(X_j \in [64,127]|c)$

Probabilitatea va arăta astfel:



# Stabilitate numerică

Regula de clasificare pentru naïve Bayes:

$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \prod_{j=1}^{n=784} P(X_j = x_j \mid c = i) \right) \times P(c = i)$$


înmulțesc  $n=784$  numere subunitare (sunt probabilități), uneori foarte aproape de 0.

Aplic funcția logaritm (e monoton crescătoare), păstrează ierarhia:

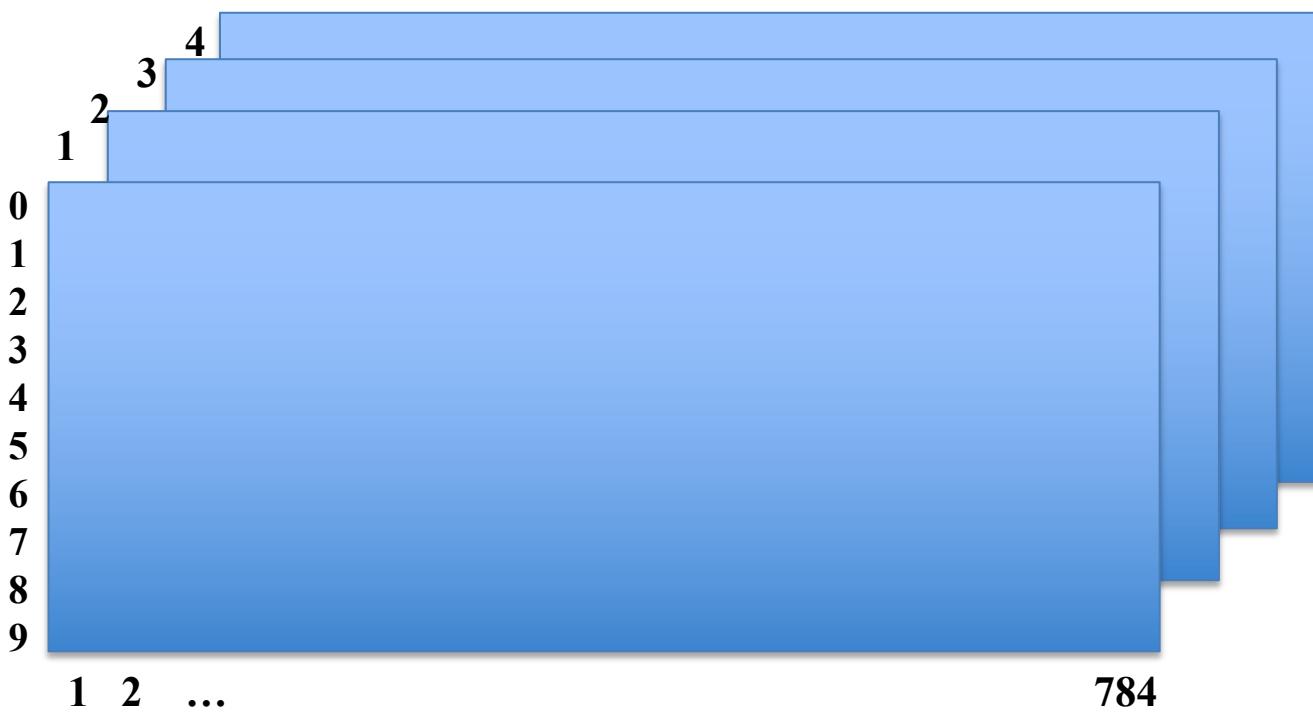
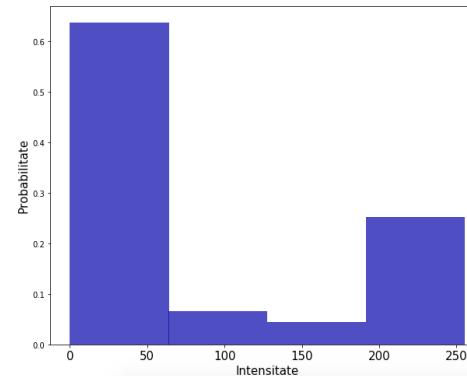
$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \log \left( \prod_{j=1}^{n=784} P(X_j = x_j \mid c = i) \right) \right) \times P(c = i))$$
$$c^* = \operatorname{argmax}_{i=0,1,\dots,9} \left( \sum_{j=1}^{n=784} \log(P(X_j = x_j \mid c = i)) + \log(P(c = i)) \right)$$

# Model parametric

Care sunt parametri învățați de către model?

**Modelul învăță din datele de antrenare pentru fiecare clasă i (linie) și componentă j (coloană)**

$P(X_j = x_j | c=i)$  sub forma unui vector de probabilități de dimensiune 4



Numărul de parametri:  
 $784 \times 10 \times 4 = 31360$

De fapt întrucât fiecare vector de dimensiune 4 are suma 1:  
 $784 \times 10 \times 3 = 24520$

**Hiperparametrul p = numărul de părți, p = 4**

# Clasificatorul Bayes pe cazul general

- X vector de n caracteristici  $X = (X_1, X_2, \dots, X_n)$
- c clase (admis/respins, bărbat/femeie,litere, cifre, etc.)
- Regula lui Bayes:

$$P(c | X) = \frac{P(X | c) \times P(c)}{P(X)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$P(c = i | X) \propto P(X | c = i) \times P(c = i)$$

- **clasificatorul naïve Bayes consideră caracteristicile independente (nu e întotdeauna adevărat acest lucru)**

$$P(X | c = i) = \prod_{j=1}^n P(X_j = x_j | c = i)$$

# Clasificatorul Bayes pe cazul general

- $X$  vector de  $n$  caracteristici  $X = (X_1, X_2, \dots, X_n)$
- $c$  clase (admis/respins, litere, cifre, etc.)
- Regula de clasificare pentru Naïve Bayes:

$$c^* = \operatorname{argmax}_i \left( \prod_{j=1}^n P(X_j = x_j \mid c = i) \right) \times P(c = i)$$

trebuie estimată din date

- stabilitate numerică:

$$c^* = \operatorname{argmax}_i \left( \log \left( \prod_{j=1}^n P(X_j = x_j \mid c = i) \right) \times P(c = i) \right)$$

$$c^* = \operatorname{argmax}_i \left( \sum_{j=1}^n \log(P(X_j = x_j \mid c = i)) + \log(P(c = i)) \right)$$

# Evaluarea performanței unui model

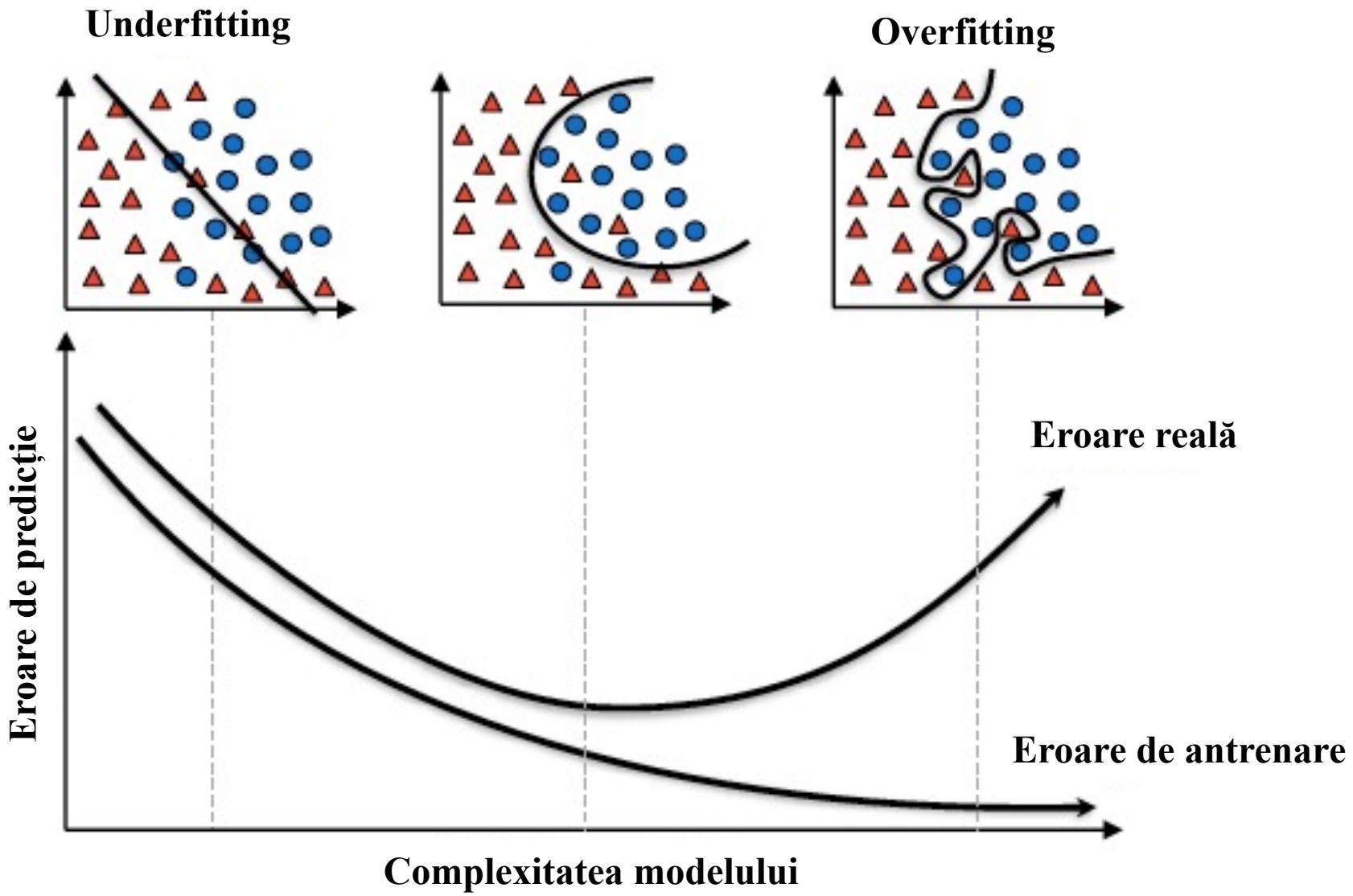
# Evaluarea unui model

- Un model este bun dacă performează bine pe date (de test) pe care nu le-a văzut înainte (la antrenare).
  - nu ține minte datele de antrenare ci are capacitatea de generalizare
  - **overfitting (supra-învățare)** – modelul performează bine pe datele de antrenare dar nu poate generaliza
  - **underfitting (sub-învățare)** – modelul performează slab și pe datele de antrenare și pe cele de test.
- Eroarea reală = eroarea pe toate datele posibile
- Eroare empirică = eroarea pe o mulțime finită de puncte
  - se mai numește și **eroare de testare, eroare de generalizare**
- Vrem să minimizăm eroarea totală, imposibil de măsurat
- Vrem să ne asigurăm că eroarea empirică aproximează eroare reală. Cum realizăm acest lucru?

# Evitarea fenomenului de overfitting

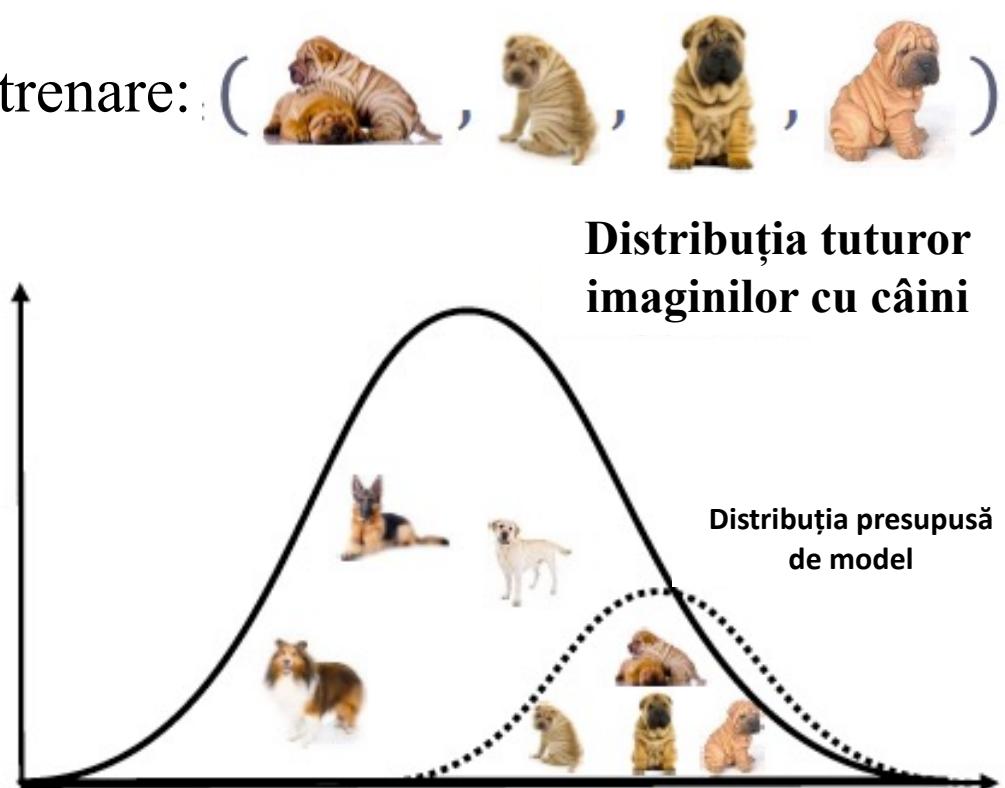
- Nu calculăm eroarea empirică pe datele folosite la antrenare
  - un set de date separat pentru testare
- Nu alegem hiperparametri modelului care să conducă la o performanță bună pe datele de test
  - un set de date separat pentru validare
  - folosim validare încrușitată (Cross Validation)
- Preferăm modele mai simple în locul celor mai complexe
  - penalizarea complexității unui model se numește **regularizare**
- Vrem să ne asigurăm că datele sunt i.i.d. (independente și identic distribuite)
  - fiecare exemplu (de antrenare sau testare) este selectat independent din aceeași distribuție
- Mai multe date!!!

# Underfitting vs. overfitting



# Date i.i.d.

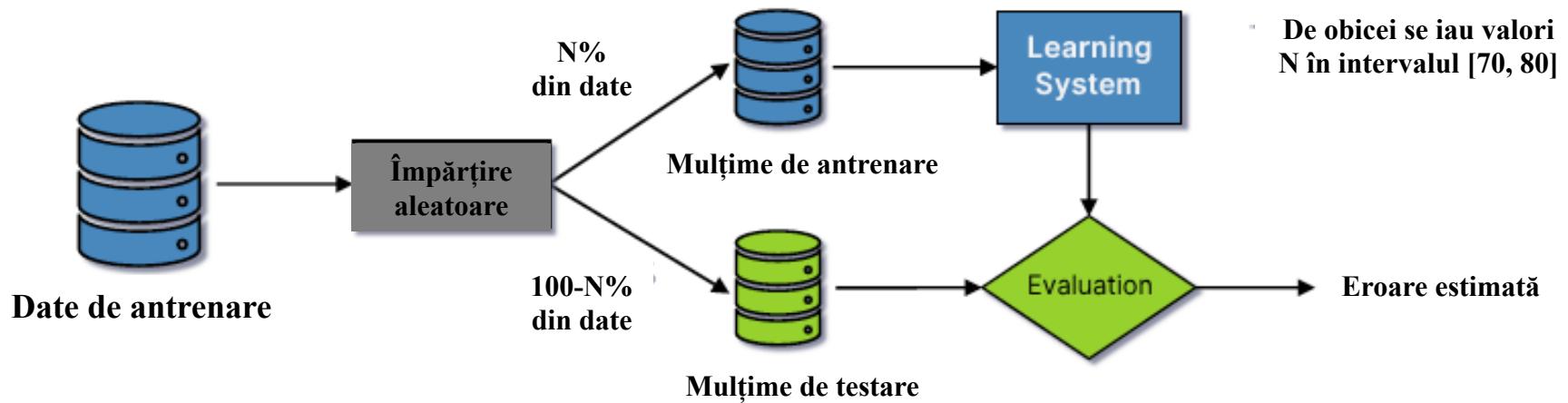
- Cei mai mulți algoritmi prespun că datele sunt independente și identic distribuite
- Presupunem că vrem să antrenăm un model care să recunoască poze cu câini
  - considerăm multimea de antrenare: ( , , ,  )
- Exemplele nu sunt i.i.d
  - cel mai probabil modelul va face greșeli pe imagini care provin din alte părți ale distribuției



# Strategii de împărțire a datelor

# Mulțime de antrenare și testare

- În practică, nu putem vedea distribuția reală a datelor și uneori nu putem avea acces la foarte multe date din această distribuție
- Așa cum am văzut, eroarea de antrenare este o estimare optimistă a erorii reale:
  - întotdeauna eroarea de antrenare va fi mai mică decât eroarea reală
  - favorizează modele complexe, care pot face overfitting pe datele de antrenare
- O posibilitate de a îmbunătăți estimarea erorii reale este împărțirea datelor în două mulțimi: antrenare și testare



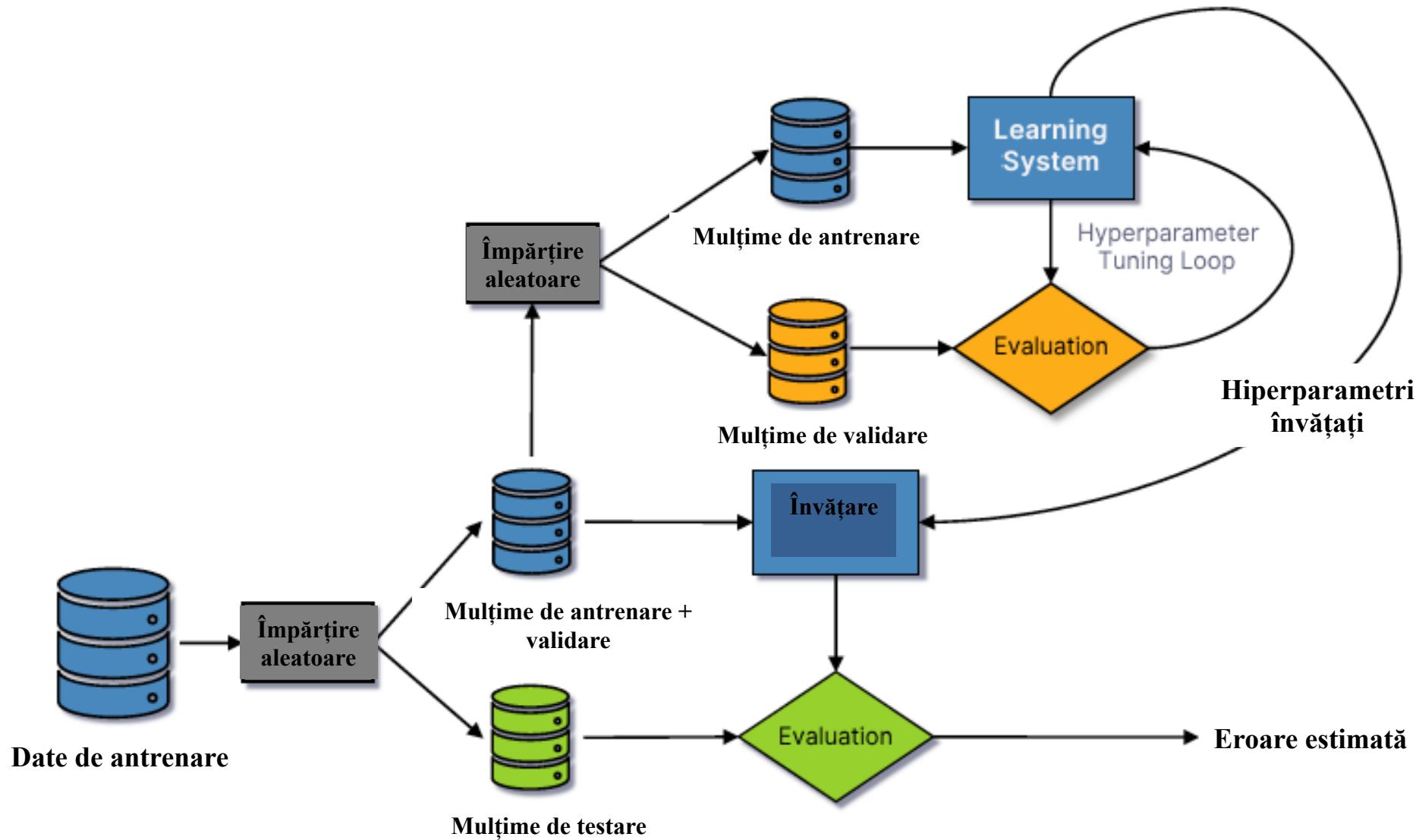
# Hiperparametri

- Unele modele au hiperparametri ce controlează procesul de antrenare

Model	Parametri	Hiperparametri
K-NN	Nu are	$K =$ numărul de vecini
Naïve Bayes	$P(X_j = x_j   c=i)$ – probabilitatea individuală de likelihood	$p$ – numărul de intervale în care împart domeniul de valori media, deviația standard unei distributii
SVM (va urma)	$w, b$ – definesc hiperplanul de separare	$C$ – controlează trade-off-ul dintre margine și acuratețe
Rețele neuronale de perceptri (va urma)	Ponderile rețelei	Arhitectura rețelei, funcții de activare, etc.

- Vreau să aleg valori pentru hiperparametri care conduc la o eroare reală mică
  - nu e bine să optimizez alegera hiperparametrilor pe mulțimea de testare (adică să fixez diverse valori pentru un hiperparametru, antrenez parametri pe mulțimea de antrenare și să aleg valoarea hiperparametrului care minimizează eroarea pe mulțimea de testare)
  - e posibil să fac overfitting în spațiul hiperparametrilor pe mulțimea de testare
  - aş vrea să nu mă ating de mulțimea de testare
- Folosesc o mulțime separată de *validare*

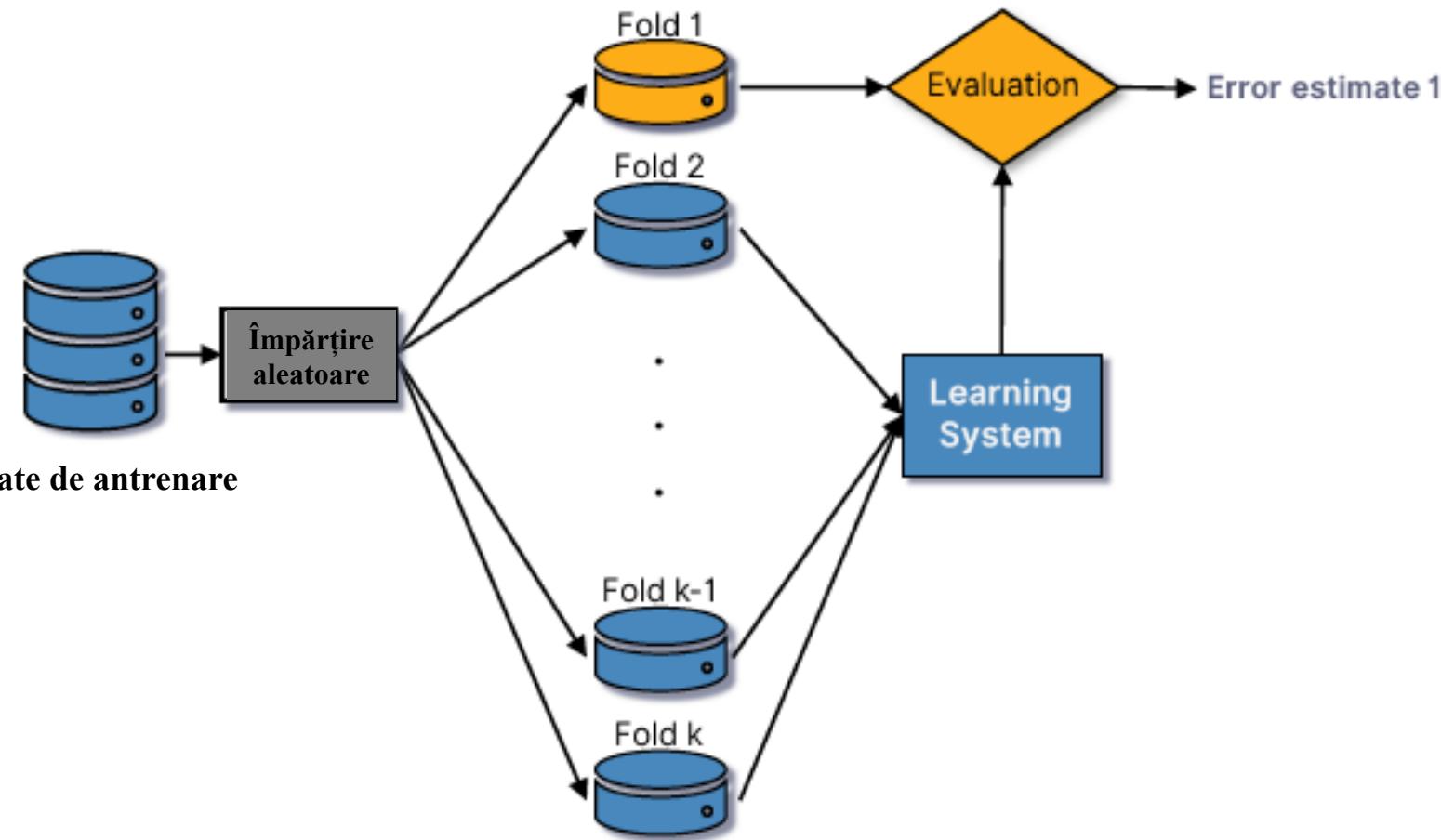
# Mulțime de antrenare, validare și testare



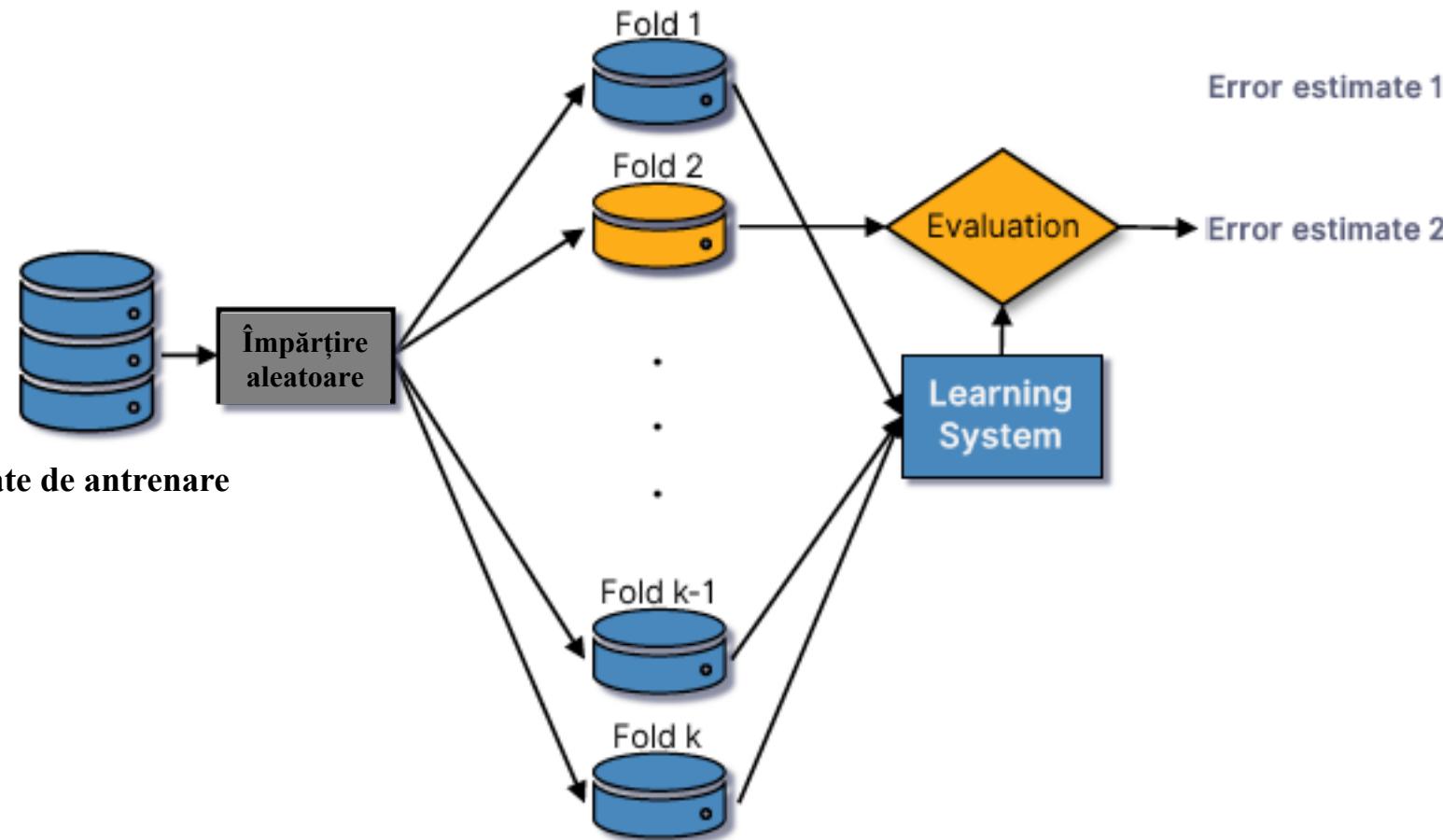
# K-fold Cross-Validation

- În practică, e posibil să nu avem atât de multe date încât să ne “permitem” o mulțime de validare
  - dacă am împărți datele în mulțime de antrenare, validare, testare am putea obține mulțimi mult prea mici
- K-fold Cross-Validation (validare încrucișată pe k părți)
  - împarte datele în k părți egale (parte = fold)
  - repetă de k-ori procesul în care antrenezi pe k-1 părți și testezi pe o parte (rămasă)
  - estimează eroarea reală ca media celor k erori de testare pe cele k părți
- Chiar dacă ne “permitem” o mulțime de validare, prin K-fold Cross Validation obținem o estimare mai bună a erorii reale
  - e nevoie de putere computațională mare

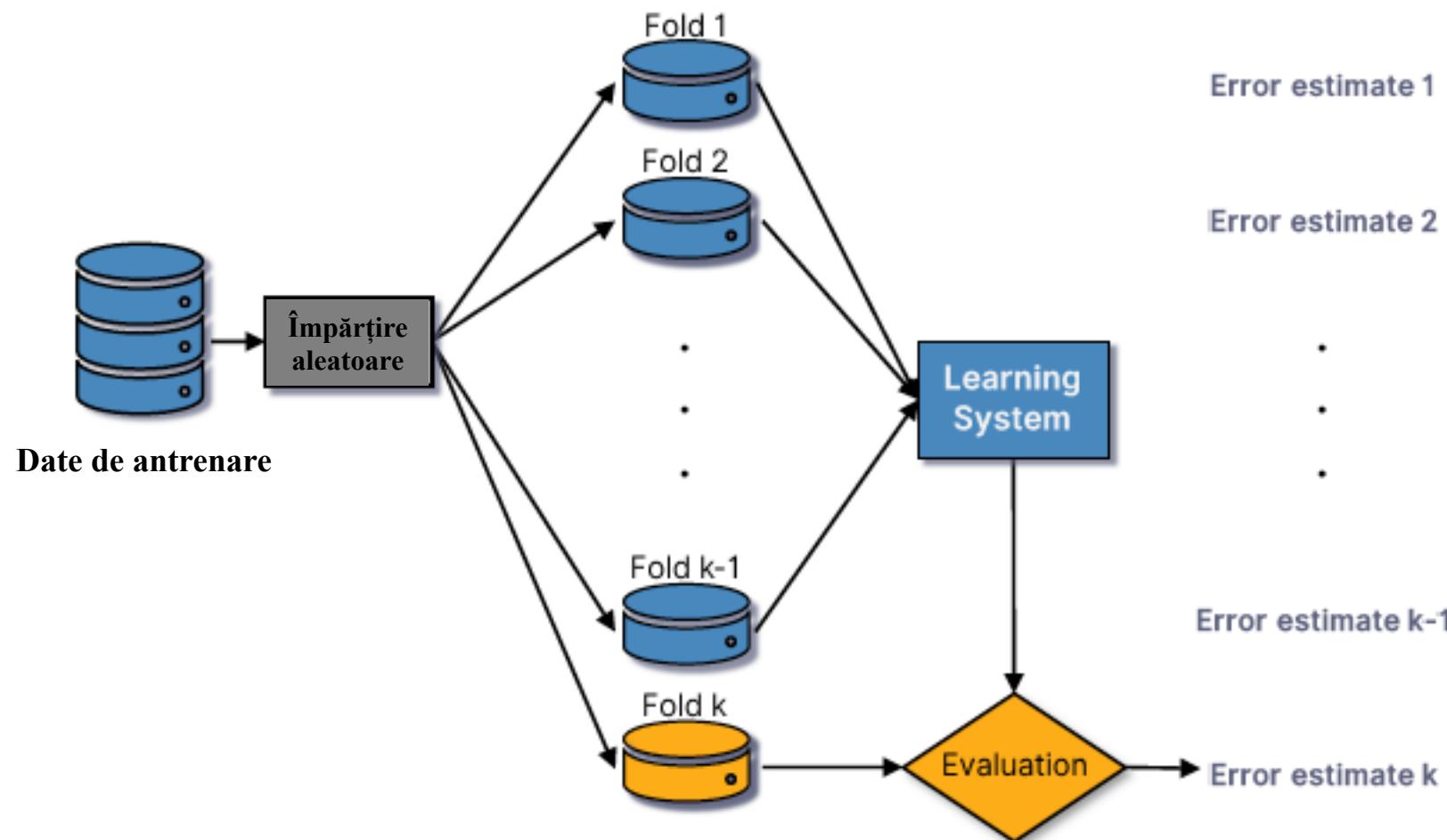
# K-fold Cross-Validation



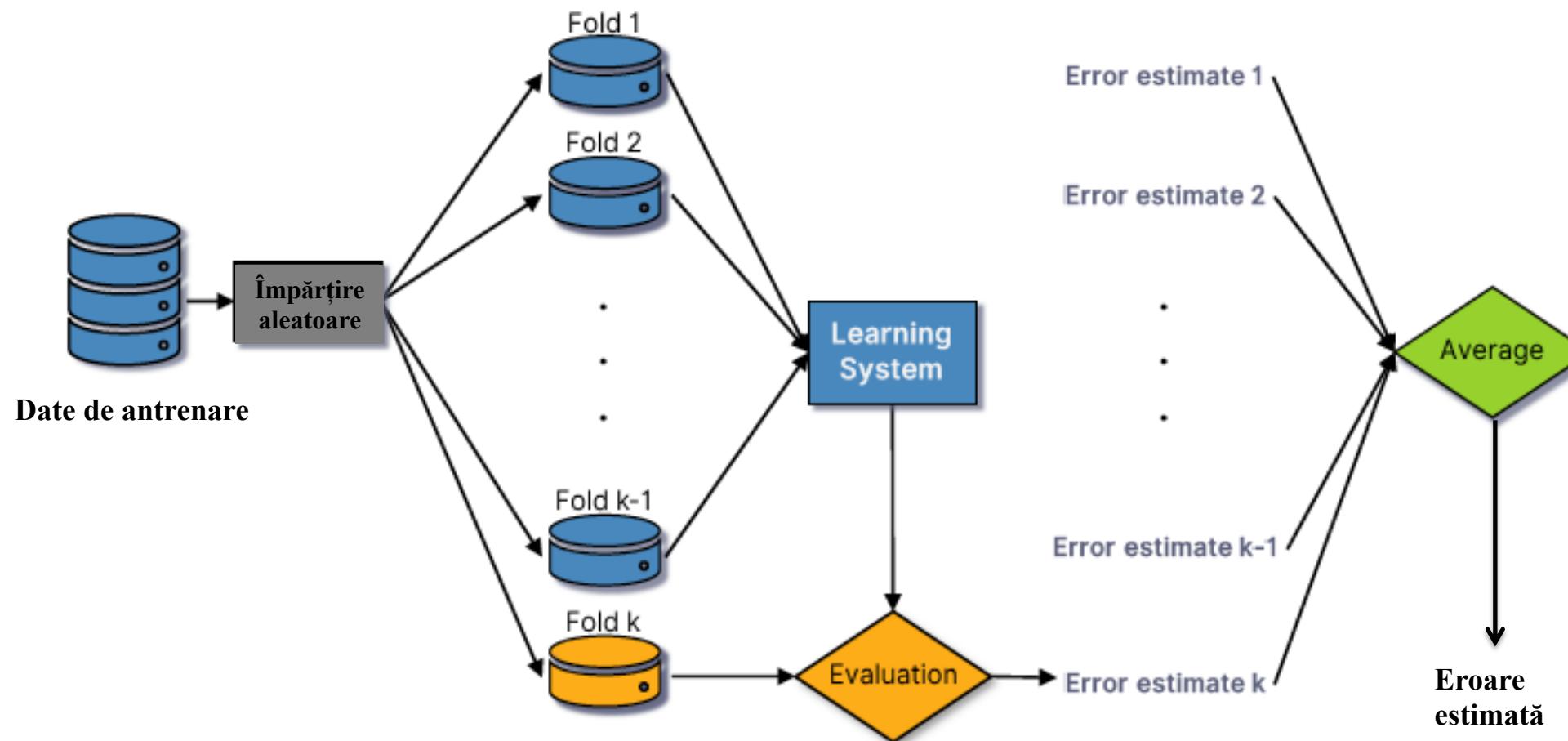
# K-fold Cross-Validation



# K-fold Cross-Validation



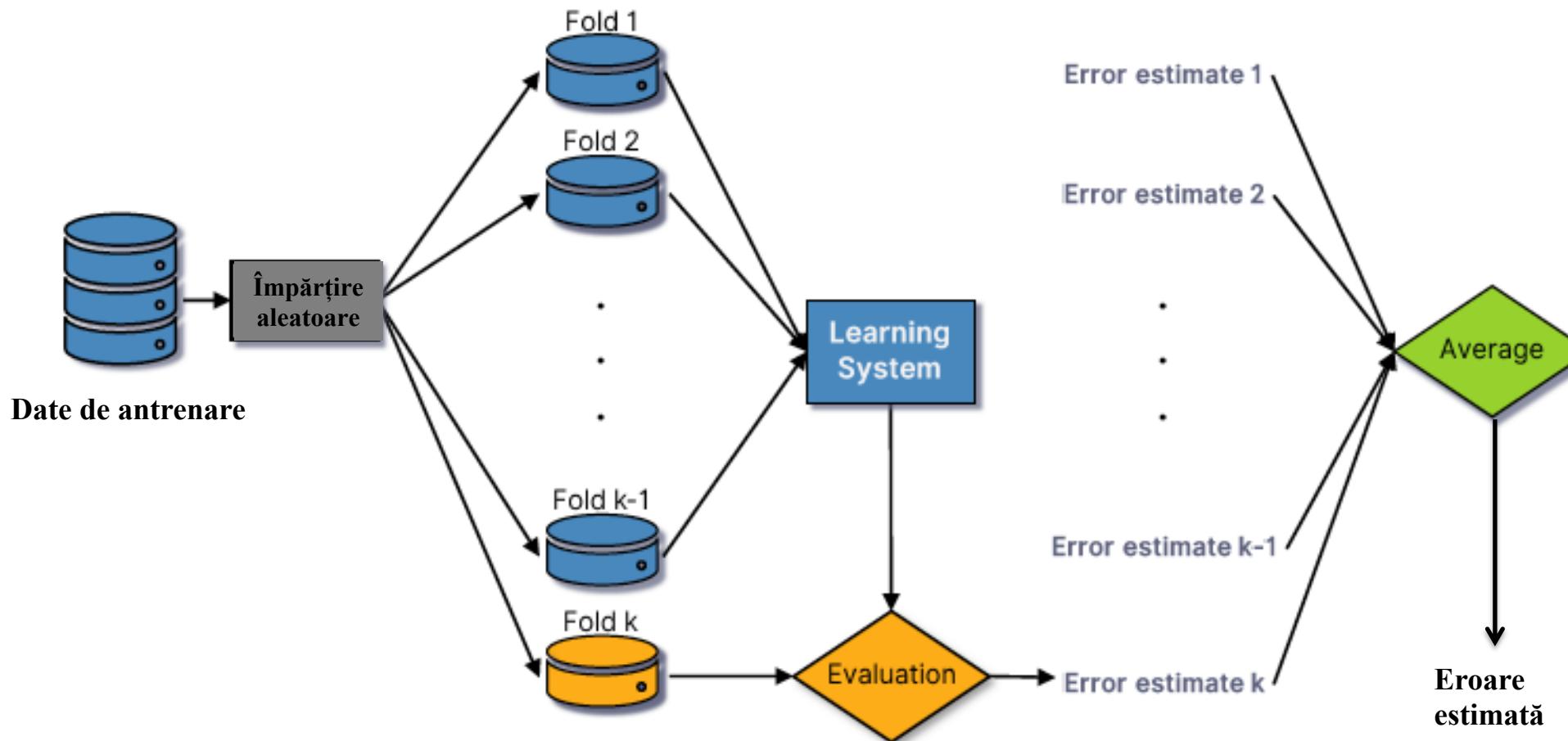
# K-fold Cross-Validation



# Leave One-Out Cross-Validation

Caz extrem: K = numărul de exemple de antrenare

Antrenăm pe K-1 exemple, testăm pe un exemplu



# Strategii de împărțire a datelor - summar

- Folosirea erorii de antrenare pentru evaluarea performanței unui model nu este o idee bună:
  - eroare de antrenare este o estimare optimistă a erorii reale
  - favorizează modele complexe care pot face overfitting
- Folosirea unui mulțimi separate de testare oferă o estimare mai bună a erorii reale
- Dacă folosim o singură mulțime test pentru alegerea hiperparametrilor putem ajunge din nou la o estimare incorectă a erorii reale
  - overfitting pe mulțimea de testare în alegerea valorilor optime a hiperparametrilor
  - soluția este să folosim o mulțime de validare
- Dacă nu avem suficiente date de antrenare sau dacă ne “permitem” din punct de vedere computațional K-Fold Cross Validation oferă o estimare mai bună a erorii reale:
  - Leave-One Out Cross Validation:  $K = \text{numărul de exemple de antrenare}$

Proiect – concurs pe platforma Kaggle

# Corpus de traduceri

- Discursuri în Parlamentul European ținute de către europarlamentari din Anglia, Scoția, Irlanda – vorbitori de limbă engleză dar fiecare cu dialectul nativ – trei clase = {England, Scotland, Ireland}
- Traduceri în cinci limbi (daneză, germană, spaniolă, italiană, olandeză) a discursurilor eurparlamentarilor din Anglia, Scoția, Irlanda
- Scopul proiectului este de a identifica dialectul sursă (englez, scoțian, irlandez) din care a fost tradus textul într-o anumită limbă
- Traducerile sunt realizate de experti vorbitori nativi ai limbii în care se face traducerea și evident vorbitori de limbă engleză

# Corpus de traduceri

		train_data	label
1	language	text	
2	dansk	<p>Dette er et fremragende initiativ, og jeg støtter fuldt ud målet om at fremhæve idrættens pædagogiske rolle. Der kan opnås store fordele ved at etablere partnerskaber mellem undervisningssektoren og idrættens organisationer.</p> <p>Som irsk medlem er jeg specielt glad for, at forslaget går ud på at dedikere 2004 til Idrættens Pædagogiske Dimension. Det falder sammen med det irske formandskab i første halvdel af 2004, og jeg kan forsikre mine kolleger, at vi alle klar over, at en række store sportsbegivenheder løber af stabelen i 2004. Der afholdes EM i fodbold og olympiske og paraolympiske lege i Athen. I forslaget anses 2003 for at være et forberedende år med visse aktiviteter.</p> <p>Ireland er vært for Special Olympics i 2003, og EU's bidrag til organiseringen af disse lege er værd at rose. For atterne fra de 160 internationale delegationer verden over er dette en enestående sportslig og kulturel begivenhed.</p>	Ireland
3	dansk	<p>Hr. formand, jeg er sikker på, at alle her er klar over, at De Grønne har grundlæggende problemer med Den Europæiske Unions behandling af fiskeriaftaler. Vi er principielt ikke imod fiskeriaftaler. Men vi protesterer stærkt mod, at jeg udarbejdede Madagaskar-betænkningen, foretog en detaljeret undersøgelse af fiskeriaftalen med Madagaskar og fandt den særdeles mangelfuld. Siden da har vi stemt imod aftalerne. Mauritius-aftalen ligner meget denne.</p> <p>Dette er grunden til, at vi sammen med EDN-gruppen og hr. Macarthy har fremsat et ændringsforslag om en midtvejsrevision af aftalen. Ved denne midtvejsrevision ser vi gerne en vurdering af EU-fiskeriets indflydelse på både landbrug og miljø.</p> <p>Jeg har et ganske enkelt spørgsmål til Kommissionen og vil meget gerne have et svar her i aften. Spørgsmålet lyder således: Hvis Parlamentet vedtager vort ændringsforslag om en uafhængig midtvejsrevision, vil Kommissionen tage det i betragtning?</p>	Ireland
4	dansk	<p>Hr. formand, folk på den nordlige halvkugle tror, at aids er et overstået kapitel. Det er naturligvis ikke tilfældet, men en skiftende livsstil og fremkomsten af effektive lægemiddelordninger har i det mindste hjulpet os med at kontrollere aids. Men på den sydlige halvkugle er billede meget anderledes. 90% af de 5 millioner nye tilfælde hvert år forekommer i de lavtlønnede lande i den sydlige verdensdel. Der findes 25 millioner mennesker med aids i Afrika og 6 millioner i Asien.</p> <p>Hvad angår aids står vi over for en ond cirkel af infektioner, arbejdssygtighed og fattigdom ud over de 16 millioner dødsfald, som vi allerede har oplevet. Der har i Afrika været 2 millioner dødsfald om året på grund af aids, hvilket er næsten dobbelt så mange som i Europa.</p> <p>Da The Economist tog til Zambia og besøgte et bestemt hospital, så de, at totredjedele af patienterne var ved at dø af aids. De rapporterede, at folks lemmer lignede knækkede kosteskifter. Når de blev spurgt om deres største bekymring, svarede de, at de ikke kunne få et bedre spisevalg. I Zambia vurderede sundhedsministeriet dengang, at halvdelen af befolkningen ville dø af aids. Det er en menneskelig katastrofe. Nu står vi over for menneskelige katastrofer i USA. Verden ser på overskrifterne, verden skrider ud af sit sædvanlige.</p> <p>Fattigdomslempelse går hånd i hånd med bekæmpelse af sygdom, invaliditet og død. Vi har desperat behov for flere ressourcer til lægermidler og vacciner, men skal også sikre os, at de i tilstrækkelig grad deles ud til folk, så de kan få adgang til dem.</p> <p>Tuberkulose dræber også. Sygdommen forårsager 2 millioner dødsfald om året, hvoraf 95% er i udviklingslandene. Det er den største årsag til hiv-dødsfald. En smittet person kan videregive den til 10 andre. Malaria er den tredje største årsag til dødsfald.</p>	England
5	dansk	<p>Hr. formand, med forbehold af nogle få ændringer støtter min gruppe helt beslutningsforslaget, som er fremlagt og udarbejdet af ordførerne, hr. Tsatsos og hr. Gil-Robles Gil-Delgado.</p> <p>Den store udfordring, som vi står over for nu, er at få forfatningen gennem regeringskonferencen, uden at den skiller ad i stumper og stykker. Derfor bifalder vi, at der i dette beslutningsforslag, som vi vedtager, ikke opfordres til at udskifte forfatningen.</p> <p>Der er skjulte farer. Der er regeringer - f.eks. Spanien - som anfægter systemet med dobbelt flertal i Rådet, og som ønsker at vende tilbage til Nice-systemet, som er kompletst, uforståeligt og ikke et særligt fornuftigt system.</p> <p>Jeg opnorer også imod dem, der ønsker henvisning til religion i forfatningen. Vi er en Union af forskellige religioner, af religiøs pluralisme. Der er mennesker i vores Union, som ikke er religiøse. Vi bør ikke gennemtrumpe et bestemt sprog.</p> <p>Til sidst vil jeg sige noget om spørgsmålet om en folkeafstemning. Det er ikke EU's opgave at fortælle medlemsstaterne, hvilken fremgangsmåde de skal anvende internt til at ratificere en traktat om en forfatning. Det er hver stats opgave.</p>	England
	dansk	<p>- Hr. formand, jeg må protestere mod den lømfærdighed, hvormed Pakistanets præsident Musharraf reagerede på Abdul Qadeer Khans tilstædelse på tv. Europas Parlamentet har tidligere taget afstand fra Pakistans økonomi af grunde til lømfærdighed.</p>	England

# Corpus de traduceri

train\_data

	language	text	label
1	dansk	Dette er et fremragende initiativ, og jeg støtter fuldt ud målet om at fremhæve idrættens pædagogiske rolle. Der kan opnås store fordele ved at etablere partnerskaber mellem undervisningssektorens og idrættens organisation Irland  Som irsk medlem er jeg specielt glad for, at forslaget går ud på at dedikere 2004 til Idrættens Pædagogiske Dimension. Det falder sammen med det irske formandskab i første halvdel af 2004, og jeg kan forsikre mine kolleger  Vi er alle klar over, at en række store sportsbegivenheder løber af stablen i 2004. Der afholdes EM i fodbold og olympiske og paralympiske lege i Athen. I forslaget anses 2003 for at være et forberedende år med visse aktiviteter.  Irland er vært for Special Olympics i 2003, og EU's bidrag til organiseringen af disse lege er værd at rose. For atterne fra de 160 internationale delegationer verden over er dette en enestående sportslig og kulturel begivenhed.	Irland
2	Deutsch	. (EN) Das ist eine ausgezeichnete Initiative, und ich begrüße das Anliegen, den erzieherischen Wert des Sports hervorzuheben, von ganzem Herzen. Die Herstellung von Partnerschaften zwischen Sportorganisationen und Bildungsministerien ist sehr wichtig. Ich freue mich über die Unterstützung Irlands für die Olympischen Spiele 2004 in Athen. Es ist eine großartige Gelegenheit, um die Erziehung durch Sport zu betonen. Ich möchte auch danken Irland für seine Unterstützung der Special Olympics im Jahr 2003.	Irland
3	Deutsch	8316 . (EN) Das ist eine ausgezeichnete Initiative, und ich begrüße das Anliegen, den erzieherischen Wert des Sports hervorzuheben, von ganzem Herzen. Die Herstellung von Partnerschaften zwischen Sportorganisationen und Bildungsministerien ist sehr wichtig. Ich freue mich über die Unterstützung Irlands für die Olympischen Spiele 2004 in Athen. Es ist eine großartige Gelegenheit, um die Erziehung durch Sport zu betonen. Ich möchte auch danken Irland für seine Unterstützung der Special Olympics im Jahr 2003.  Als irischen Abgeordneten freut es mich ganz besonders, dass das Jahr 2004 zum Jahr der Erziehung durch Sport ausgerufen werden soll. In der ersten Hälfte des Jahres 2004 wird Irland den Ratsvorsitz übernehmen, und ich freue mich auf eine erfolgreiche Zusammenarbeit mit Irland.  Wir alle wissen, dass für 2004 eine Reihe bedeutender Sportereignisse geplant sind. So werden die Fußballeuropameisterschaften sowie die Olympischen Spiele und die Paralympics in Athen stattfinden. Laut Vorschlag soll das Land Irland Gastgeber der Special Olympics im Jahr 2003 sein, und der Beitrag der EU zur Organisation dieses Ereignisses verdient es ebenfalls, hervorgehoben zu werden. Für die Sportler der 160 internationalen Delegationen ist dies eine großartige Gelegenheit, um ihre Talente zu zeigen und sich gegenseitig zu inspirieren.  Irland wird Gastgeber der Special Olympics im Jahr 2003 sein, und der Beitrag der EU zur Organisation dieses Ereignisses verdient es ebenfalls, hervorgehoben zu werden. Für die Sportler der 160 internationalen Delegationen ist dies eine großartige Gelegenheit, um ihre Talente zu zeigen und sich gegenseitig zu inspirieren.	Irland
4	Deutsch	8317 Herr Präsident, ich bin sicher, daß jedem hier bekannt ist, daß die Grünen ein grundsätzliches Problem mit der Haltung der Europäischen Union gegenüber Fischereiabkommen haben. Wir sind nicht prinzipiell gegen Fischereiabkommen eingetragen, aber wir fordern eine Änderung.  Als ich den Madagaskar-Bericht erstellt habe, haben wir das Fischereiabkommen mit Madagaskar gründlich geprüft und festgestellt, daß es erhebliche Mängel aufwies. Seither haben wir gegen solche Abkommen gestimmt. Ich kann Ihnen versichern, daß wir dies weiter tun werden.  Das ist der Grund, warum wir - zusammen mit der EDN-Faktion und Herrn Macartney - einen Änderungsantrag eingebracht haben, der eine Halbzeitbilanz des Abkommens fordert. Wir möchten, daß in dieser Halbzeitbilanz die Mängel aufgedeckt werden.  Ich möchte der Kommission eine einfache Frage stellen und hätte gerne noch heute abend eine Antwort darauf. Ich möchte gerne folgendes wissen: Wenn das Parlament heute unseren Änderungsantrag im Hinblick auf eine Änderung des Abkommens annehmen sollte, was würde dann passieren?	Irland
5	Deutsch	8318 Herr Präsident, im Norden denken die Menschen, das Aids-Problem sei überstanden. Das ist es natürlich nicht, aber veränderte Lebensgewohnheiten und neue, wirksame medikamentöse Behandlungen haben uns zumindest in den Norden gebracht. Aber im Süden zeigt sich ein völlig anderes Bild. Von den fünf Millionen neuen Fällen in jedem Jahr treten 95 % in den südlichen Ländern mit niedrigen Einkommen auf. 25 Millionen Aids-Kranke leben in Afrika und sechs Millionen in anderen Entwicklungsländern.  Was Aids betrifft, befinden wir uns bei mittlerweile 16 Millionen Toten in einem Teufelskreis von Infektion, Unvermögen und Armut. Jährlich zwei Millionen Tote in Afrika, das ist ein Viertel aller dortigen Todesfälle. Zehn Prozent der Aids-Toten sind Kinder unter 15 Jahren.  Als Mitarbeiter des Economist in Sambia ein spezielles Krankenhaus besuchten, mussten sie sehen, dass zwei Drittel der Patienten an Aids sterben. Die Gliedmaßen der Menschen glichen den Berichten zufolge zerbrochener Stäbe.  Das Gesundheitsministerium in Sambia schätzt seinerzeit ein, dass die Hälfte der Bevölkerung an Aids sterben wird. Das ist eine menschliche Katastrophe. Jetzt stehen wir in Amerika vor einer menschlichen Katastrophe. Das ist eine menschliche Katastrophe.  Armutslinderung geht mit dem Kampf gegen Krankheit, Behinderung und Tod Hand in Hand. Wir brauchen dringend mehr Mittel für Medikamente und Impfstoffe, müssen aber auch für ihre ordnungsgemäße Verteilung an die Bevölkerung sorgen.  Todbringend ist natürlich auch die Tuberkulose. Auf ihr Konto gehen jährlich zwei Millionen Todesfälle, davon 95 % in Entwicklungsländern; sie ist die Haupttodesursache bei HIV-Kranken; eine infizierte Person kann die Krankheit an 15 Personen weitergeben.	England
6	Deutsch	8319 Herr Präsident, vorbehaltlich einiger weniger Änderungen unterstützt meine Fraktion den Entschließungsantrag uneingeschränkt, der uns von den Berichterstattern Herrn Tsatsos und Herrn Gil-Robles Gil-Delgado vorgelegt wurde. Ich kann Ihnen versichern, daß wir diesen Antrag unterstützen werden.  Wir stehen nun vor der großen Herausforderung, die Verfassung durch die Regierungskonferenz zu bekommen, ohne dass sie Stück für Stück demonstriert wird. Deshalb begrüßen wir es, dass in diesem Entschließungsantrag eine klare Position bezogen wird.  Doch es lauern Gefahren. Es gibt Regierungen, wie zum Beispiel die spanische, die das System der doppelten Mehrheit im Rat ablehnen und zu dem komplizierten, unverständlichen und wenig sinnvollen System von Nizza zurückgreifen wollen.  Ich stimme auch nicht mit denjenigen überein, die einen Verweis auf die Religion in die Verfassung aufnehmen wollen. Wir sind eine Union des religiösen Pluralismus, in der verschiedene Religionen vertreten sind. In unserer Union ist die Religionsfreiheit eine wichtige Werte.  Abschließend möchte ich noch auf das Thema Volkbefragung eingehen. Es steht der Europäischen Union nicht zu, ihren Mitgliedstaaten vorzuschreiben, welche internen Verfahren sie zur Ratifizierung eines Vertrags anwenden.	England

# Corpus de traduceri

train\_data

	language	text	label
1	dansk	Dette er et fremragende initiativ, og jeg støtter fuldt ud målet om at fremhæve idrættens pædagogiske rolle. Der kan opnås store fordele ved at etablere partnerskaber mellem undervisningssektoren og idrættens organisationer. Irland Som irsk medlem er jeg specielt glad for, at forslaget går ud på at dedikere 2004 til Idrættens Pædagogiske Dimension. Det falder sammen med det irske formandskab i første halvdel af 2004, og jeg kan forsikre mine kolleger Vi er alle klar over, at en række store sportsbegivenheder løber af stablen i 2004. Der afholdes EM i fodbold og olympiske og paralympiske lege i Athen. I forslaget anses 2003 for at være et forberedende år med visse aktiviteter. Irland er vært for Special Olympics i 2003, og EU's bidrag til organiseringen af disse lege er værd at rose. For atterne fra de 160 internationale delegationer verden over er dette en enestående sportslig og kulturel begivenhed.	Ireland
2	Deutsch	. (EN) Das ist eine ausgezeichnete Initiative, und ich begrüße das Anliegen, den erzieherischen Wert des Sports hervorzuheben, von ganzem Herzen. Die Herstellung von Partnerschaften zwischen Sportorganisationen und Bildungsinstanzen ist ein wichtiger Beitrag zu einer besseren Zusammenarbeit. Irland Als irischen Abgeordneten freut es mich ganz besonders, dass das Jahr 2004 zum Jahr der Erziehung durch Sport ausgerufen werden soll. In der ersten Hälfte des Jahres 2004 wird Irland den Ratsvorsitz übernehmen, und ich freue mich auf die Zusammenarbeit mit anderen Mitgliedsstaaten. Wir alle wissen, dass für 2004 eine Reihe bedeutender Sportereignisse geplant sind. So werden die Fußballeuropameisterschaften sowie die Olympischen Spiele und die Paralympics in Athen stattfinden. Laut Vorschlag soll das Land Irland Gastgeber der Special Olympics im Jahr 2003 sein, und der Beitrag der EU zur Organisation dieses Ereignisses verdient es ebenfalls, hervorgehoben zu werden. Für die Sportler der 160 internationalen Delegationen ist dies eine großartige Gelegenheit.	Ireland
3	español	. (EN) Ésta es una excelente iniciativa y yo suscribo plenamente el objetivo de destacar el valor educativo del deporte. Puede ganarse mucho con el establecimiento de colaboraciones entre las organizaciones deportivas y las instituciones de educación. Irlanda Como diputado procedente de Irlanda, me complace particularmente que esta propuesta dedique el año 2004 a la Educación a través del Deporte. Esto coincidirá con la presidencia irlandesa de la Unión en el primer semestre de 2004. Los países miembros sabemos que para 2004 habrá una serie de eventos deportivos importantes. Se van a celebrar el Campeonato Europeo de Fútbol y los Juegos Olímpicos y Paralímpicos de Atenas. La propuesta prevé que Irlanda será la anfitriona de los Juegos Olímpicos Especiales en 2003, y es digna de alabanza también la aportación de la UE a la organización de este acontecimiento. Será una experiencia cultural y deportiva única para los 160 delegados internacionales.	Ireland
4	español	Señor Presidente, seguro que todos los presentes saben que el enfoque de la Unión Europea por lo que respecta a los Acuerdos pesqueros plantea algunos problemas fundamentales para los Verdes. No somos contrarios a la pesca sostenible. Irlanda Cuando elaboré el informe sobre Madagascar, realizamos un análisis detallado del Acuerdo pesquero con dicho país y detectamos serias insuficiencias en el mismo. Desde entonces hemos votado en contra de dichos Acuerdos pesqueros en la Asamblea. Por esto hemos presentado, conjuntamente con el Grupo I-EDN y con el Sr. Macartney, una enmienda por la que se solicita una evaluación intermedia del Acuerdo. Nuestro deseo es que esta revisión examine el impacto de la pesca sostenible en Madagascar. Yo quisiera plantear una pregunta muy sencilla a la Comisión y desearía recibir una respuesta esta noche. Mi pregunta es: ¿aceptaría la Comisión una revisión de este tipo si el Parlamento aprueba nuestra enmienda por la que se solicita una evaluación intermedia?	Ireland
5	español	Señor Presidente, en el Norte la gente piensa que en cuanto al SIDA todo se ha acabado. Evidentemente no es así, pero los cambios en los estilos de vida y la llegada de tratamientos con fármacos eficaces nos han ayudado. Irlanda Pero en el Sur, la imagen es muy distinta. El noventa y cinco por ciento de los cinco millones de nuevos casos anuales se dan en países del Sur con baja renta. Hay 25 millones de personas que padecen el SIDA en África y 600 mil en Asia. Irlanda En lo que respecta al SIDA nos enfrentamos a un círculo vicioso de infección, incapacidad y pobreza que acompaña a los 16 millones de muertes a las que ya hemos asistido: dos millones de muertos al año en África, una cifra similar en Asia. Irlanda Cuando The Economist estuvo en Zambia y analizó un hospital en concreto, comprobó que dos tercios de sus pacientes morían de SIDA. Según contaban, los miembros de estas personas parecían palos de escoba rotos. Irlanda En Zambia el departamento de sanidad estimaba entonces que la mitad de la población moriría de SIDA. Esto es un desastre humano. Estamos asistiendo ahora a desastres humanos en EE.UU. El mundo está pendiente de lo que sucede en Zambia. Irlanda La reducción de la pobreza está estrechamente relacionada con la guerra contra la enfermedad, la discapacidad y la muerte. Necesitamos desesperadamente más recursos para fármacos y vacunas, pero debemos garantizar que estos recursos lleguen a las personas más necesitadas. Irlanda La tuberculosis, por supuesto, también mata. A ella se deben dos millones de muertes anuales, el 95% de éstas en países en desarrollo; es la causa principal de muerte en los casos de VIH; una persona infectada puede contagiar a otros.	England
6	español	Señor Presidente, aunque con unas pocas enmiendas, mi Grupo apoya plenamente la propuesta de resolución que se nos ha presentado, y que han elaborado con tanto acierto los ponentes, el Sr. Tsatsos y el Sr. Gil-Robles. Irlanda El gran reto a que nos enfrentamos ahora es que la Constitución sea aprobada en la CIG sin desmantelarla pieza a pieza. Por este motivo, nos complace que esta propuesta de resolución que adoptaremos no pida cambios drásticos. Irlanda Pero nos acechan algunos peligros. Algunos gobiernos -España, por ejemplo- se oponen al sistema de doble mayoría en el Consejo y quieren volver al sistema de Niza, que es un sistema complejo, difícil de entender y no muy transparente. Irlanda También me opongo a quienes desean hacer referencia a la religión en la Constitución. Somos una Unión de diversas religiones, de pluralismo religioso. En la Unión hay personas que no son religiosas. En la Constitución no debe haber referencias a la religión. Irlanda Finalmente, quiero tratar el tema del referendo. No está en manos de la Unión Europea decir a sus Estados miembros qué procedimiento deberían utilizar internamente para ratificar un tratado constitucional. Es un asunto que	England

# Corpus de traduceri

		train_data	label
1	language	text	
2	dansk	Dette er et fremragende initiativ, og jeg støtter fuldt ud målet om at fremhæve idrættens pædagogiske rolle. Der kan opnås store fordele ved at etablere partnerskaber mellem undervisningssektorens og idrættens organisationer. Ireland  Som irsk medlem er jeg specielt glad for, at forslaget går ud på at dedikere 2004 til Idrættens Pædagogiske Dimension. Det falder sammen med det irske formandskab i første halvdel af 2004, og jeg kan forsikre mine kolleger, at vi alle klar over, at en række store sportsbegivenheder løber af stablen i 2004. Der afholdes EM i fodbold og olympiske og paralympiske lege i Athen. I forslaget anses 2003 for at være et forberedende år med visse aktiviteter. Irland er vært for Special Olympics i 2003, og EU's bidrag til organiseringen af disse lege er værd at rose. For atterne fra de 160 internationale delegationer verden over er dette en enestående sportslig og kulturel begivenhed.	Ireland
3	Deutsch	. (EN) Das ist eine ausgezeichnete Initiative, und ich begrüße das Anliegen, den erzieherischen Wert des Sports hervorzuheben, von ganzem Herzen. Die Herstellung von Partnerschaften zwischen Sportorganisationen und Bildungsinstanzen ist ein wichtiger Beitrag zur Entwicklung des Sports in Irland.  Als irischen Abgeordneten freut es mich ganz besonders, dass das Jahr 2004 zum Jahr der Erziehung durch Sport ausgerufen werden soll. In der ersten Hälfte des Jahres 2004 wird Irland den Ratsvorsitz übernehmen, und ich kann mich auf die Arbeit mit dem irischen Präsidenten freuen.  Wir alle wissen, dass für 2004 eine Reihe bedeutender Sportereignisse geplant sind. So werden die Fußballeuropameisterschaften sowie die Olympischen Spiele und die Paralympics in Athen stattfinden. Laut Vorschlag soll das Land Irland Gastgeber der Special Olympics im Jahr 2003 sein, und der Beitrag der EU zur Organisation dieses Ereignisses verdient es ebenfalls, hervorgehoben zu werden. Für die Sportler der 160 internationalen Delegationen ist dies eine großartige Gelegenheit.  Ireland	Ireland
4	español	. (EN) Ésta es una excelente iniciativa y yo suscribo plenamente el objetivo de destacar el valor educativo del deporte. Puede ganarse mucho con el establecimiento de colaboraciones entre las organizaciones deportivas y las instituciones educativas. Irlanda es un país que apuesta por el deporte y su desarrollo social.  Como diputado procedente de Irlanda, me complace particularmente que esta propuesta dedique el año 2004 a la Educación a través del Deporte. Esto coincidirá con la presidencia irlandesa de la Unión en el primer semestre de 2004.  Todos somos conscientes de que en 2004 tendrá lugar una serie de eventos deportivos importantes. Se van a celebrar el Campeonato Europeo de Fútbol y los Juegos Olímpicos y Paralímpicos de Atenas. La propuesta prevé que Irlanda sea la anfitriona de los Juegos Olímpicos Especiales en 2003, y es digna de alabanza también la aportación de la UE a la organización de este acontecimiento. Será una experiencia cultural y deportiva única para los países europeos.  Irlanda	Irlanda
5	italiano	Si tratta di un'eccellente iniziativa e condivido totalmente l'obiettivo di mettere in rilievo il valore educativo dello sport. Costruire un partenariato tra organizzazioni sportive e istituti scolastici può dare molti frutti.  In qualità di deputato irlandese sono particolarmente lieto della proposta di designare il 2004 l'Anno europeo dell'educazione tramite lo sport perché andrà a coincidere con la Presidenza irlandese dell'Unione nella prima metà dell'anno.  Come è noto nel corso del 2004 si svolgeranno importanti manifestazioni sportive: il Campionato mondiale di calcio e i Giochi olimpici e paralimpici ad Atene. La proposta prevede che il 2003 sia un anno preparatorio nel quale Irlanda ospiterà le Olimpiadi speciali del 2003 e il contributo dell'Unione europea all'organizzazione di tale evento è degno di nota. Si tratterà di un'esperienza culturale e sportiva unica per gli atleti partecipanti provenienti da tutti i paesi europei.  Irlanda	Irlanda
6	italiano	Signor Presidente, tutti i presenti sanno certamente che i verdi non condividono l'operato dell'Unione europea in materia di accordi di pesca. Non siamo contrari per principio agli accordi di pesca ma avanziamo forti riserve sulla loro efficacia.  Quando mi occupavo della relazione sul Madagascar, avevamo fatto uno studio dettagliato sull'accordo di pesca con tale paese e lo avevamo trovato gravemente carente. Da allora abbiamo votato contro tali accordi. L'accordo di pesca con il Madagascar è stato respinto.  Per questo motivo, assieme al gruppo EDN e all'onorevole Macartney, abbiamo proposto un emendamento inteso ad ottenere una revisione intermedia dell'accordo. Vorremmo che essa esaminasse l'impatto della pesca dell'Unione europea sulle attività peschistiche del Madagascar.  Desidero rivolgere una semplice domanda alla Commissione e mi auguro che mi venga fornita una risposta questa sera. La domanda è la seguente: qualora il Parlamento approvasse il nostro emendamento a favore di una revisione intermedia dell'accordo di pesca con il Madagascar, quale sarebbe il vostro voto? Irlanda	Irlanda
7	italiano	Signor Presidente, nel Nord del mondo si pensa che l'AIDS sia un problema ormai risolto; naturalmente non è così, ma i cambiamenti nelle abitudini di vita e l'introduzione di nuovi farmaci ci hanno almeno consentito di compiere progressi importanti.  Nel Sud del mondo però il quadro è completamente differente: il 95 percento dei cinque milioni di nuovi casi che si registrano ogni anno si verificano nei paesi a basso reddito del Sud. In Africa vi sono 25 milioni di persone affette da AIDS.  Nel caso dell'AIDS ci troviamo di fronte a un circolo vizioso di infezione, incapacità e povertà che spiega i sedici milioni di morti che già dobbiamo lamentare; in Africa, in particolare, si contano due milioni di morti all'anno, ossia più di quelli causati dalla tubercolosi.  Quando gli inviati di The Economist nello Zambia hanno visitato un ospedale, hanno constatato che due terzi dei pazienti stavano morendo di AIDS; secondo la testimonianza dei giornalisti, le membra delle persone ricoverate erano talmente gonfie che non potevano più muoversi.  Secondo le stime allora avanzate dal Ministero della sanità dello Zambia, è possibile che metà della popolazione del paese muoia di AIDS; si tratta di una terribile tragedia umanitaria. Assistiamo ora ad analoghe tragedie anche in altri paesi.  La riduzione della povertà si accompagna alla lotta contro le malattie, l'invalidità e la morte. Abbiamo un disperato bisogno di maggiori risorse per farmaci e vaccini, ma dobbiamo anche assicurarne l'adeguata distribuzione fra i paesi più poveri.  Anche la tubercolosi è naturalmente un flagello mortale: miete due milioni di vittime l'anno, il 95 percento delle quali nei paesi in via di sviluppo, ed è la causa principale delle morti da HIV poiché una persona infetta può trasmettere il virus a molti altri. England	England
8	italiano	Signor Presidente, fatti salvi alcuni emendamenti, il mio gruppo sostiene pienamente la proposta di risoluzione presentata ed abilmente elaborata dai relatori, onorevoli Tsatsos e Gil-Robles Gil-Delgado.  La grande sfida che si presenta ora è far passare la Costituzione attraverso la CIG senza che venga demolita pezzo per pezzo. Questo è il motivo per cui siamo compiaciuti del fatto che la proposta di risoluzione che adotteremo sarà molto più completa e meno frammentaria.  Vi sono pericoli in agguato. Alcuni governi - la Spagna, per esempio - si oppongono al sistema della doppia maggioranza in seno al Consiglio e vogliono tornare al sistema di Nizza, che è complesso, incomprensibile e non molto trasparente.  Sono anche in disaccordo con coloro che vogliono includere nella Costituzione un riferimento alla religione. Siamo un'Unione di diverse religioni, di pluralismo religioso. Nell'Unione vi sono persone che non sono religiose. Non crediamo che la Costituzione debba avere nulla a che fare con la religione.  Infine, la questione del referendum. Non spetta all'Unione europea dire ai suoi Stati membri quale procedura seguire internamente per la ratifica di un Trattato che istituisce la Costituzione. E' una decisione che devono prendere i singoli Stati membri. England	England

# Corpus de traduceri

train\_data

language	text	label
dansk	Dette er et fremragende initiativ, og jeg støtter fuldt ud målet om at fremhæve idrættens pædagogiske rolle. Der kan opnås store fordele ved at etablere partnerskaber mellem undervisningssektoren og idrættens organisationer. Irland Som irsk medlem er jeg specielt glad for, at forslaget går ud på at dedikere 2004 til Idrættens Pædagogiske Dimension. Det falder sammen med det irske formandskab i første halvdel af 2004, og jeg kan forsikre mine kolleger Vi er alle klar over, at en række store sportsbegivenheder løber af stablen i 2004. Der afholdes EM i fodbold og olympiske og paralympiske lege i Athen. I forslaget anses 2003 for at være et forberedende år med visse aktiviteter. Irland er vært for Special Olympics i 2003, og EU's bidrag til organiseringen af disse lege er værd at rose. For atterne fra de 160 internationale delegationer verden over er dette en enestående sportslig og kulturel begivenhed.	Ireland
Deutsch	. (EN) Das ist eine ausgezeichnete Initiative, und ich begrüße das Anliegen, den erzieherischen Wert des Sports hervorzuheben, von ganzem Herzen. Die Herstellung von Partnerschaften zwischen Sportorganisationen und Bildungsinstanzen ist ein wichtiger Beitrag zu einer erfolgreichen Ausbildung. Irland Als irischen Abgeordneten freut es mich ganz besonders, dass das Jahr 2004 zum Jahr der Erziehung durch Sport ausgerufen werden soll. In der ersten Hälfte des Jahres 2004 wird Irland den Ratsvorsitz übernehmen, und ich freue mich auf die Zusammenarbeit mit anderen Mitgliedsstaaten. Wir alle wissen, dass für 2004 eine Reihe bedeutender Sportereignisse geplant sind. So werden die Fußballeuropameisterschaften sowie die Olympischen Spiele und die Paralympics in Athen stattfinden. Laut Vorschlag soll das Jahr 2004 als Jahr der Erziehung durch Sport ausgerufen werden. Irland wird Gastgeber der Special Olympics im Jahr 2003 sein, und der Beitrag der EU zur Organisation dieses Ereignisses verdient es ebenfalls, hervorgehoben zu werden. Für die Sportler der 160 internationalen Delegationen ist dies eine großartige Gelegenheit.	Ireland
español	. (EN) Esta es una excelente iniciativa y yo suscribo plenamente el objetivo de destacar el valor educativo del deporte. Puede ganarse mucho con el establecimiento de colaboraciones entre las organizaciones deportivas y las instituciones educativas. Irland Como diputado procedente de Irlanda, me complace particularmente que esta propuesta dedique el año 2004 a la Educación a través del Deporte. Esto coincidirá con la presidencia irlandesa de la Unión en el primer semestre. Todos somos conscientes de que en 2004 tendrá lugar una serie de eventos deportivos importantes. Se van a celebrar el Campeonato Europeo de Fútbol y los Juegos Olímpicos y Paralímpicos de Atenas. La propuesta prevé que Irlanda será la anfitriona de los Juegos Olímpicos Especiales en 2003, y es digna de alabanza también la aportación de la UE a la organización de este acontecimiento. Será una experiencia cultural y deportiva única para los países miembros.	Ireland
italiano	Si tratta di un'eccellente iniziativa e condivido totalmente l'obiettivo di mettere in rilievo il valore educativo dello sport. Costruire un partenariato tra organizzazioni sportive e istituti scolastici può dare molti frutti. In qualità di deputato irlandese sono particolarmente lieto della proposta di designare il 2004 l'Anno europeo dell'educazione tramite lo sport perché andrà a coincidere con la Presidenza irlandese dell'Unione nella prima metà dell'anno. Come è noto nel corso del 2004 si svolgeranno importanti manifestazioni sportive: il Campionato mondiale di calcio e i Giochi olimpici e paralimpici ad Atene. La proposta prevede che il 2003 sia un anno preparatorio nel quale Irlanda ospiterà le Olimpiadi speciali del 2003 e il contributo dell'Unione europea all'organizzazione di tale evento è d'emozione di nota. Si tratterà di un'esperienza culturale e sportiva unica per gli atleti partecipanti provenienti da tutti i paesi.	Ireland
Nederland	Dit is een uitstekend initiatief en ik sta volledig achter de doelstelling om speciaal de aandacht te vestigen op de educatieve waarde van de sport. Er valt veel te winnen door partnerschappen tussen sportorganisaties en onderwijsinstellingen. Als parlementslid uit Ierland ben ik bijzonder blij met dit voorstel om in het jaar 2004 speciaal aandacht te schenken aan opvoeding door sport. Dit valt in de eerste helft van het jaar samen met het eerste voorzitterschap van de Unie. Wij weten allen dat er in 2004 een aantal belangrijke sportevenementen zullen plaatsvinden, zoals de Europese voetbalkampioenschappen en ook de Olympische Spelen en de Paralympics in Athene. Volgens het voorstel wordt Ierland zal in 2003 gastheer zijn voor de Speciale Olympische Spelen in 2003. De bijdrage van de EU aan de organisatie van dit evenement verdient ook onze lof. Dit belooft voor de deelnemende sportlieden, die uitkomen in verschillende disciplines, een fantastische ervaring te zijn.	Ireland
Nederland	Mijnheer de Voorzitter, ik weet zeker dat iedereen hier weet dat de Groenen fundamentele problemen hebben met de manier waarop de Europese Unie visserijovereenkomsten benadert. Wij zijn niet tegen visserijovereenkomsten in principe. Toen ik aan het verslag over Madagascar werkte, hebben wij een gedetailleerde studie over de visserijovereenkomst met Madagascar gemaakt, en deze overeenkomst kreeg een dikke onvoldoende. Sedertdien hebben we te maken met de negatieve effecten van deze overeenkomst. Daarom hebben wij, samen met de fractie van onafhankelijkheden voor het Europa van de Nationale Staten en de heer Macartney, een amendement voorgesteld waarin gevraagd wordt om halverwege de looptijd van de overeenkomst een evaluatieverslag te presenteren. Ik heb een zeer simpele vraag voor de Commissie en zou graag vanavond een antwoord willen hebben. Ik zou willen weten of, als het Parlement ons amendement voor een evaluatieverslag halverwege de looptijd van de overeenkomst goedkeurt, wij kunnen rekenen op een positieve reactie van de Commissie.	Ireland
Nederland	Mijnheer de Voorzitter, in het Noorden denken mensen dat aids alweer achter de rug is. Dat is natuurlijk niet het geval. Andere leefwijzen en de komst van doeltreffende therapieën hebben er echter in ieder geval toe bijgedragen dat de aantal nieuwe gevallen in het Noorden nu veel minder is. In het Zuiden is het beeld evenwel heel anders. Van de 5 miljoen nieuwe gevallen die er jaarlijks bij komen, nemen lage-inkomenslanden in het Zuiden 95 procent voor hun rekening. Er leven 25 miljoen mensen met aids in Afrika. Waar het aids betreft, worden wij geconfronteerd met een vicieuze cirkel van infectie, arbeidsongeschiktheid en armoede. Daarnaast zijn er ook nog de 16 miljoen doden die we al gezien hebben: 2 miljoen doden per jaar in Afrika. Toen verslaggevers van The Economist een bepaald ziekenhuis in Zambia bezochten, ontdekten zij dat tweederde van de patiënten aldaar aan aids stierven. De ledematen van mensen, zo meldden zij, zagen eruit als gebrokken stokken. In Zambia schatte het ministerie van Volksgezondheid destijds dat de helft van de bevolking aan aids zou sterven. Dat is een humanitaire ramp. We worden op dit moment met humanitaire rampen in Amerika geconfronteerd. Armoedebestrijding gaat hand in hand met de oorlog tegen ziekte, invaliditeit en dood. We hebben dringend meer geld nodig voor geneesmiddelen en vaccins. We moeten er echter ook voor zorgen dat deze op geschikte wijze worden gebruikt. Tuberculose is uiteraard ook moordend. Deze ziekte is verantwoordelijk voor 2 miljoen doden per jaar, waarvan 95 procent in ontwikkelingslanden. Zij is de belangrijkste doodsoorzaak bij HIV-gedetecteerden. Een besmette persoon kan de ziekte overbrengen aan 10 andere mensen.	England
Nederland	Mijnheer de Voorzitter, behoudens enkele amendementen staat mijn fractie vierkant achter de ontwerpresolutie die de rapporteurs, de heer Tsatsos en de heer Gil-Robles Gil-Delgado, ons hebben voorgelegd en die zij zo voor ons hebben voorbereid.	England

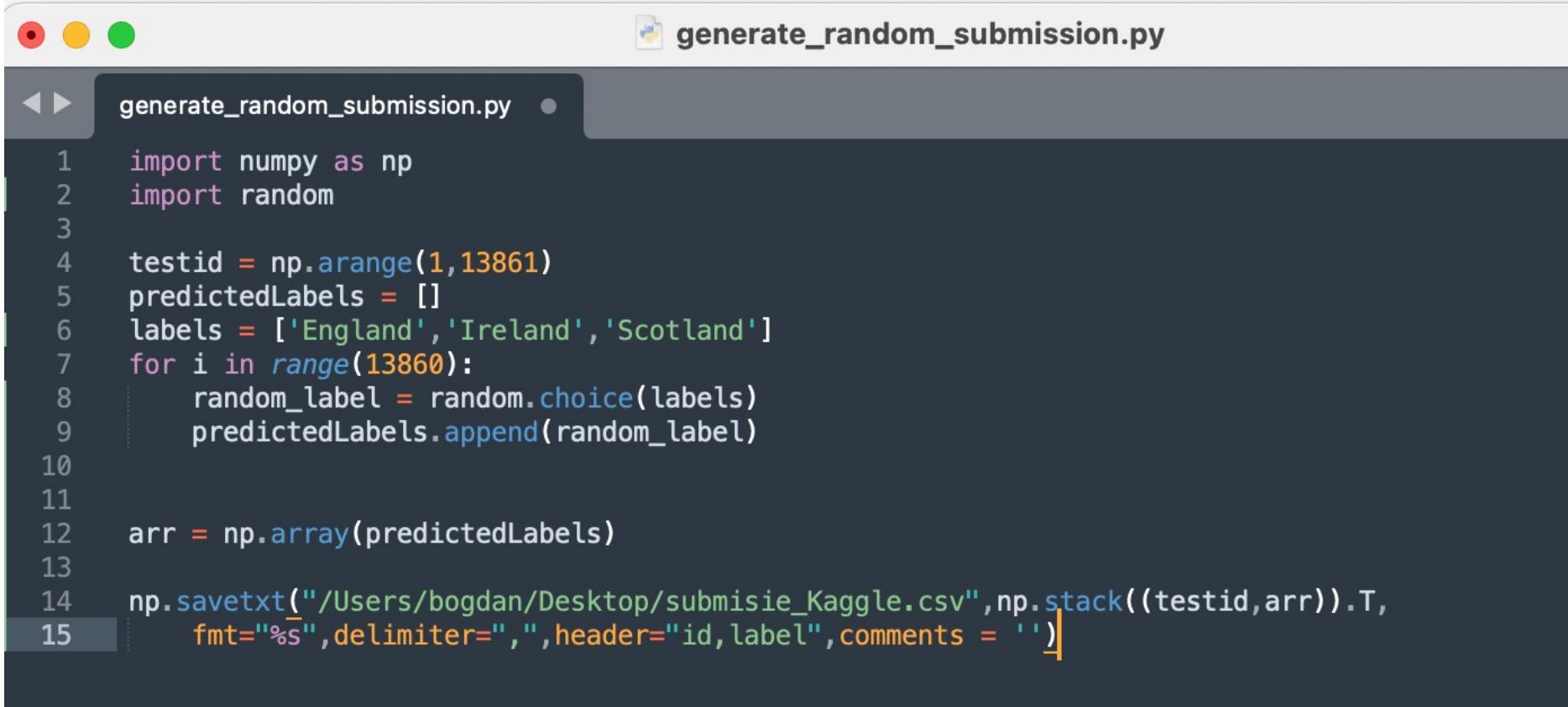
# Date proiect

- Antrenare:
  - 41570 exemple de antrenare = texte cu traduceri, ordonate câte 8314 în funcție de limba traducerii (limba traducerii este etichetată)
  - 41570 exemple de antrenare =  $8314 \text{ texte traduse} \times 5 \text{ limbi străine}$  (daneză, germană, spaniolă, italiană, olandeză)
  - fiecare exemplu de antrenare este etichetat cu una din clasele {England, Scotland, Ireland}
- Testare:
  - 13860 exemple de testare =  $2772 \text{ texte traduse} \times 5 \text{ limbi străine}$  (daneză, germană, spaniolă, italiană, olandeză), dar nu sunt ordonate iar limba traducerii nu este etichetată
  - pentru fiecare exemplu de testare trebuie să preziceți eticheta/clasa
  - public test (40%) vs private test (60%)

# Platforma Kaggle

- pentru participare la proiect
  - vă faceți cont pe platforma [www.kaggle.com](http://www.kaggle.com)
  - puteți participa numai pe bază de invitație accesând link-ul de mai jos:  
<https://www.kaggle.com/t/57190c0bbefe46338fc310f2232edf3f>
  - dați Join Competition și apoi puteți participa încărcând predicțiile voastre.
- **2 submisii/zi** (sub forma unui fișier CSV – demo)
- puteți să vă alegeti la final cele mai bune două submisii pe care le considerați voi
- public test (40%) vs. private test (60%)
- termen limită duminică, 13 noiembrie, 23:59
- prezentare proiecte în săptămâna 14-18 noiembrie

# Demo – submisie Kaggle



The screenshot shows a code editor window titled "generate\_random\_submission.py". The file contains the following Python code:

```
1 import numpy as np
2 import random
3
4 testid = np.arange(1,13861)
5 predictedLabels = []
6 labels = ['England', 'Ireland', 'Scotland']
7 for i in range(13860):
8     random_label = random.choice(labels)
9     predictedLabels.append(random_label)
10
11
12 arr = np.array(predictedLabels)
13
14 np.savetxt("/Users/bogdan/Desktop/submisie_Kaggle.csv", np.stack((testid,arr)).T,
15             fmt="%s", delimiter=",", header="id,label", comments = '')
```

The code uses NumPy and the random module to generate a CSV file named "submisie\_Kaggle.csv" for a Kaggle competition. The CSV file contains two columns: "id" and "label". The "id" column ranges from 1 to 13860, and the "label" column is randomly assigned from the list ["England", "Ireland", "Scotland"] for each row.

# Sistem de notare proiect

- proiectul valorează 1.5 puncte din nota finală
- partea de concurs Kaggle = 1 punct
  - locul 1 = 1 punct
  - locul 2 = 0.99 puncte
  - ...
  - Locul 50 = 0.51 puncte
  - locul 51+ = 0.5 puncte (cât timp aveți o performanță > baseline)
- partea de documentație + prezentare = 0.5 puncte
- important: pentru fiecare submisie să știți ce ați făcut (cod Python, parametri, etc). La final vă veți alege 2 submisii care credeți voi că sunt cele mai bune (pot fi același model cu parametri diferiți)

# Documentație proiect - pdf

- descrieți în detaliu (1-2 pagini) **2 modele diferite** folosite (kNN, Naïve Bayes, SVM, Rețea Neuronală):
  - ce caracteristici folosiți;
  - care sunt parametri, hiperparametri modelului;
  - cum antrenați parametri/hiperparametri;
  - cât durează antrenarea;
  - ce performanță ati obținut pe cele 40% de date din setul de date de test public pe Kaggle;
- **pentru un singur model prezențați rezultatele în urma antrenării în maniera 5 fold cross-validation (cursul de azi) + matricea de confuzie asociată**

# Predare proiect

- predarea proiectului înseamnă trimiterea documentației și a codului Python pentru fiecare submisie
- trimiteți la adresa de email: [ub.fmi.cti.ia@gmail.com](mailto:ub.fmi.cti.ia@gmail.com) un email până luni, 14 noiembrie, ora 23:59 cu următoarele fișiere:
  - un fișier pdf cu documentația voastră
  - două fișiere python cu codul pentru submisiile voastre
  - respectați formatul de mai jos



361\_Alexe\_Bogdan\_documentatie.pdf



361\_Alexe\_Bogdan\_submisie1\_cod.py



361\_Alexe\_Bogdan\_submisie2\_cod.py

# Prezentare proiect

- este individuală, are loc în săptămâna 7 (14-18 noiembrie)
- constă într-o discuție cu Alexandra/Sergiu/Bogdan de maxim 10 minute
  - prezentarea voastră 3-5 minute (ce modele ați folosit la cele 2 submisii)
  - 3-5 minute întrebări din partea noastră
- vom face o programare pe care o vom afișa din timp
- încercăm să ne încadrăm în timpul orelor de laborator/proiect

# Regulament de integritate

- regulament privind activitatea studenților la UB:  
[http://fmi.unibuc.ro/ro/pdf/2015/consiliu/UB\\_Regulament\\_studenti\\_2015.pdf](http://fmi.unibuc.ro/ro/pdf/2015/consiliu/UB_Regulament_studenti_2015.pdf)
- regulament de etică și profesionalism la  
FMI:[http://fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament\\_etica\\_FMI.pdf](http://fmi.unibuc.ro/ro/pdf/2015/consiliu/Regulament_etica_FMI.pdf)

- Se consideră **incident minor** cazul în care un student/ o studentă:
- a. preia codul sursă/ rezolvarea unei teme de la un coleg/ o colegă și pretinde că este rezultatul efortului propriu;
  - nu copiați codul de la colegi – veți primi 0 puncte + referat de incident minor

Project – laborator

# Laborator – modelul BOW

## 1. Modelul Bag-of-Words (BOW)

Modelul Bag-of-words este o reprezentare simplificată a textelor folosită în procesarea limbajului natural și în regăsirea informației. Conform acestui model, reprezentăm un text numărând de câte ori apare un cuvânt dintr-un anumit dicționar în textul respectiv. Prin folosirea unei asemenea reprezentări, informația legată de ordinea cuvintelor, gramatică, topică, sensul cuvintelor se pierde.

# Laborator – modelul BOW

Exemplu<sup>1</sup>: considerăm textele 1 și 2 de mai jos în limba engleză.

- (1) John likes to watch movies. Mary likes movies too.
- (2) John also likes to watch football games.

Eliminând semnele de punctuație obținem listele de cuvinte pentru cele două texte:

```
"John", "likes", "to", "watch", "movies", "Mary", "likes", "movies", "too"  
"John", "also", "likes", "to", "watch", "football", "games"
```

Considerăm dicționarul D format din reuniunea tuturor cuvintelor din cele 2 texte:

```
D = {"John", "likes", "to", "watch", "movies", "Mary", "too", "also",  
"football", "games"}
```

# Laborator – modelul BOW

Considerăm dicționarul D format din reuniunea tuturor cuvintelor din cele 2 texte:

```
D = {"John", "likes", "to", "watch", "movies", "Mary", "too", "also",
      "football", "games"}
```

Reprezentăm fiecare text numărând de câte ori apare fiecare cuvânt din dicționarul D în fiecare text. Obținem reprezentările bag-of-words următoare:

```
BoW1 = {"John":1, "likes":2, "to":1, "watch":1, "movies":2, "Mary":1,
         "too":1, "also":0, "football":0, "games":0};
```

```
BoW2 = {"John":1, "likes":1, "to":1, "watch":1, "movies":0, "Mary":0,
         "too":0, "also":1, "football":1, "games":1};
```

# Laborator – modelul BOW

- (1) John likes to watch movies. Mary likes movies too.
- (2) John also likes to watch football games.

Considerăm dicționarul D format din reuniunea tuturor cuvintelor din cele 2 texte:

```
D = {"John", "likes", "to", "watch", "movies", "Mary", "too", "also",
      "football", "games"}
```

Reprezentăm fiecare text numărând de câte ori apare fiecare cuvânt din dicționarul D în fiecare text. Obținem reprezentările bag-of-words următoare:

```
BoW1 = {"John":1, "likes":2, "to":1, "watch":1, "movies":2, "Mary":1,
         "too":1, "also":0, "football":0, "games":0};
```

```
BoW2 = {"John":1, "likes":1, "to":1, "watch":1, "movies":0, "Mary":0,
         "too":0, "also":1, "football":1, "games":1};
```

Pentru dicționarul D fixat, reprezentările se pot scrie sub forma de vectori de frecvențe:

```
v1 = [1, 2, 1, 1, 2, 1, 1, 0, 0, 0];
v2 = [1, 1, 1, 1, 0, 0, 0, 1, 1, 1];
```

# Laborator – modelul BOW

Pentru dicționarul D fixat, reprezentările se pot scrie sub forma de vectori de frecvențe:

```
v1 = [1, 2, 1, 1, 2, 1, 1, 0, 0, 0];  
v2 = {1, 1, 1, 1, 0, 0, 0, 1, 1, 1};
```

Suma elementelor fiecarui vector reprezintă numărul de cuvinte din text. De obicei, pentru probleme de clasificare în care încercăm să discriminăm între texte de diferite lungimi se folosesc reprezentări normalize. Spre exemplu, folosind norma  $L_1$  (suma absolută a elementelor unui vector) obținem vectorii normalizați  $L_1$ :

$$v1_{L1} = \frac{v1}{\|v1\|_1} = \left[ \frac{1}{9}, \frac{2}{9}, \frac{1}{9}, \frac{1}{9}, \frac{2}{9}, \frac{1}{9}, \frac{1}{9}, 0, 0, 0 \right]$$

$$v2_{L1} = \frac{v2}{\|v2\|_1} = \left[ \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, 0, 0, 0, \frac{1}{7}, \frac{1}{7}, \frac{1}{7} \right]$$

Folosind norma  $L_2$  (norma Euclidiană) obținem vectorii normalizați  $L_2$ :

$$v1_{L2} = \frac{v1}{\|v1\|_2} = \left[ \frac{1}{\sqrt{13}}, \frac{2}{\sqrt{13}}, \frac{1}{\sqrt{13}}, \frac{1}{\sqrt{13}}, \frac{2}{\sqrt{13}}, \frac{1}{\sqrt{13}}, \frac{1}{\sqrt{13}}, 0, 0, 0 \right]$$

$$v2_{L2} = \frac{v2}{\|v2\|_2} = \left[ \frac{1}{\sqrt{7}}, \frac{1}{\sqrt{7}}, \frac{1}{\sqrt{7}}, \frac{1}{\sqrt{7}}, 0, 0, 0, \frac{1}{\sqrt{7}}, \frac{1}{\sqrt{7}}, \frac{1}{\sqrt{7}} \right]$$

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 5

# Bonusul de la laborator

- puteți acumula maxim 1 punct bonus de la laborator;
- la examenul final, se adaugă acest bonus la notă;
- **pentru prima parte a semestrului aveți alocat un bonus de 0,5 puncte**
  - Alexandra și Sergiu stabilesc care sunt aceste condiții (posibil să puncteze prezența la 2 ore de laborator cu 0,05 puncte);
- pentru a doua parte a semestrului aveți alocat un bonus de 0,5 puncte
  - Irina stabilește care sunt aceste condiții;

# Eroare în multimea de testare

test\_data

1

text

Hr. formand, selv om vi i høj grad sympatiserer med de borgere, der befinner sig i denne uheldige situation, mener vi i UKIP, at problemet med frataelse af ejendom i Valencia burde have været behandlet ved et planlægning er et område, der skal forblive på lokalt plan, for at lokalområdernes behov skal møde forståelse. En centraliseret politik gør ikke noget bedre, den forværer tværtimod problemet. Det har vi set. For det første er der den fælles fiskeripolitik med det særliges skadelige kvotesystem. Hyldet som et miljøprojekt har den gjort næsten uoprettelig skade på fiskebestandene. En meget stor del af fiskeforarbejdningen er i Spanien, og det er ikke en god idé at få en centraliseret politik. De problemer, som de ikke-spanske statsborgere, herunder mange britiske statsborgere, i Valencia står over for, bør løses mellem regeringerne. Jeg er ked af, at det ikke er lykkedes den britiske regering at opnå nogen resultater.

2

Byplanlægning er et område, der skal forblive på lokalt plan, for at lokalområdernes behov skal møde forståelse. En centraliseret politik gør ikke noget bedre, den forværer tværtimod problemet. Det har vi set. For det første er der den fælles fiskeripolitik med det særliges skadelige kvotesystem. Hyldet som et miljøprojekt har den gjort næsten uoprettelig skade på fiskebestandene. En meget stor del af fiskeforarbejdningen er i Spanien, og det er ikke en god idé at få en centraliseret politik. De problemer, som de ikke-spanske statsborgere, herunder mange britiske statsborgere, i Valencia står over for, bør løses mellem regeringerne. Jeg er ked af, at det ikke er lykkedes den britiske regering at opnå nogen resultater.

3

Quiero dejar constancia de mi apoyo a este informe, como un gran paso para facilitar la aplicación del principio de «quien contamina paga». Cuando se producen daños medioambientales, la primera pregunta es: ¿Cuál es el responsable? El requisito de que la Comisión presente un informe, en el plazo de seis años, también nos dará la oportunidad de revisar esta importante cuestión y de evaluar convenientemente su impacto en la práctica. En un mundo ideal, yo esperaría que los ciudadanos estuviesen protegidos frente a cualquier riesgo de daño medioambiental, y afortunadamente estamos haciendo progresos con algunas medidas preventivas.

7583

Mijnheer de fungerend voorzitter van de Raad, morgen zal dit Parlement stemmen over een amendement op uw nationaal waterplan, dus wij hopen dat u vandaag van uw glas water geniet. Als de Europese Unie het proces van Lissabon een schoolrapport zou geven, dan zou daarvan vermoedelijk staan: "moet beter zijn best doen". Wij hebben het afgelopen jaar enige vooruitgang gezien met betrekking tot de waterkwestie. Ik vrees echter dat uit dit schoolrapport eveneens zou blijken dat de Unie op teveel gebieden een onvoldoende haalt. In de Commissiemeeting aan de Europese Raad van Barcelona wordt verklaard dat men de richtlijn inzake het overnamebod moet accepteren. Het feit dat de richtlijn inzake het overnamebod verworpen is betekent een grote tegenslag voor het concurrentievermogen binnen de Unie en ik roep de andere politieke fracties en de Raad op samen met ons de richtlijn inzake het overnamebod te accepteren. Een ander punt waarover de schoolmeester de Raad op de vingers zou tikken is het Europees octrooi. Als wij de kloof tussen Europa en de Verenigde Staten op het gebied van vernieuwing willen dichten, dan moet dat gaan door de richtlijn inzake het overnamebod te accepteren. Tot slot zou ons rapport laten zien dat deze leerling te weinig ambitie heeft. Als wij willen dat de Europese Unie uitblinkt in de klas, moeten wij voortmaken met de liberalisering van de energie- en gasmarkten, en dat moet gaan door de richtlijn inzake het overnamebod te accepteren. Wanneer de Top van Barcelona wordt geëvalueerd, zullen wij u beoordelen naar uw schoolrapport. Zoals de zaken er nu voorstaan, moet de leerling beter zijn best doen.

7584

Señor Presidente, en primer lugar, acaso debemos precipitarnos a estrechar los lazos con Egipto, con la esperanza de que las cuestiones sin resolver y un fallo injusto se disiparán y se resolverán por sí solas. ¿Aprueba el Parlamento las violaciones del derecho a un juicio justo, la libertad de expresión y la libertad de asociación, ya sea ésta social, política, religiosa o sexual? Porque eso es lo que estaremos diciendo. Hr. formand, jeg er blevet anmodet om at stille et mundtligt ændringsforslag. Som en mellemvej og for at få dette ændringsforslag vedtaget foreslår jeg, at vi i slutningen af punkt 37 tilføjer "og bidrage til bekæmpelse af". (Formanden konstaterede, at ingen gjorde indsigelse, hvorfor det mundtlige ændringsforslag kunne tages i betragtning) (Forslaget til beslutning vedtages) Stemmeforklaringer Budget 2002

7585

7586

# Eroare în multimea de testare

Mijnheer de fungerend voorzitter van de Raad, morgen zal dit Parlement stemmen over een amendement op uw nationaal waterplan, dus wij hopen dat u vandaag van uw glas water geniet.

Als de Europese Unie het proces van Lissabon een schoolrapport zou geven, dan zou daarin vermoedelijk staan: "moet beter zijn best doen". Wij hebben het afgelopen jaar enige vooruitgang gezien met betrekking tot de richtlijn over de voorwaarden voor de levering van elektriciteit.

Ik vrees echter dat uit dit schoolrapport eveneens zou blijken dat de Unie op teveel gebieden een onvoldoende haalt. In de Commissiemeedenkdocumentatie aan de Europese Raad van Barcelona wordt verklaard dat met ons samen moet worden gewerkt om de economische groei te waarborgen.

Het feit dat de richtlijn inzake het overnamebod verworpen is betekent een grote tegenslag voor het concurrentievermogen binnen de Unie en ik roep de andere politieke fracties en de Raad op samen met ons te werken om de economische groei te waarborgen.

Een ander punt waarover de schoolmeester de Raad op de vingers zou tikken is het Europees octrooi. Als wij de kloof tussen Europa en de Verenigde Staten op het gebied van vernieuwing willen dichten, dan moet ons rapport leren zien dat deze leerling te weinig ambitie heeft.

Tot slot zou ons rapport laten zien dat deze leerling te weinig ambitie heeft. Als wij willen dat de Europese Unie uitblinkt in de klas, moeten wij voortmaken met de liberalisering van de energie- en gasmarkten, de arbeidsmarkt en de landbouwmarkt.

Wanneer de Top van Barcelona wordt geëvalueerd, zullen wij u beoordelen naar uw schoolrapport. Zoals de zaken er nu voorstaan, moet de leerling beter zijn best doen.

7583

7584

7585

7586

Señor Presidente, en primer lugar, acaso debemos precipitarnos a estrechar los lazos con Egipto, con la esperanza de que las cuestiones sin resolver y un fallo injusto se disiparán y se resolverán por sí solos.

¿Aprueba el Parlamento las violaciones del derecho a un juicio justo, la libertad de expresión y la libertad de asociación, ya sea ésta social, política, religiosa o sexual? Porque eso es lo que estaremos diciendo.

Hr. formand, jeg er blevet anmodet om at stille et mundtligt ændringsforslag. Som en mellemvej og for at få dette ændringsforslag vedtaget foreslår jeg, at vi i slutningen af punkt 37 tilføjer "og bidrage til bekæmpelse af".

(Formanden konstaterede, at ingen gjorde indsigelse, hvorfor det mundtlige ændringsforslag kunne tages i betragtning)

(Forslaget til beslutning vedtages) Stemmeklaringer

Budget 2002

## Fișierul soluție:

7584

**7583** Ireland

# Recapitulare – cursul trecut

1. Clasificatorul naïve Bayes
2. Evaluarea performanței unui model
3. Strategii de împărțire a datelor
4. Proiect
  - concurs pe platforma Kaggle
  - demo
  - laborator săptămâna 5

# Cuprinsul cursului de azi

1. Mașini cu vectori support (SVMs – support vector machines)
2. Perceptronul

# Mașini cu vectori suport (SVMs – Support Vector Machines)

# Separabilitate liniară

Două multimi de puncte într-un spațiu de dimensiune  **$n$**  sunt liniar separabile dacă există un hiperplan (subspațiu de dimensiune  **$n - 1$** ) care le separă perfect.

**Exemplu:**  $n = 1$

**multimi liniar separabile**



**multimi liniar neseparabile**



# Separabilitate liniară

Două multimi de puncte într-un spațiu de dimensiune  **$n$**  sunt liniar separabile dacă există un hiperplan (subspațiu de dimensiune  **$n - 1$** ) care le separă perfect.

**Exemplu:**  $n = 1$

multimi liniar separabile



multimi liniar neseparabile



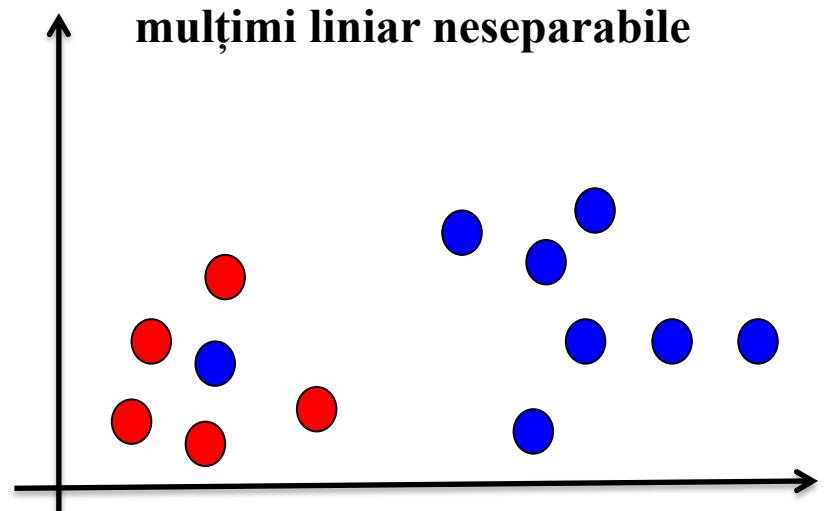
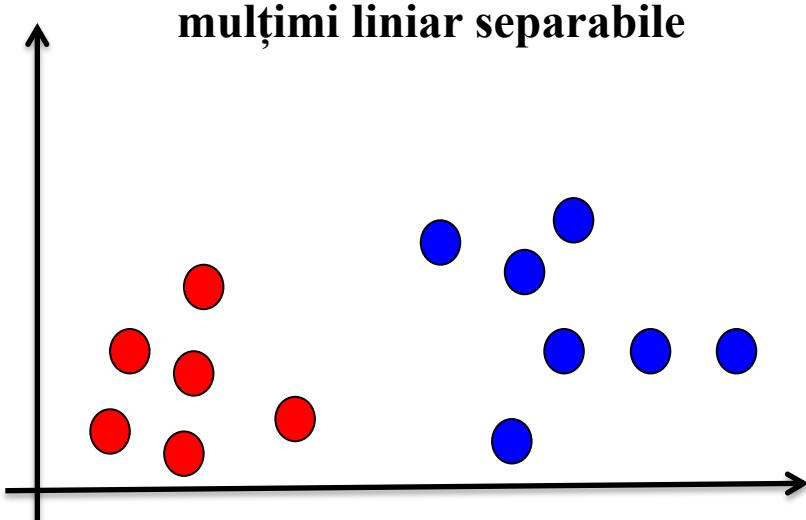
Hiperplan de dimensiune 0 = un punct.  
Există o infinitate de hiperplane (puncte) care separă mulțimile perfect.

Nu există nici un hiperplan (punct) care separă mulțimile perfect.

# Separabilitate liniară

Două multimi de puncte într-un spațiu de dimensiune  $n$  sunt liniar separabile dacă există un hiperplan (subspațiu de dimensiune  $n - 1$ ) care le separă perfect.

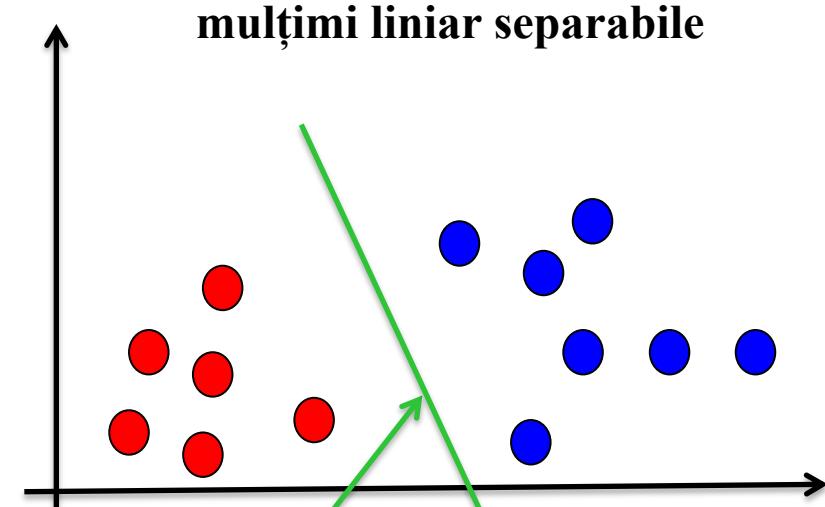
**Exemplu:**  $n = 2$



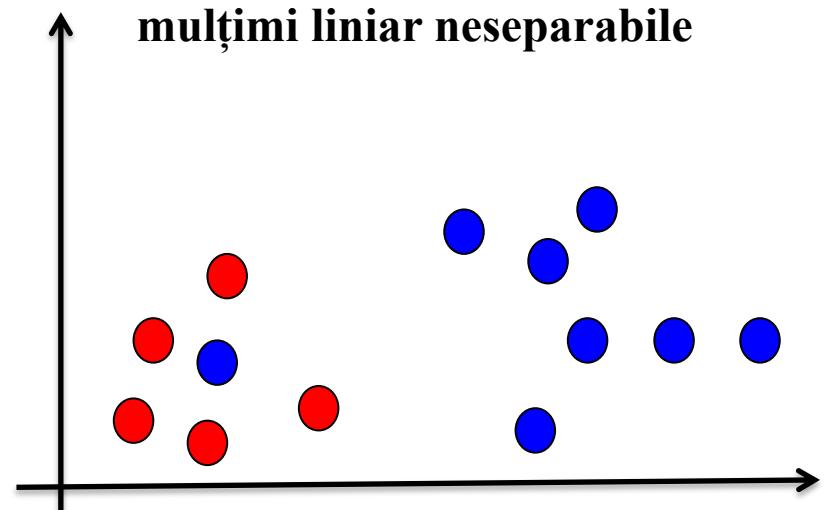
# Separabilitate liniară

Două multimi de puncte într-un spațiu de dimensiune  $n$  sunt liniar separabile dacă există un hiperplan (subspațiu de dimensiune  $n - 1$ ) care le separă perfect.

**Exemplu:**  $n = 2$



Hiperplan de dimensiune 1 = o dreaptă  
Există o infinitate de hiperplane (drepte) care separă mulțimile perfect.



Nu există niciun hiperplan (dreaptă) care separă mulțimile perfect.

# Separabilitate liniară

Două multimi de puncte într-un spațiu de dimensiune  **$n$**  sunt liniar separabile dacă există un hiperplan (subspațiu de dimensiune  **$n - 1$** ) care le separă perfect.

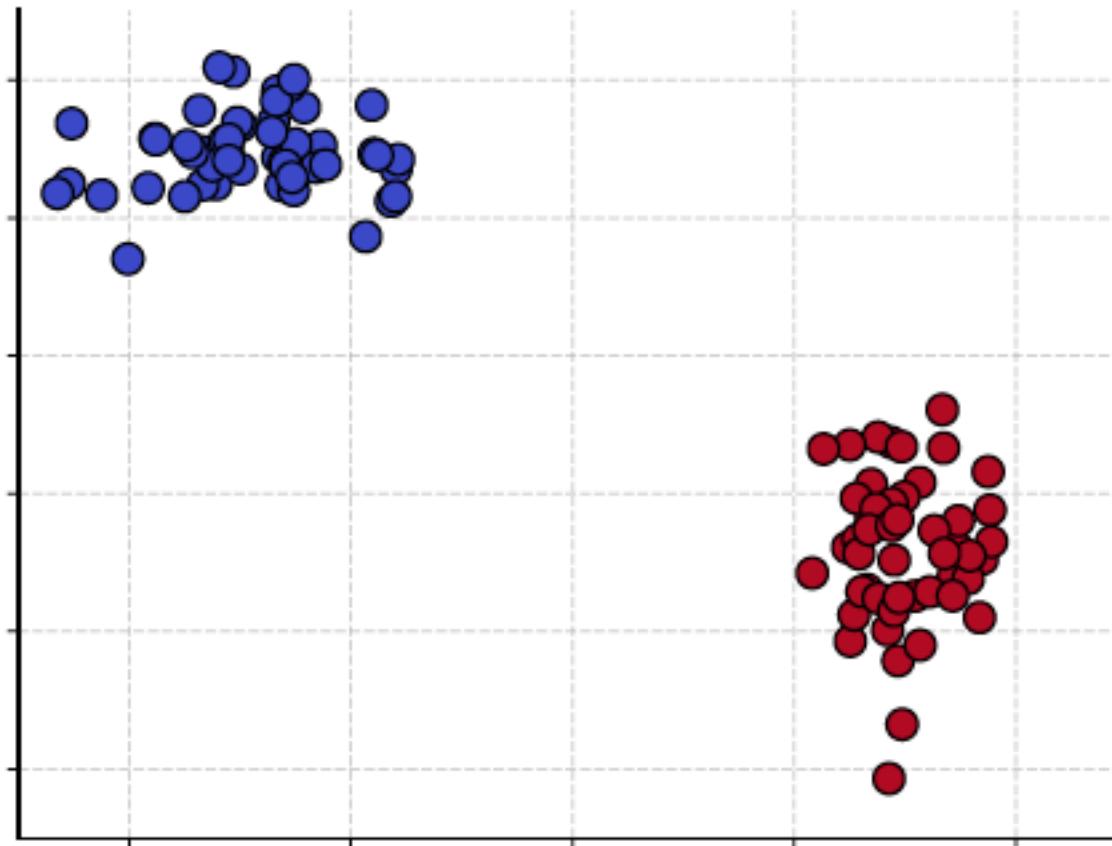
Un hiperplan în 1D este un **punct**.

Un hiperplan în 2D este o **dreaptă**.

Un hiperplan în 3D este un **plan**.

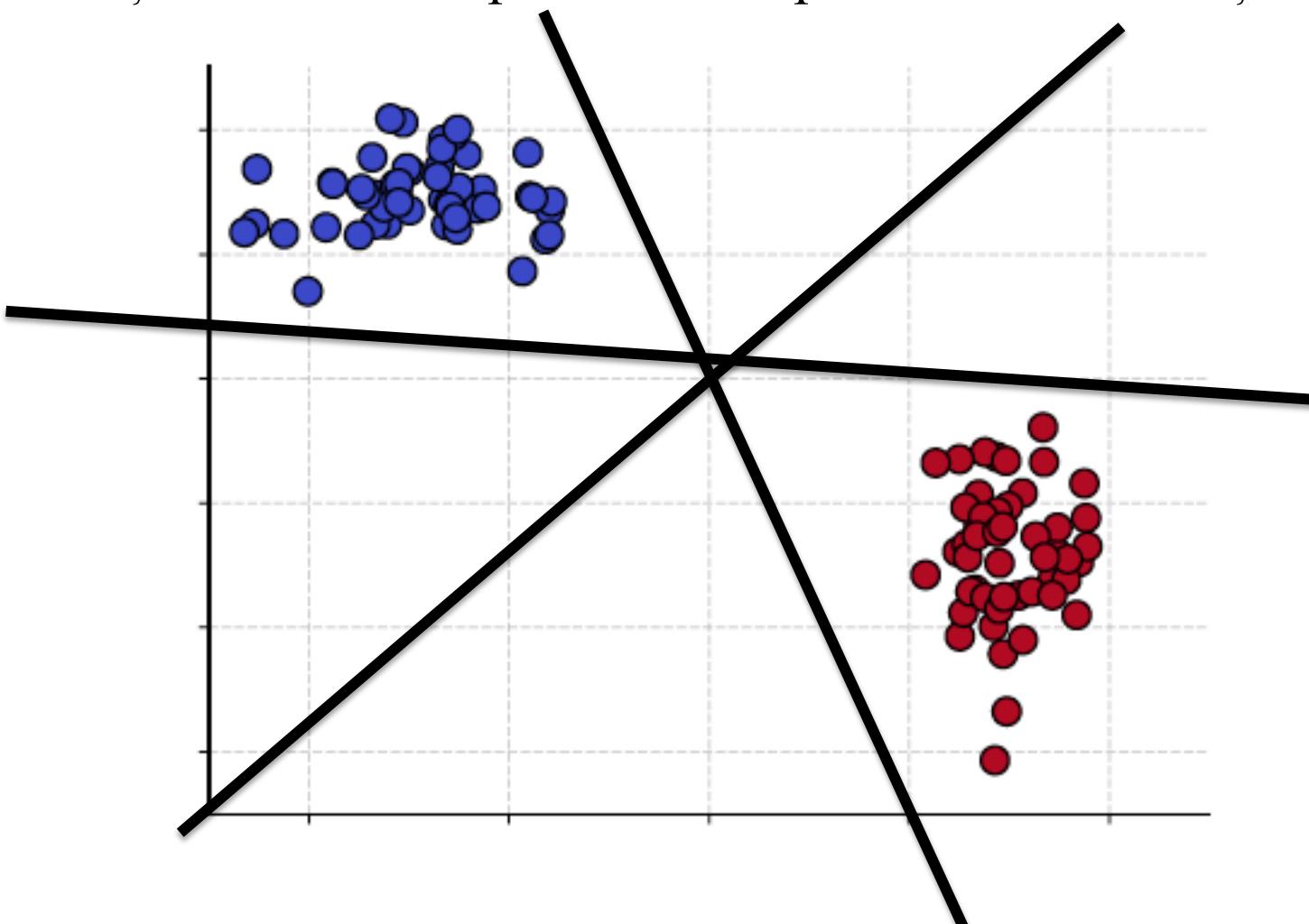
# Frontiera de separare (decision boundary)

- cum ați desena o dreaptă care să separe cele două multimi de puncte?



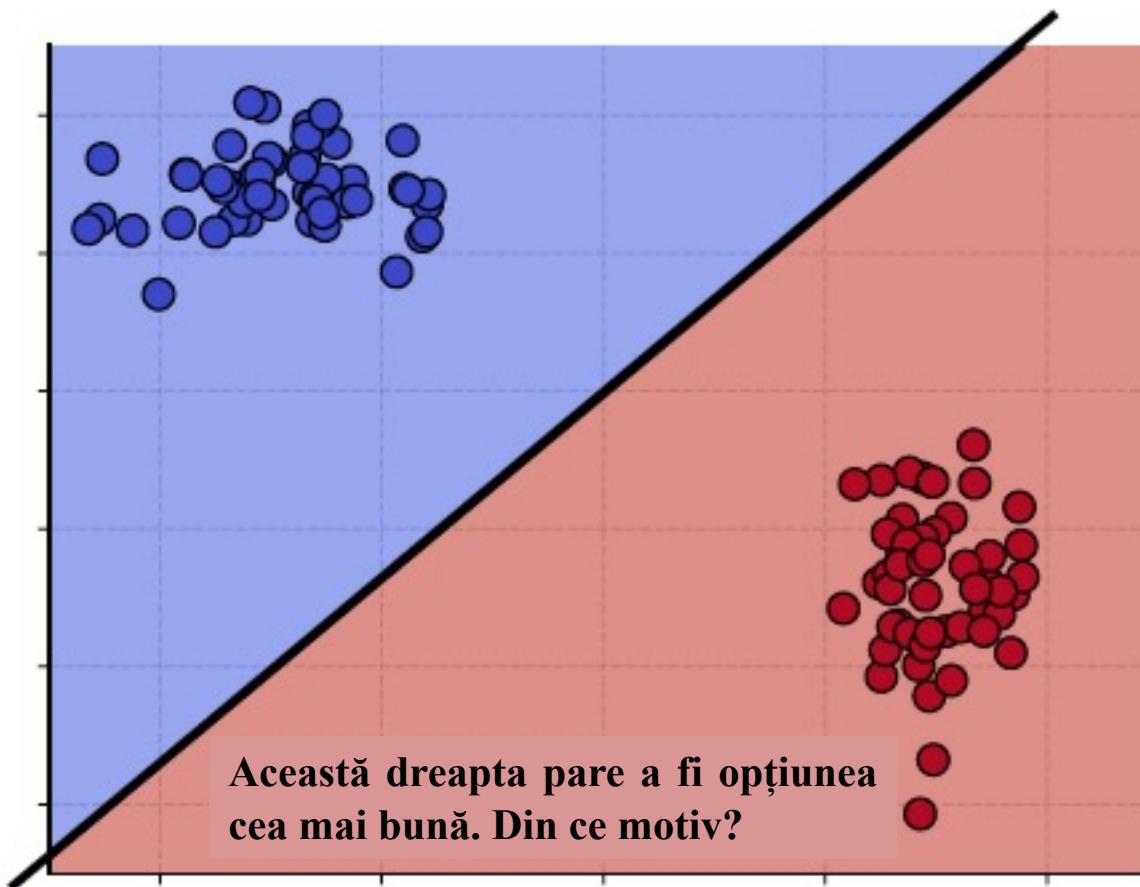
# Frontiera de separare (decision boundary)

- cum ați desena o dreaptă care să separe cele două multimi de puncte?



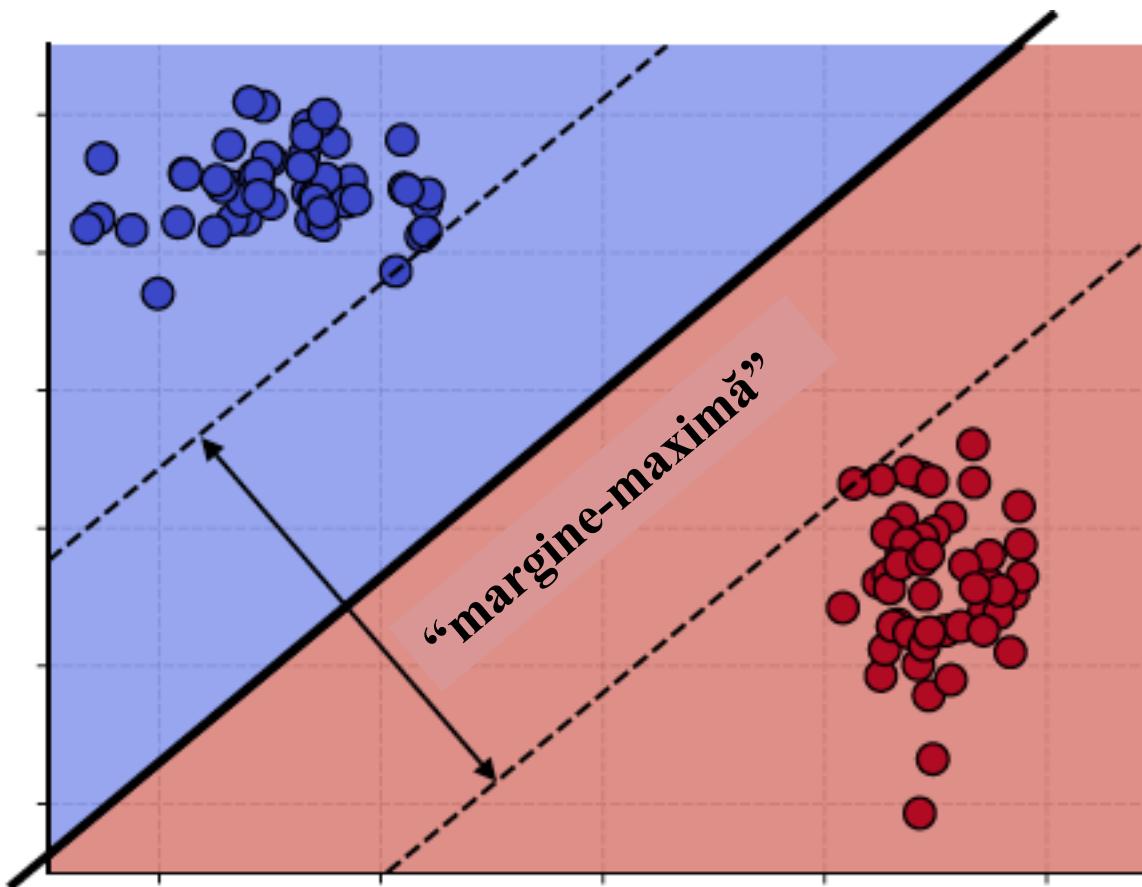
# Frontiera de separare (decision boundary)

- cum ați desena o dreaptă care să separe cele două multimi de puncte?



# Frontiera de separare (decision boundary)

- cum ați desena o dreaptă care să separe cele două multimi de puncte?



# Definiția unui SVM

- Un **SVM** (support vector machine = mașină cu vectori suport) este un clasificator liniar binar nonprobabilist.
  - **clasificator**: metodă de învățare supervizată care are scop predictia de clase
  - **liniar**: frontieră de decizie este un hiperplan în n-dimensiuni
  - **binar**: învață să discrimineze între 2 clase (clasa + și clasa -)
  - **nonprobabilist**: rezultatul unui SVM nu este limitat, nu poate fi interpretat ca o probabilitate

Există extensii care fac ca SVM să poate fi folosit în probleme de regresie, să aibă o frontieră neliniară, cu mai multe clase și să fie probabilist

- Un SVM încearcă să găsească hiperplanul de separare care este cel mai departe posibil (are margine maximă) de punctele de antrenare
  - un punct este clasificat + sau -, în funcție de ce parte a hiperplanului se află

# Învățarea unui hiperplan de separare “bun”

x – caracteristici (features)

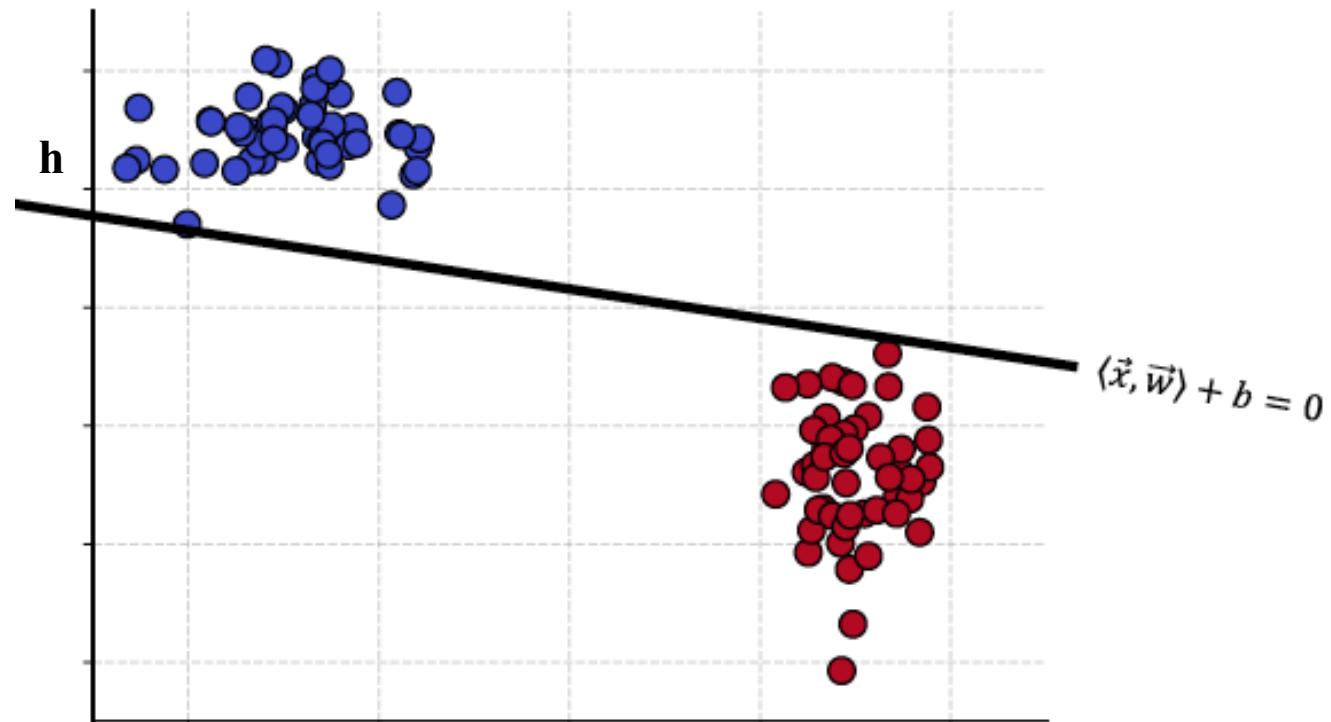
y – etichete (labels)

$$\langle \vec{x}_+, \vec{w} \rangle + b \geq 0 \quad \forall \vec{x}_+ \in \{\vec{x}^{(i)} | y^{(i)} = +1\}$$

$$\langle \vec{x}_-, \vec{w} \rangle + b < 0 \quad \forall \vec{x}_- \in \{\vec{x}^{(i)} | y^{(i)} = -1\}$$

Hiperplanul  $h$  satisfac condițiile impuse dar nu are nicio margine.

Un hiperplan  $h$  este cu atât mai bun cu cât are o margine mai mare.



# Învățarea unui hiperplan de separare “bun”

x – caracteristici (features)

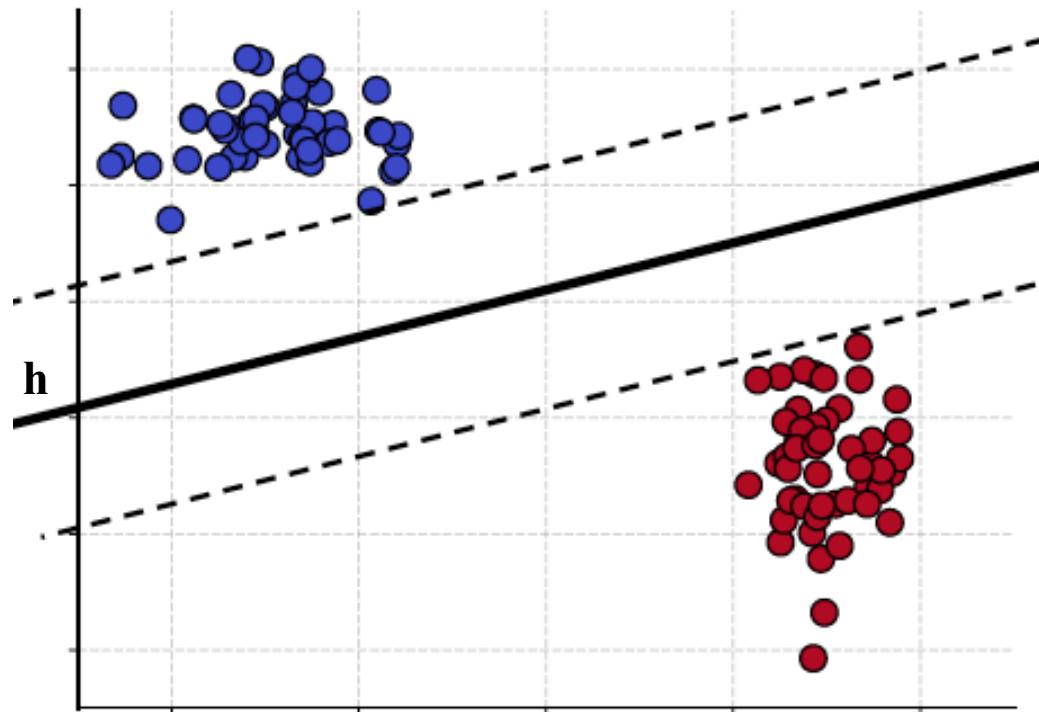
y – etichete (labels)

$$\langle \vec{x}_+, \vec{w} \rangle + b \geq 0 \quad \forall \vec{x}_+ \in \{\vec{x}^{(i)} | y^{(i)} = +1\}$$

$$\langle \vec{x}_-, \vec{w} \rangle + b < 0 \quad \forall \vec{x}_- \in \{\vec{x}^{(i)} | y^{(i)} = -1\}$$

Hiperplanul  $h$  satisfac condițiile impuse dar nu are nicio margine.

Un hiperplan  $h$  este cu atât mai bun cu cât are o margine mai mare.



# Regula de decizie + problema de optimizare

- Regula de decizie este:

$$\vec{x} \text{ este exemplu + dacă } \langle \vec{x}, \vec{w} \rangle + b \geq 0$$

$$\vec{x} \text{ este exemplu - dacă } \langle \vec{x}, \vec{w} \rangle + b < 0$$

- Pentru a obține hiperplanul de margine maximă trebuie să rezolvăm problema de optimizare (nu intrăm în detaliu):

$$\text{minimize} \quad \frac{\|\vec{w}\|^2}{2}$$

$$\text{cu constrângerile} \quad y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0$$

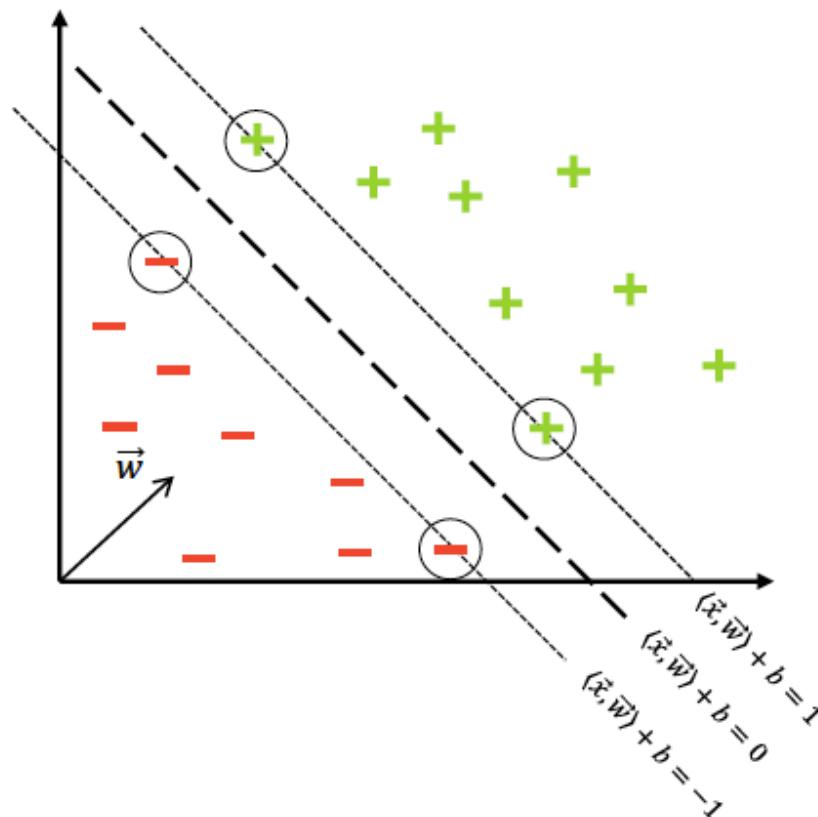
Forma primală pentru SVM

# Vectori suport

- Exemple de antrenare  $(\vec{x}^{(i)}, y^{(i)})$  de la margine satisfac:

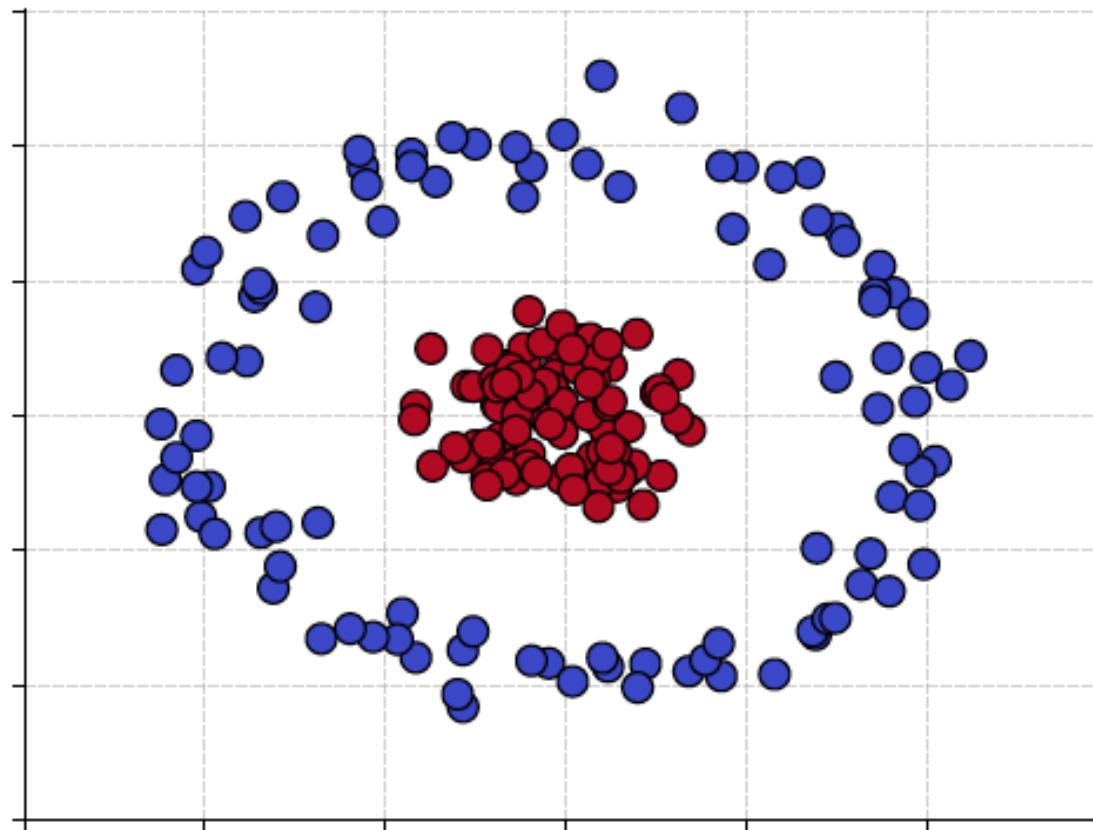
$$y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 = 0$$

- Aceste exemple de antrenare se numesc **vectori suport (Support Vectors)**



# Liniar separabilitate?

- În practică, exemplele de antrenare pozitive și negative nu sunt liniar separabile

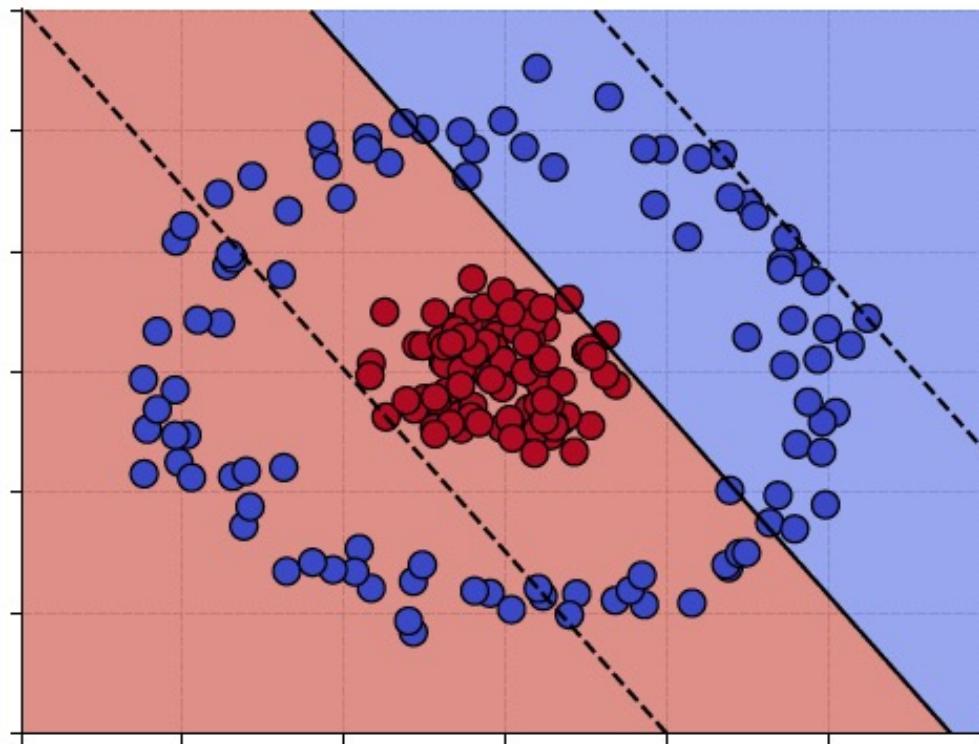


# Liniar separabilitate?

- În practică, exemplele de antrenare pozitive și negative nu sunt liniar separabile
- Toate calculele precedente s-au bazat pe liniar separabilitatea exemplelor din mulțimea de antrenare
  - există un hiperplan  $h$  care separă + de – (“**hard-margin**” linear SVM)
- Cum putem trata cazul de date care nu sunt liniar separabile folosind SVM?
- Două soluții:
  1. **“Kernel trick”** – maparea datelor într-un spațiu cu dimensionalitate mult mai mare în care acestea sunt separabile
  2. **“Soft-margin” SVM** – permitem SVM-ului să facă câteva greșeli la antrenare (folosim o funcție cost) în schimbul obținerii unei margini bune

# Date care nu sunt liniar separabile

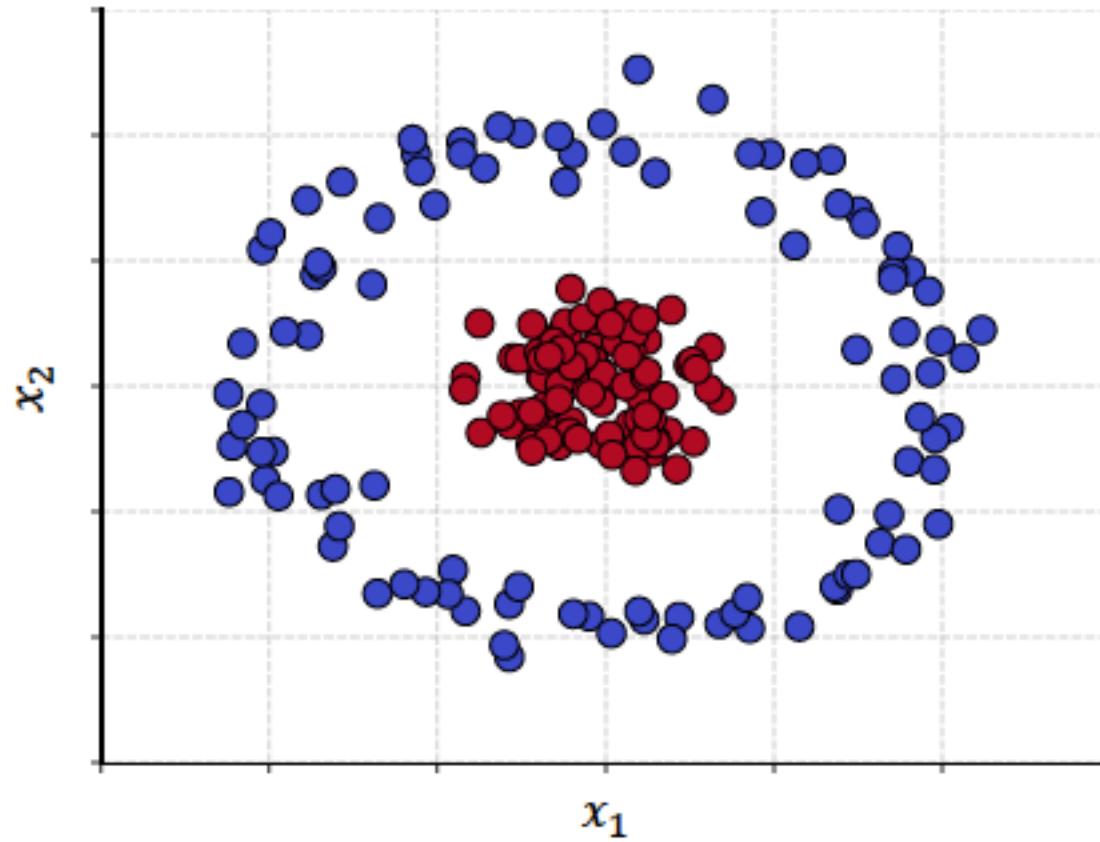
- În practică, exemplele de antrenare pozitive și negative nu sunt liniar separabile
- Nu putem găsi un hiperplan soluție folosind “hard-margin” linear SVM



# The kernel trick

$$\vec{x} \in \mathbb{R}, \vec{x} = (x_1, x_2)$$

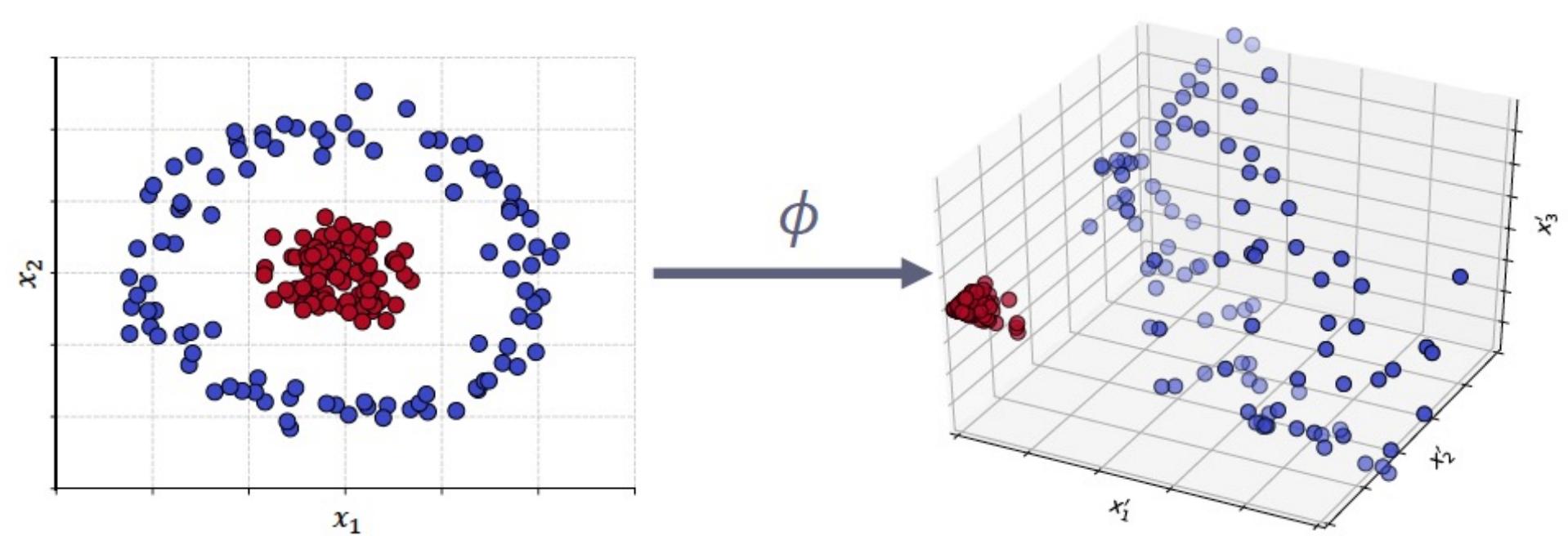
$$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3, \phi((x_1, x_2)) = (x'_1, x'_2, x'_3) = (x_1^2, x_2^2, x_1 x_2 \sqrt{2})$$



# The kernel trick

$$\vec{x} \in \mathbb{R}, \vec{x} = (x_1, x_2)$$

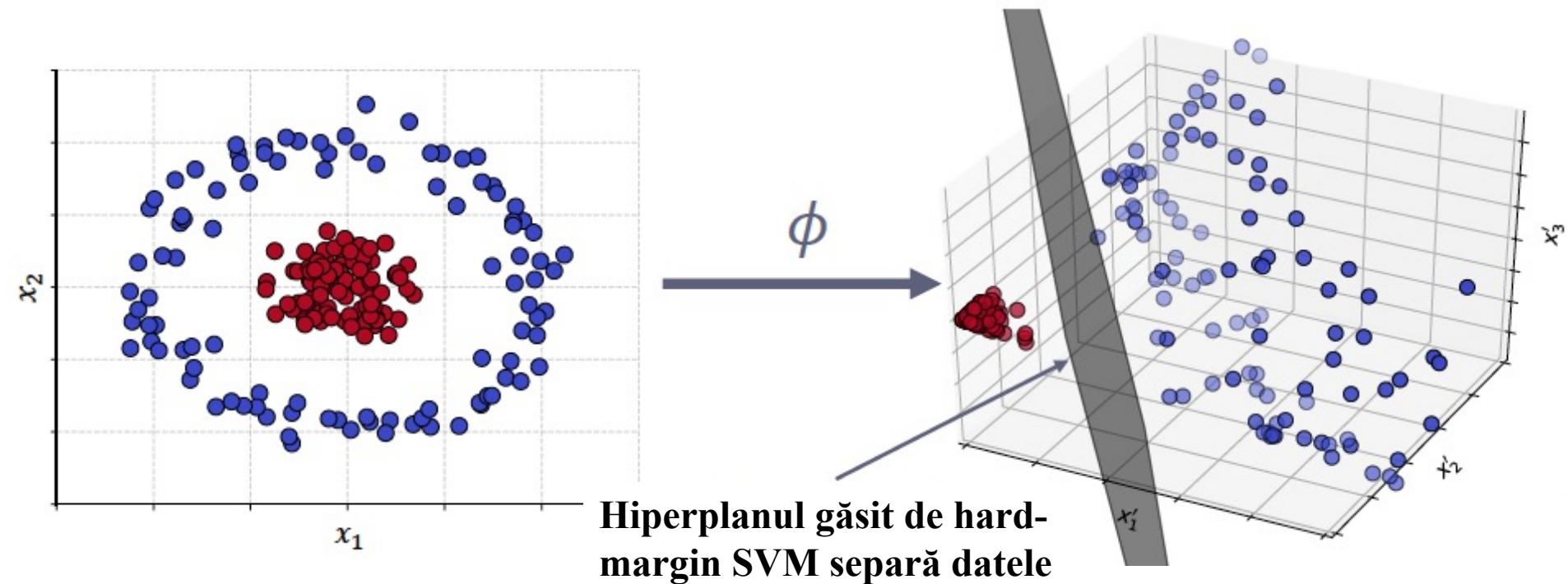
$$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3, \phi((x_1, x_2)) = (x'_1, x'_2, x'_3) = (x_1^2, x_2^2, x_1 x_2 \sqrt{2})$$



# The kernel trick

$$\vec{x} \in \mathbb{R}, \vec{x} = (x_1, x_2)$$

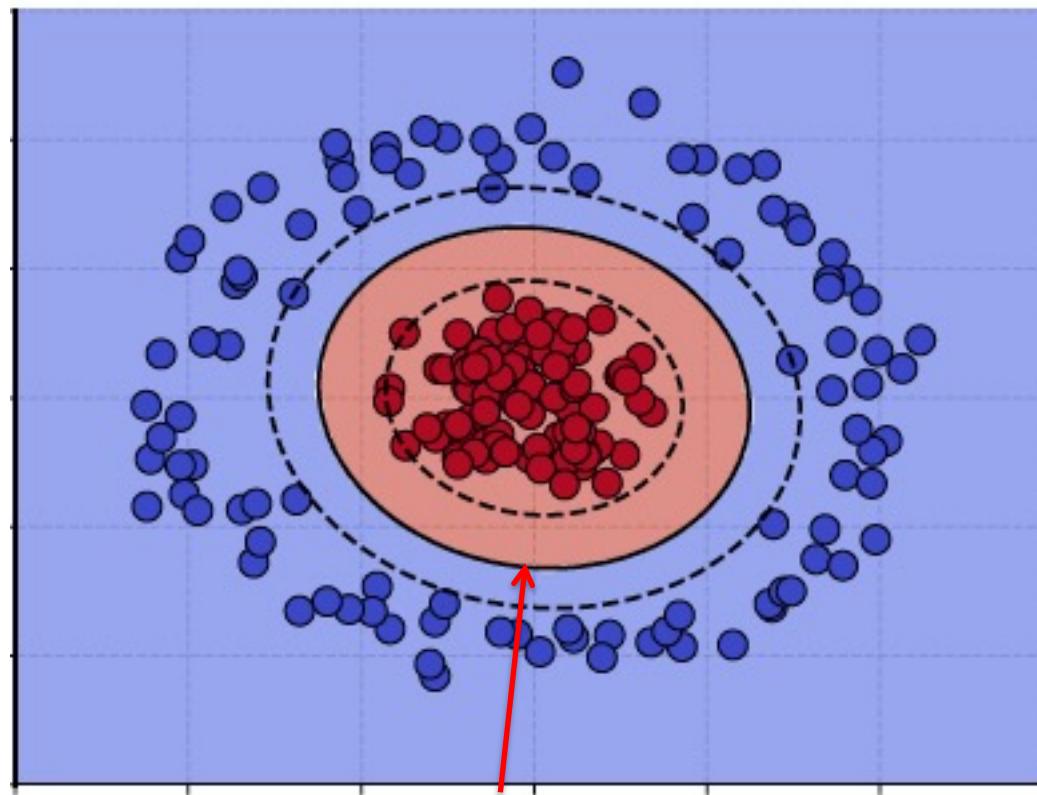
$$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3, \phi((x_1, x_2)) = (x'_1, x'_2, x'_3) = (x_1^2, x_2^2, x_1 x_2 \sqrt{2})$$



# The kernel trick

$$\vec{x} \in \mathbb{R}, \vec{x} = (x_1, x_2)$$

$$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3, \phi((x_1, x_2)) = (x'_1, x'_2, x'_3) = (x_1^2, x_2^2, x_1 x_2 \sqrt{2})$$



Hiperplanul găsit de hard-margin SVM se translatează  
în spațiul original într-o frontieră de decizie neliniară

# The kernel trick

- Constă în folosirea unei funcții kernel (nucleu)  $\phi$  care mapează datele într-un alt spațiu (de obicei de dimensionalitate mult mai mare) în care datele sunt liniar separabile
- Orice algoritm care folosește o asemenea abordare se numește metodă kernel
- Funcția kernel poate fi văzută ca o funcție de similaritate între date.

# Cele mai cunoscute funcții kernel

Linear kernel:

$$K(\vec{x}, \vec{y}) = \langle \vec{x}, \vec{y} \rangle$$

Polynomial kernel:

$$K(\vec{x}, \vec{y}) = (\langle \vec{x}, \vec{y} \rangle + c)^d$$

Radial basis function (RBF) kernel:

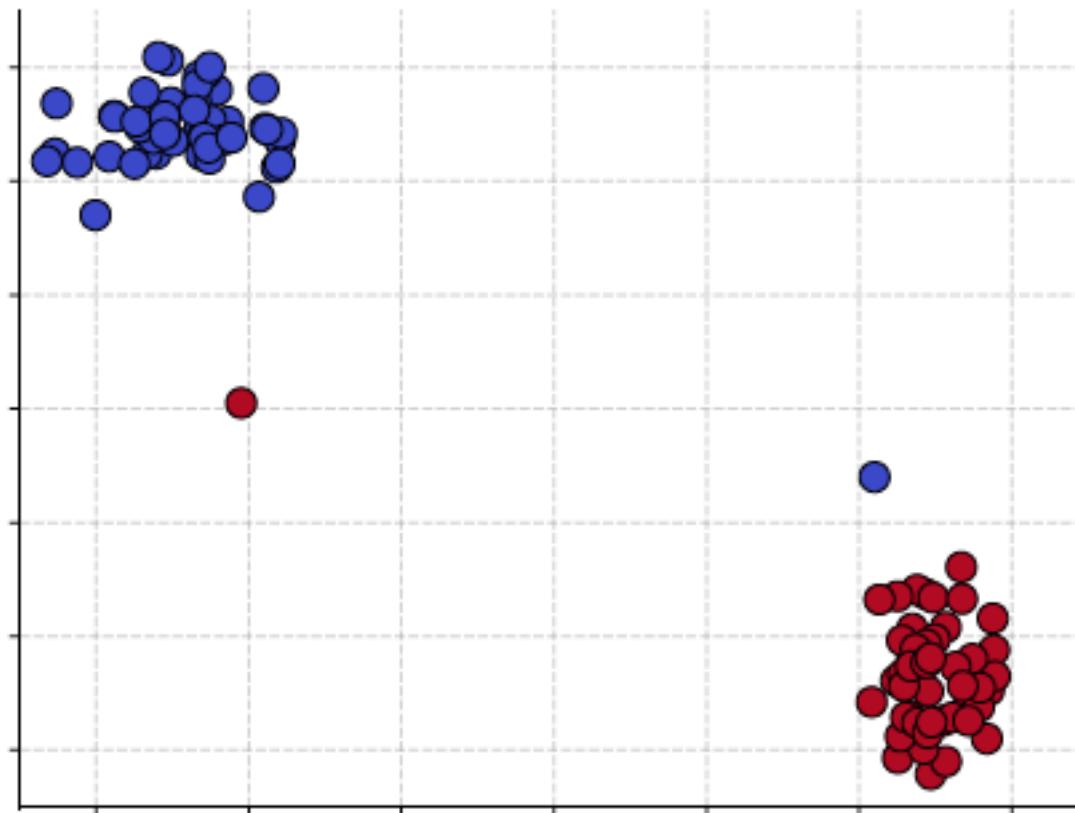
$$K(\vec{x}, \vec{y}) = e^{-\frac{\|\vec{x}-\vec{y}\|^2}{2\sigma^2}}$$

Sigmoid kernel:

$$K(\vec{x}, \vec{y}) = \tanh(\gamma \langle \vec{x}, \vec{y} \rangle + c)$$

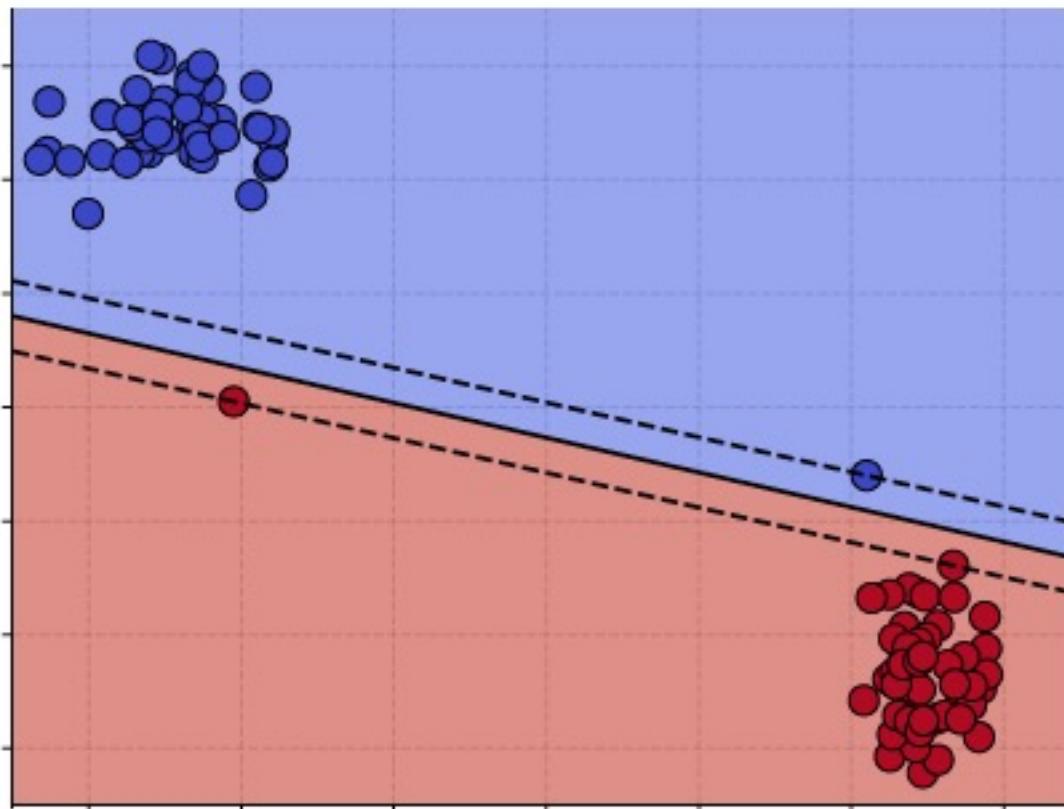
# Soft-margin SVM

- Ce se întâmplă dacă antrenăm un hard-margin SVM pe setul de date de mai jos?



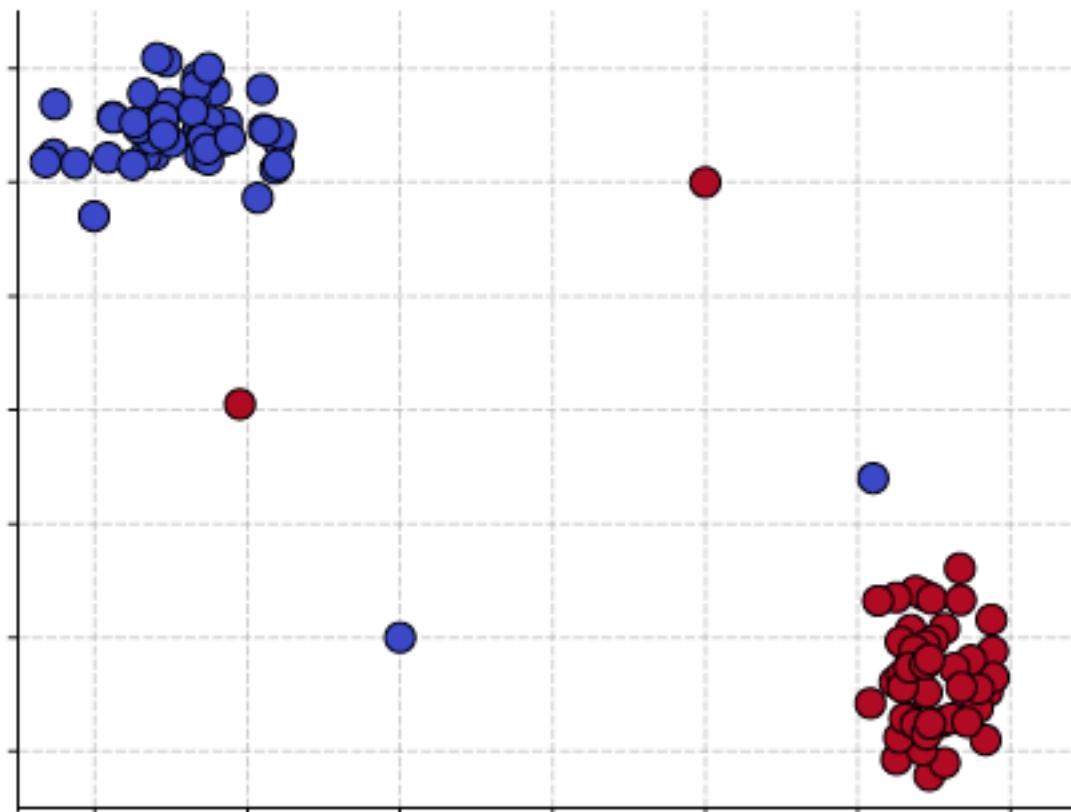
# Soft-margin SVM

- Ce se întâmplă dacă antrenăm un hard-margin SVM pe setul de date de mai jos?
- SVM-ul găsește un hiperplan care separă datele perfect însă acesta are o margine foarte mică



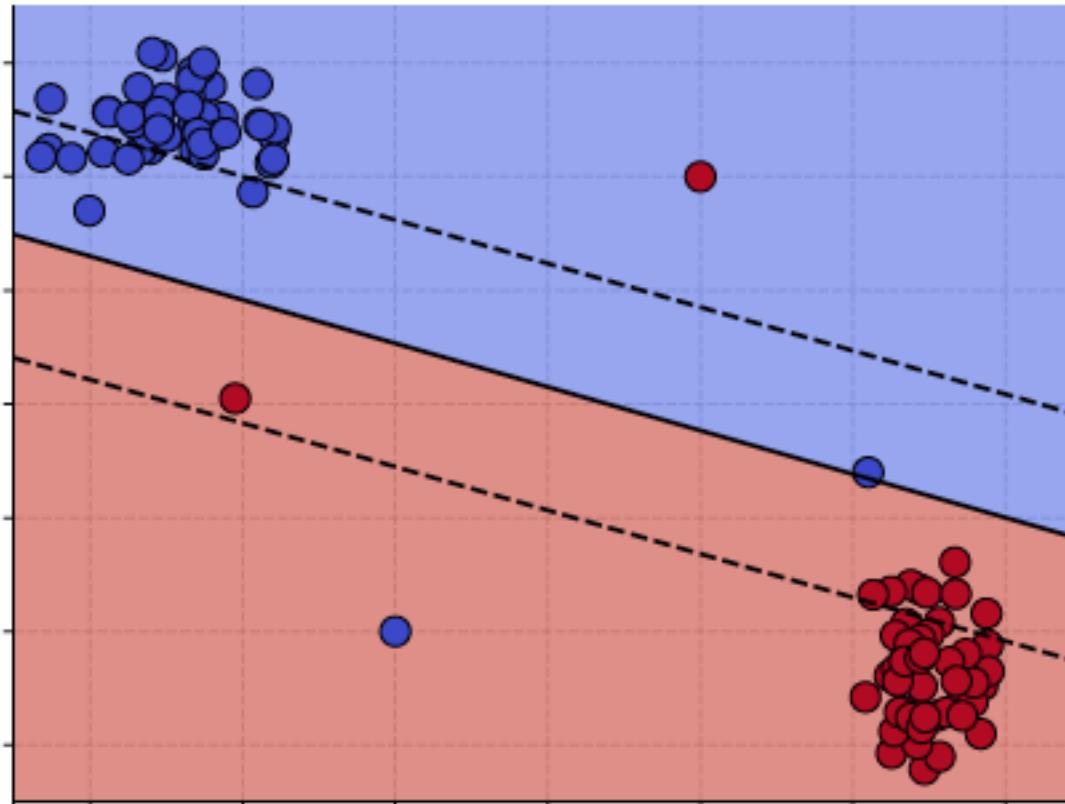
# Soft-margin SVM

- Ce se întâmplă dacă antrenăm un hard-margin SVM pe setul de date de mai jos?



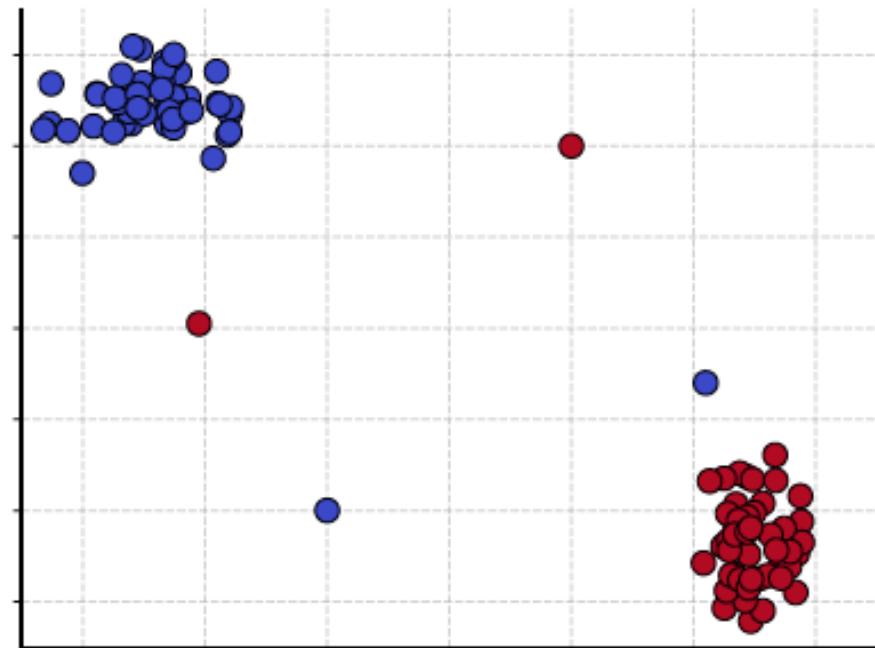
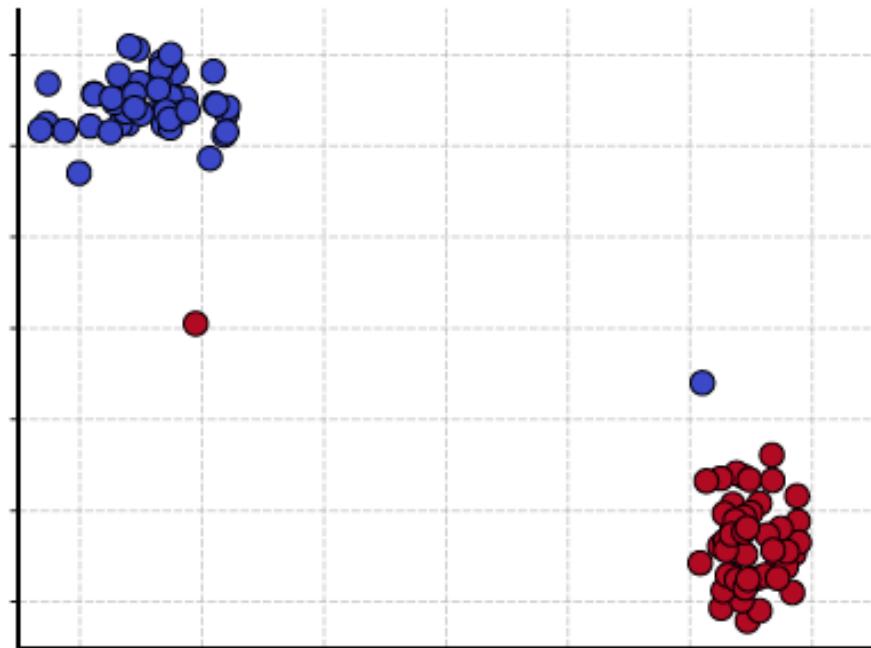
# Soft-margin SVM

- SVM-ul nu reușește să găsească un hiperplan care separă datele perfect pentru că ele sunt liniar neseparabile
- Găsește totuși un hiperplan care face cele mai puține greșeli
  - dacă nu are un număr limită de iterări ciclează...



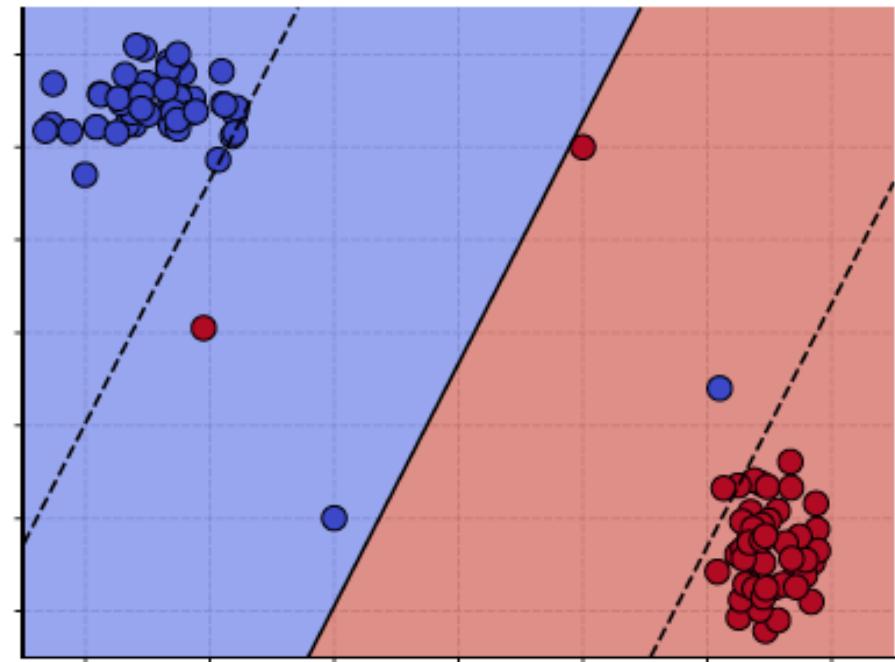
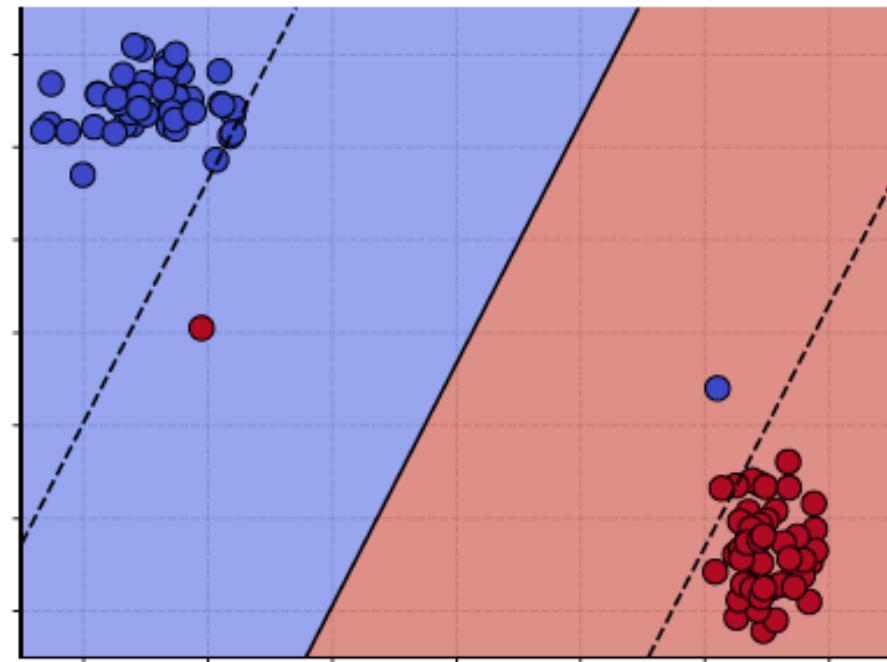
# Soft-margin SVM

- Trade-off între găsirea unui hiperplan cu margine mică care separă perfect datele și găsirea unui hiperplan cu margine mare care face câteva greșeli (exemple de antrenare misclasificate)



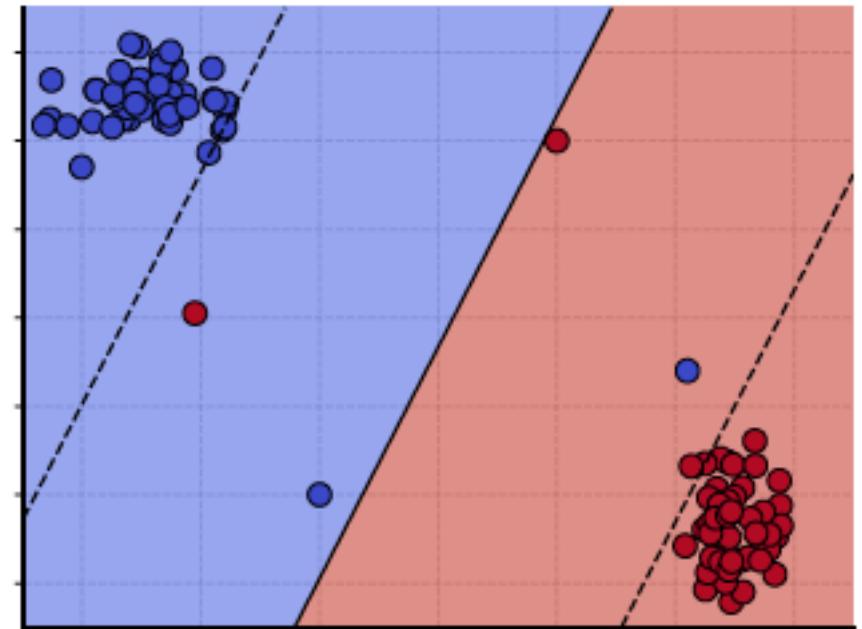
# Soft-margin SVM

- Trade-off între găsirea unui hiperplan cu margine mică care separă perfect datele și găsirea unui hiperplan cu margine mare care face câteva greșeli (exemple de antrenare misclasificate)



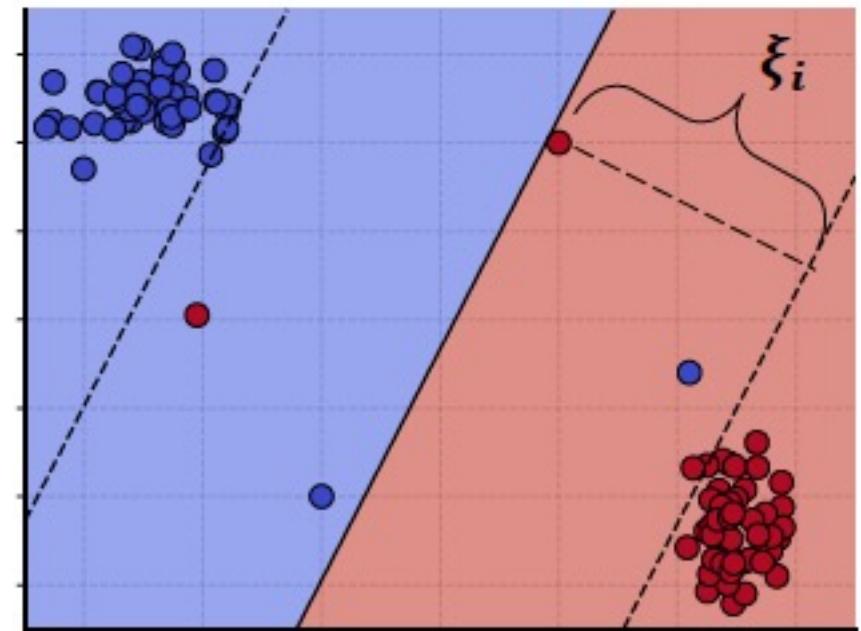
# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)



# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)
- Introducem variabilele  $\xi_i \geq 0$
- Cuantifică cât de mult exemplul de antrenare  $\vec{x}^{(i)}$  este de partea cealaltă a marginii

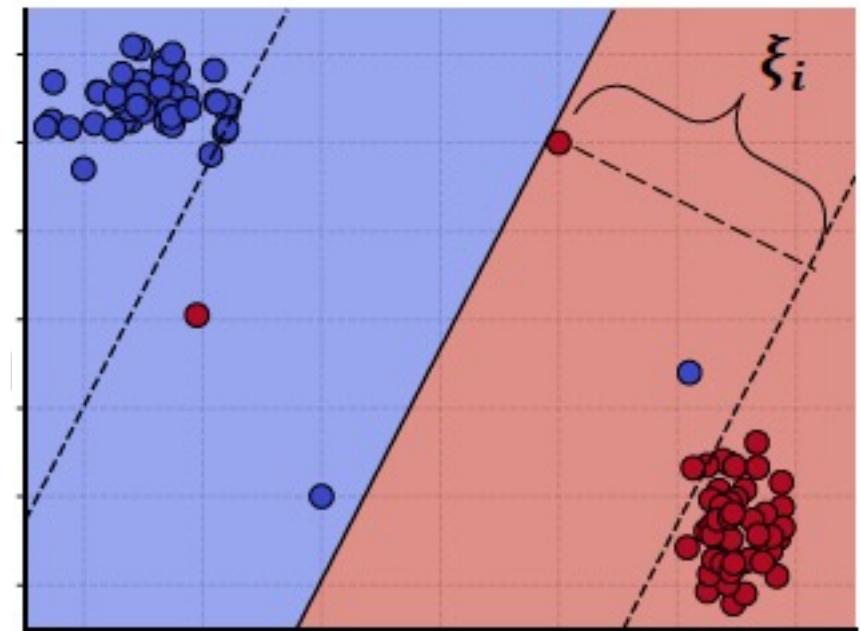


# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)
- Introducem variabilele  $\xi_i \geq 0$
- Căntărește de la 0 la infinit cât de mult exemplul de antrenare  $\vec{x}^{(i)}$  este de partea cealaltă a marginii

Constrângerile  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0$

devin:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq -\xi_i$



# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)
- Introducem variabilele  $\xi_i \geq 0$
- Cuantifică cât de mult exemplul de antrenare  $\vec{x}^{(i)}$  este de partea cealaltă a marginii

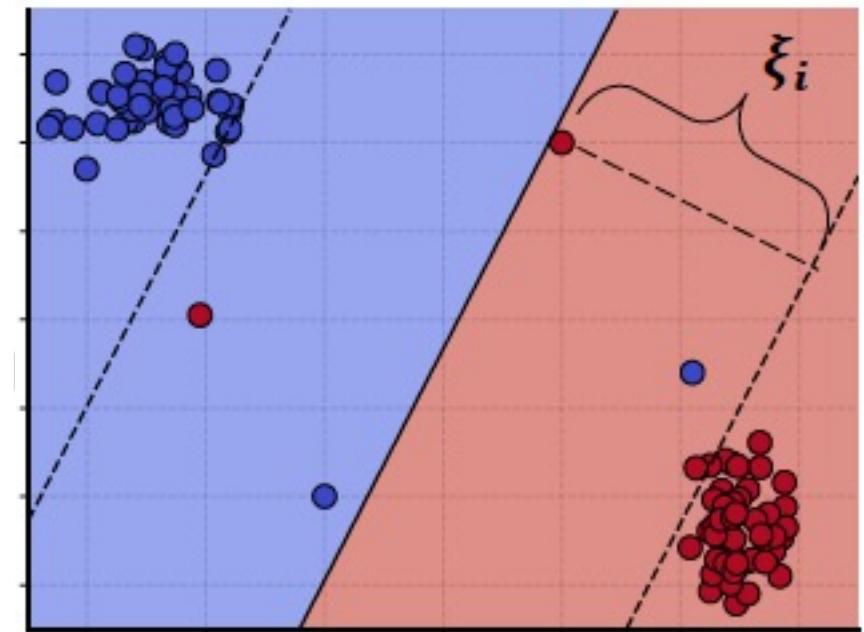
Constrângerile  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0$

devin:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq -\xi_i$

Problema de optimizare nouă:

$$\text{minimize} \frac{\|\vec{w}\|^2}{2} + C \sum_i \xi_i$$

constrângerii:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) \geq 1 - \xi_i$



# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)
- Introducem variabilele  $\xi_i \geq 0$
- Cuantifică cât de mult exemplul de antrenare  $\vec{x}^{(i)}$  este de partea cealaltă a marginii

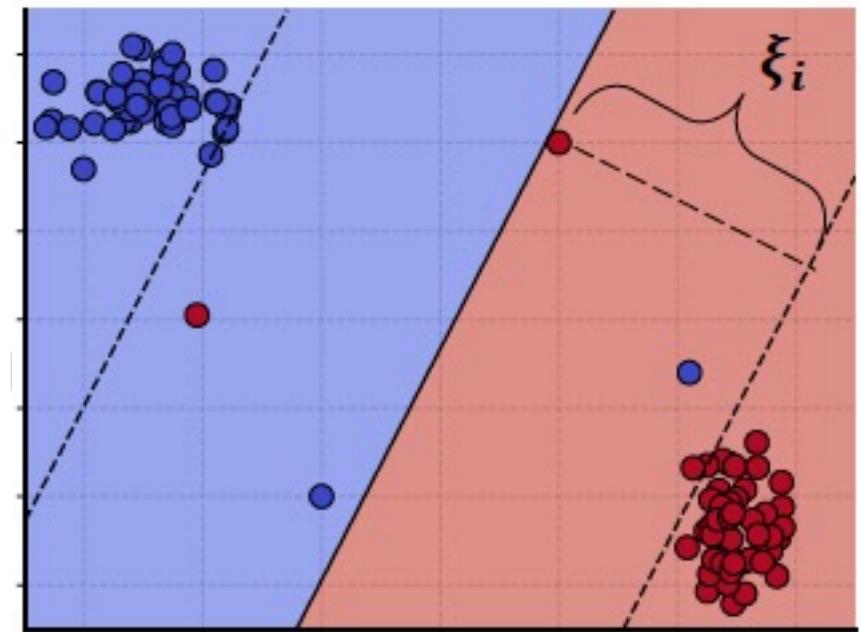
Constrângerile  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0$

devin:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq -\xi_i$

Problema de optimizare nouă:

$$\text{minimize} \frac{\|\vec{w}\|^2}{2} + C \sum_i \xi_i$$

constrângeri:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) \geq 1 - \xi_i$



**Hiperparametrul C controlează tradeoff-ul dintre maximizare marginii și acuratețea modelului**

# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)
- Introducem variabilele  $\xi_i \geq 0$
- Cuantifică cât de mult exemplul de antrenare  $\vec{x}^{(i)}$  este de partea cealaltă a marginii

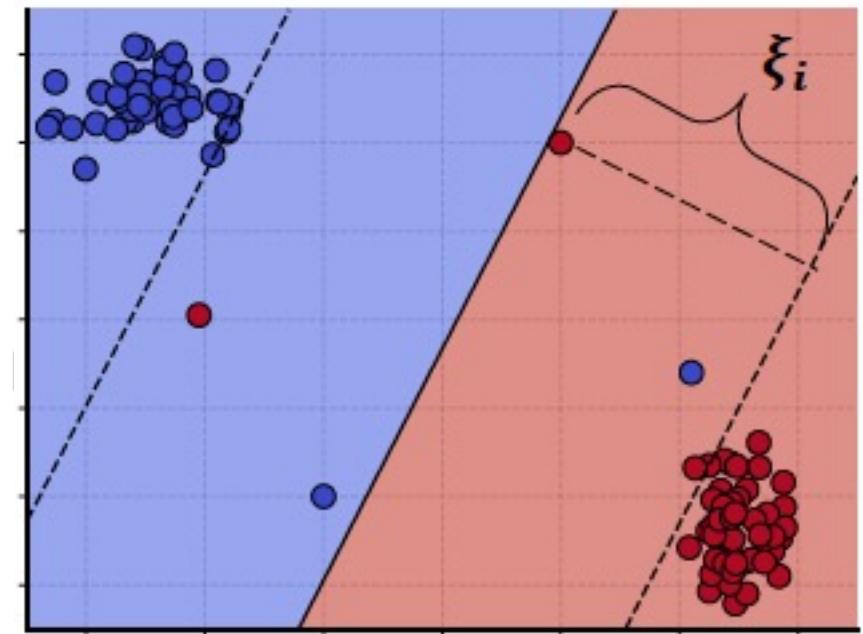
Constrângerile  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0$

devin:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq -\xi_i$

Problema de optimizare nouă:

$$\text{minimize} \frac{\|\vec{w}\|^2}{2} + C \sum_i \xi_i$$

constrângeri:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) \geq 1 - \xi_i$



Valoare mică a lui C (aproape de 0) – maximizează marginea, marginea poate fi încalcată mai mult, penalitate mică

# Soft-margin SVM

- Soft-margin SVM: permite ca anumite exemple de antrenare să fie misclasificate (clasificate greșit), cu un cost (penalitate)
- Introducem variabilele  $\xi_i \geq 0$
- Cuantifică cât de mult exemplul de antrenare  $\vec{x}^{(i)}$  este de partea cealaltă a marginii

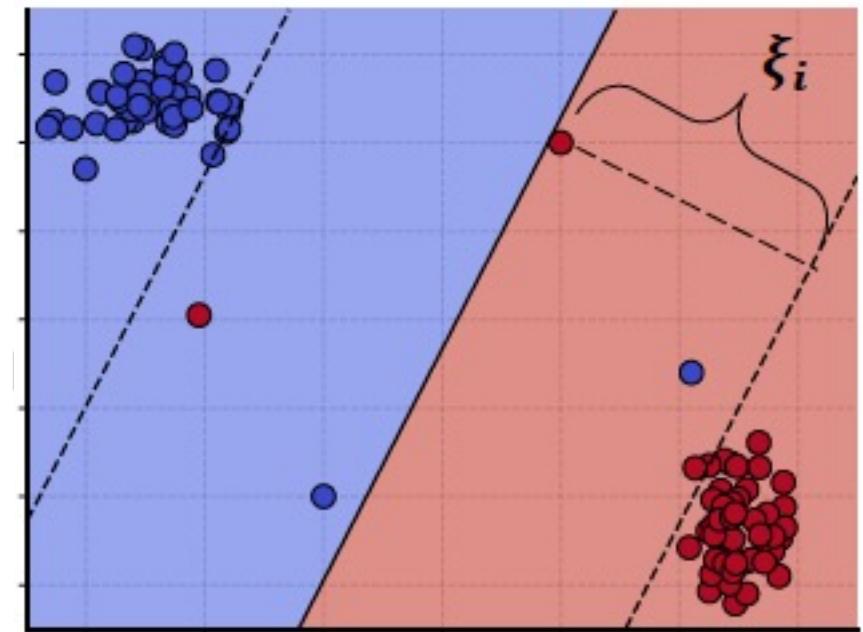
Constrângerile  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq 0$

devin:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) - 1 \geq -\xi_i$

Problema de optimizare nouă:

$$\text{minimize} \frac{\|\vec{w}\|^2}{2} + C \sum_i \xi_i$$

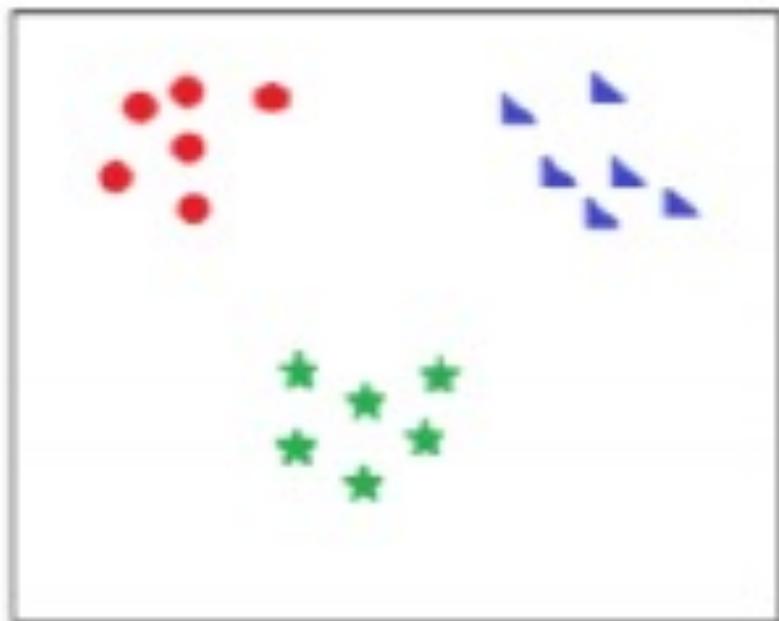
constrângeri:  $y^{(i)}(\langle \vec{x}^{(i)}, \vec{w} \rangle + b) \geq 1 - \xi_i$



Valoare mare a lui C –margine nu prea mare, marginea nu poate fi încalcată mult, penalitate mare

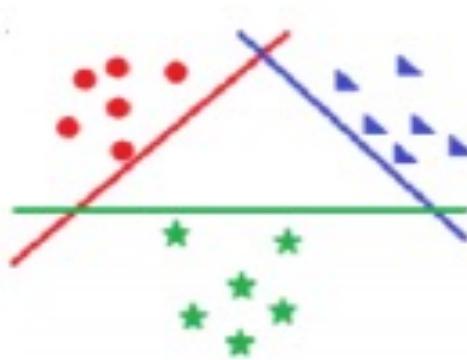
# SVM pentru clasificarea cu mai multe clase

- Un SVM este un clasificator binar (prin construcție). Putem să folosim un SVM în problemele de clasificare cu mai multe clase?



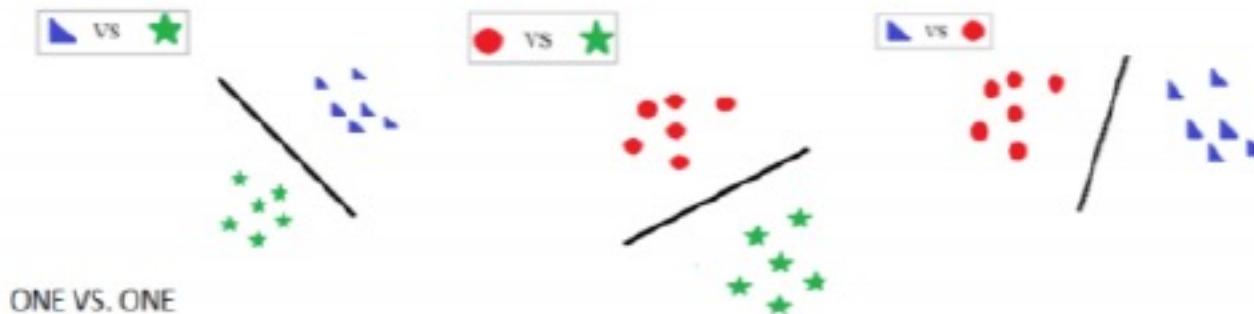
# SVM pentru clasificarea cu mai multe clase

- Un SVM este un clasificator binar (prin construcție). Putem să folosim un SVM în problemele de clasificare cu mai multe clase?
- One-versus-rest (OVR) = One-versus-all (OVA)
  - antrenează  $n$  clasificatori, câte unul pentru fiecare clasă
  - pentru antrenarea fiecărui clasificator: exemplele + sunt cele inițiale, exemplele – sunt date de reuniunea exemplelor celor  $n-1$  clase rămase
  - la inferență (testare): rulăm toți clasificatorii, alegem clasa pe baza clasificatorului cu marginea cea mai mare (cel mai încrezător că exemplul de testare este clasificat + în clasa respectivă)



# SVM pentru clasificarea cu mai multe clase

- Un SVM este un clasificator binar (prin construcție). Putem să folosim un SVM în problemele de clasificare cu mai multe clase?
- One-versus-one (OVO)
  - antrenează  $n(n-1)/2$  clasificatori, câte unul pentru fiecare pereche de clase
  - la inferență (testare): rulăm toți clasificatorii și alegem clasa care a fost selectată de cele mai multe ori



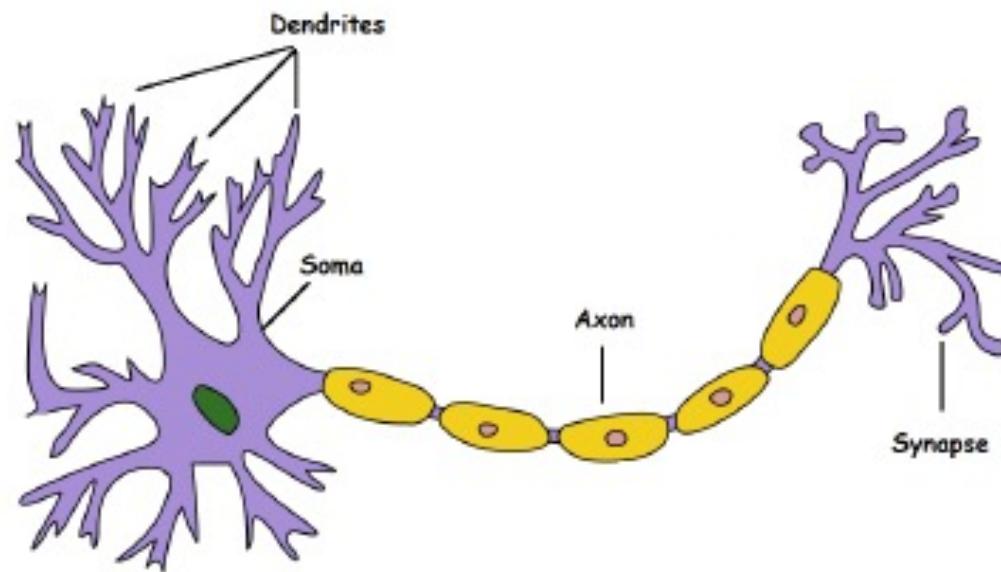
# Perceptronul

# Neuronul biologic

Perceptronul este un clasificator liniar inspirat de neuronul uman.

Cum funcționează un neuron:

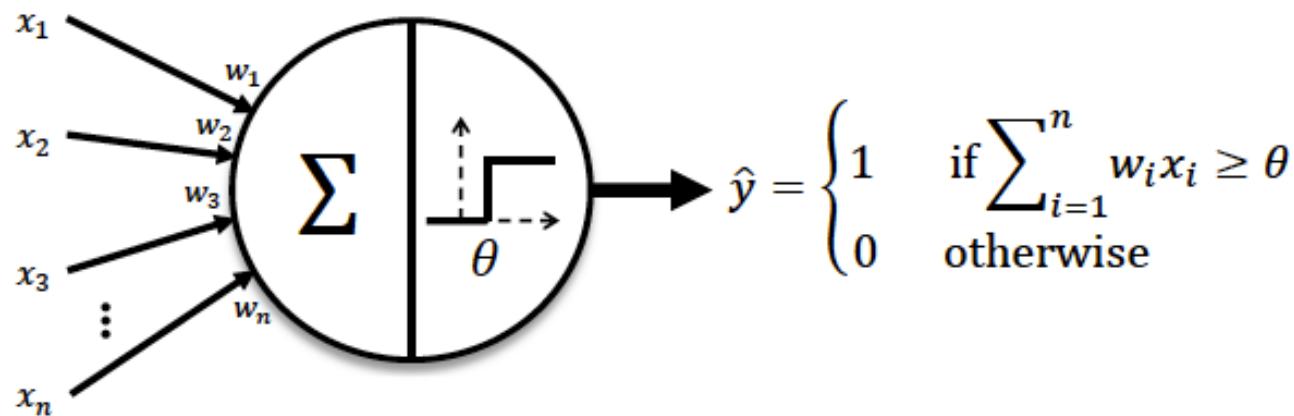
- semnale electrice (intrări) sunt transmise prin **dendrite**
- aceste semnale duc la acumularea unui potențial electric în **corpul neuronului (soma)**
- când potențialul electric acumulat depășește un anumit prag (threshold) un semnal electric (ieșirea neuronului) este transmis prin intermediul unui **axon**
- conectarea axonului cu dendritele altor neuroni se realizează prin **sinapse**



# Neuronul artificial

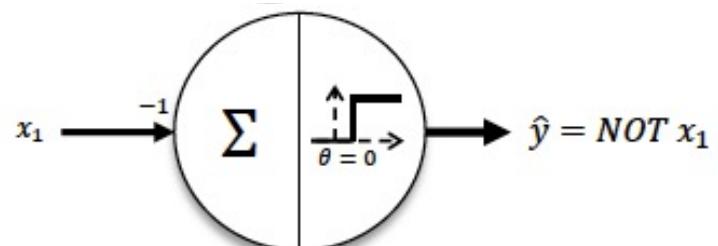
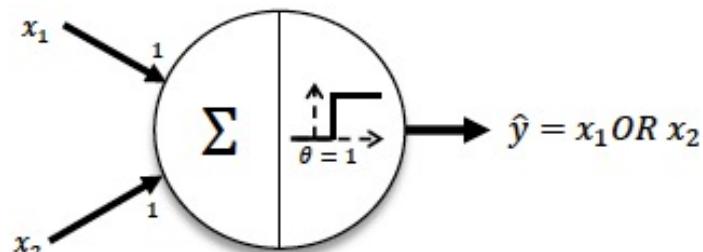
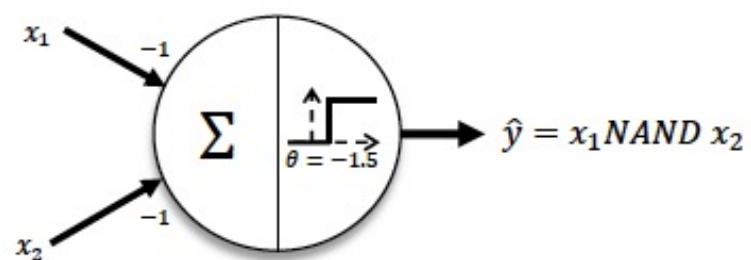
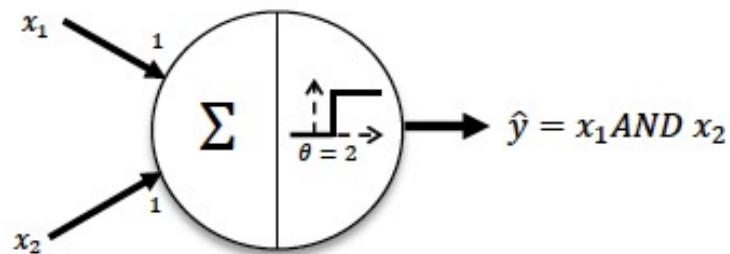
- Primul model al unui neuron artificial a fost propus de Warren McCulloch și Walter Pitts în 1943.
- Conform acestui model semnalele electrice de intrare erau valori booleene (0 - absent, 1 - prezent) de tip *excitator* sau *inhibitor*.
  - dacă numărul de intrări prezente (valoare 1) de tip excitator (pondere 1) era mai mare decât numărul de intrări prezente de tip inhibitor (pondere -1) atunci neuronul se activa
- Matematic, ieșirea neuronului este 1 dacă suma ponderată a intrărilor depășește un prag, altfel este 0.

$$x_1, x_2, \dots, x_n \in \{0,1\}, \quad w_1, w_2, \dots, w_n \in \{-1,1\}, \quad \theta \in \mathbb{R}, \quad \hat{y} \in \{0,1\}$$



# Neuronul artificial

$$\hat{y} = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i x_i \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

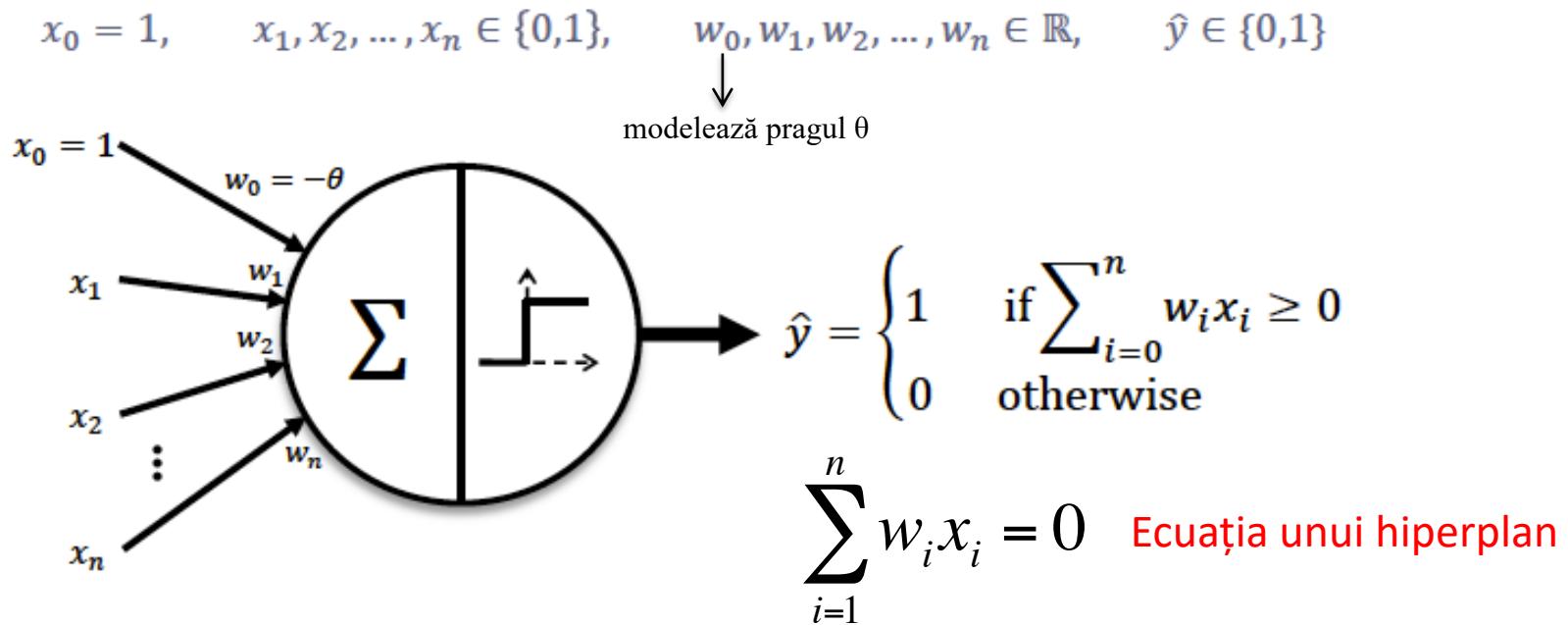


# Regula de învățare a lui Hebb

- Regula de învățare a lui Donald Hebb (1949) spune că legătura dintre doi neuroni devine mai puternică pe măsura ce un neuron trimite semnale electrice care activează celălalt neuron:
  - conform acestei reguli, intrările nu sunt numai de tip excitator sau inhibitor dar pot fi ponderate de sinapse (unele conexiuni dintre neuroni sunt mai puternice decât celelalte)
  - regula de învățare actualiza ponderile sinaptice (creștea ponderea sinaptică a unui neuron asupra altui neuron dacă acesta este activat)

# Perceptronul

- Plecând de la modelul neuronului artificial McCulloch –Pitts și cu ideea lui Hebb în minte, Frank Rosenblatt inventează în 1957 o mașină și un algoritm de învățare asociat, pe care le va numi Perceptron, creat pentru recunoașterea cifrelor din imagini.
  - răspunsul celulelor fotosensibile – intrări, potențiometre – ponderi sinaptice, motoare electrice se ocupau de actualizarea de ponderi



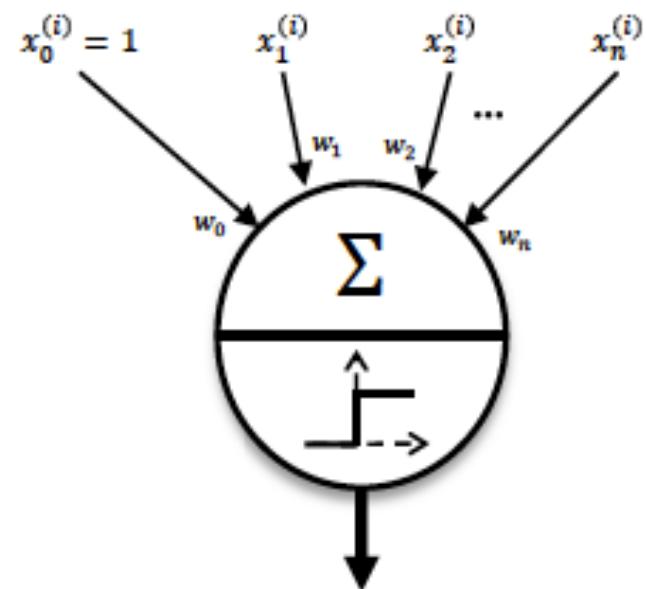
# Algoritmul de învățare al perceptronului

- Mulțimea de exemple de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \{0,1\}^{n+1} (x_0^{(i)} = 1, \forall i), y^{(i)} \in \{0,1\}$$

- Ponderile  $w_j$  sunt inițializate cu 0

*Scopul învățării este de a învăța un set de ponderi  $w_0, w_1, \dots, w_n$  care minimizează eroare reală. Aceste ponderi se învăță pe mulțimea de antrenare  $E$ .*



$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

# Algoritmul de învățare al perceptronului

- Mulțimea de exemple de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \{0,1\}^{n+1} (x_0^{(i)} = 1, \forall i), y^{(i)} \in \{0,1\}$$

- Ponderile  $w_j$  sunt inițializate cu 0
- Pentru fiecare exemplu de antrenare  $(\vec{x}^{(i)}, y^{(i)})$  din mulțimea E:
  - dacă eticheta prezisă = eticheta reală  $\hat{y}^{(i)} == y^{(i)}$  nu facem actualizare
  - dacă  $\hat{y}^{(i)} == 0$  și  $y^{(i)} == 1$  creștem ponderile tuturor intrărilor active
  - dacă  $\hat{y}^{(i)} == 1$  și  $y^{(i)} == 0$  scădem ponderile tuturor intrărilor active

$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

# Algoritmul de învățare al perceptronului

- Mulțimea de exemple de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \{0,1\}^{n+1} (x_0^{(i)} = 1, \forall i), y^{(i)} \in \{0,1\}$$

- Ponderile  $w_j$  sunt inițializate cu 0
- Pentru fiecare exemplu de antrenare  $(\vec{x}^{(i)}, y^{(i)})$  din mulțimea E:
  - dacă eticheta prezisă = eticheta reală  $\hat{y}^{(i)} == y^{(i)}$  nu facem actualizare
  - dacă  $\hat{y}^{(i)} == 0$  și  $y^{(i)} == 1$  creștem ponderile tuturor intrărilor active
  - dacă  $\hat{y}^{(i)} == 1$  și  $y^{(i)} == 0$  scădem ponderile tuturor intrărilor active
- Regula de actualizare a perceptronului

$$w_j = w_j + \Delta w_j, \text{ unde}$$

$$\Delta w_j = \eta(y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

Cantitatea prin care schimbăm ponderea  
(rata de învățare)

Setează semnul actualizării

Selectează intrările active

$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

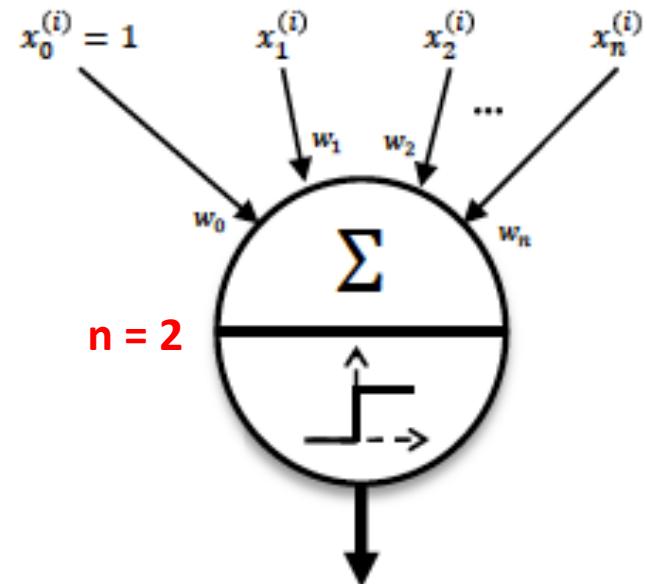
# Algoritmul de învățare al perceptronului

```
1 def perceptron(X, y, n_epochs, η):
2     m, n = X.shape # number of samples, number of inputs
3     for j in range(n):
4         wj = 0
5     for epoch in range(n_epochs): # an “epoch” is a run through all training data.
6         for i in range(m): # a “training step” is one update of the weights.
7             ŷ(i) = unit_step_function(Σj=0n wjxj(i))      ŷ(i) = {1 if Σj=0n wjxj(i) ≥ 0
8             for j in range(n):
9                 wj += η(y(i) - ŷ(i)) · xj(i)
```

# Exemplu – învățarea funcției OR

$$E = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \{0,1\}^{n+1} (x_0^{(i)} = 1, \forall i), y^{(i)} \in \{0,1\}$$

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1



$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} > 0 \\ 0 & \text{otherwise} \end{cases}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$
$\vec{x}^{(1)}$	1	0	0	0	0	0	0	0
$\vec{x}^{(2)}$	1	0	1	1	0	0	0	0
$\vec{x}^{(3)}$	1	1	0	1	0	0	0	0
$\vec{x}^{(4)}$	1	1	1	1	0	0	0	0

Accuracy: 25%

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	1	0	0	0	0	0	0	0	0	0	0
$\vec{x}^{(2)}$	1	0	1	1	0	0	0	0	0	0	0
$\vec{x}^{(3)}$	1	1	0	1	0	0	0	0	0	0	0
$\vec{x}^{(4)}$	1	1	1	1	0	0	0	0	0	0	0
Accuracy: 25%											
	$\vec{x}^{(1)}$										
	$\vec{x}^{(2)}$										
	$\vec{x}^{(3)}$										
	$\vec{x}^{(4)}$										

Epoch 1

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

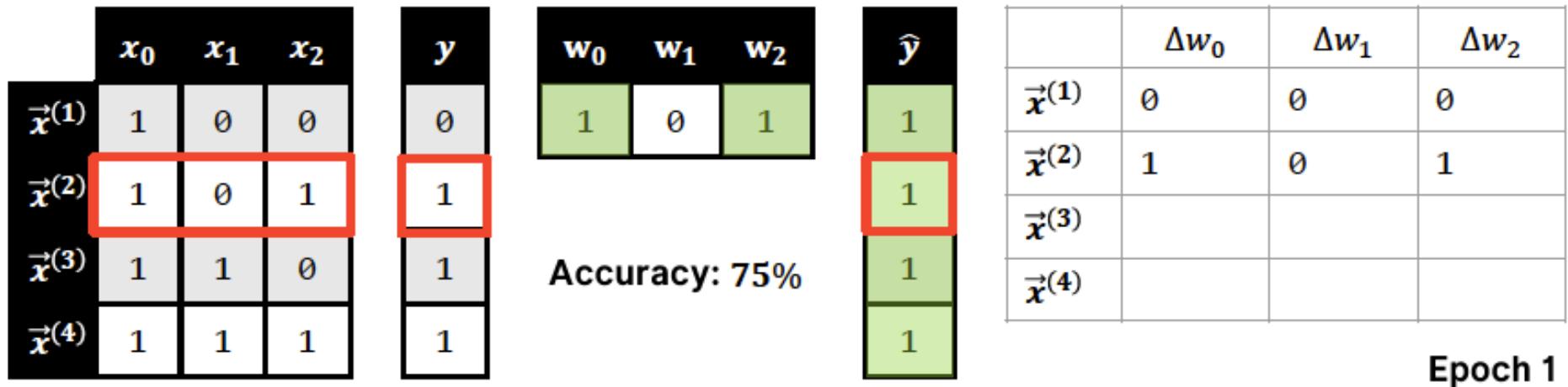
# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$		$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	1	0	0	0	0	0	0	0	$\vec{x}^{(1)}$	0	0	0
$\vec{x}^{(2)}$	1	0	1	1	0	0	0	0	$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	1	1	0	1	0	0	0	0	$\vec{x}^{(3)}$			
$\vec{x}^{(4)}$	1	1	1	1	0	0	0	0	$\vec{x}^{(4)}$			
Accuracy: 25%									Epoch 1			

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR



$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

	$w_0$	$w_1$	$w_2$
	1	0	1

	$\hat{y}$
	1
	1
	1
	1

Accuracy: 75%

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	0	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	0	0	0
$\vec{x}^{(4)}$			

Epoch 1

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

	$w_0$	$w_1$	$w_2$
	1	0	1

Accuracy: 75%

	$\hat{y}$
	1
	1
	1

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	0	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	0	0	0
$\vec{x}^{(4)}$	0	0	0

Epoch 1

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	1	0	0	0	1	0	1	1	-1	0	0
$\vec{x}^{(2)}$	1	0	1	1	1	0	1	1			
$\vec{x}^{(3)}$	1	1	0	1	1	1	0	1			
$\vec{x}^{(4)}$	1	1	1	1	1	1	1	1			
Accuracy: 75%											
	$\vec{x}^{(1)}$										
	$\vec{x}^{(2)}$										
	$\vec{x}^{(3)}$										
	$\vec{x}^{(4)}$										

Epoch 2

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	1	0	0	0	0	0	1	0	-1	0	0
$\vec{x}^{(2)}$	1	0	1	1	0	0	1	1			
$\vec{x}^{(3)}$	1	1	0	1	0	0	1	0			
$\vec{x}^{(4)}$	1	1	1	1	0	0	1	1			
Accuracy: 75%											
	$\vec{x}^{(1)}$										
	$\vec{x}^{(2)}$										
	$\vec{x}^{(3)}$										
	$\vec{x}^{(4)}$										

Epoch 2

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

	$w_0$	$w_1$	$w_2$
	0	0	1

	$\hat{y}$
	0
	1
	0
	1

Accuracy: 75%

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$	0	0	0
$\vec{x}^{(3)}$			
$\vec{x}^{(4)}$			

Epoch 2

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

Accuracy: 75%

	$\hat{y}$
	0
	1
	0
	1

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$	0	0	0
$\vec{x}^{(3)}$	1	1	0
$\vec{x}^{(4)}$			

Epoch 2

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

Accuracy: 75%

	$w_0$	$w_1$	$w_2$
	1	1	1

	$\hat{y}$
	1
	1
	1
	1

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$	0	0	0
$\vec{x}^{(3)}$	1	1	0
$\vec{x}^{(4)}$			

Epoch 2

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$		$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	1	0	0	0	1	1	1	1	$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$	1	0	1	1				1	$\vec{x}^{(2)}$	0	0	0
$\vec{x}^{(3)}$	1	1	0	1				1	$\vec{x}^{(3)}$	1	0	
$\vec{x}^{(4)}$	1	1	1	1				1	$\vec{x}^{(4)}$	0	0	0
Accuracy: 75%												
Epoch 2												

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

Accuracy: 75%

	$\hat{y}$
	1
	1
	1
	1

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$			
$\vec{x}^{(3)}$			
$\vec{x}^{(4)}$			

Epoch 3

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$
$\vec{x}^{(1)}$	1	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	1	1	0
$\vec{x}^{(4)}$	1	1	1

**y**

$w_0$	$w_1$	$w_2$
0	1	1

$\hat{y}$

**Accuracy: 100%**

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$			
$\vec{x}^{(3)}$			
$\vec{x}^{(4)}$			

## Epoch 3

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

	$w_0$	$w_1$	$w_2$
	0	1	1

Accuracy: 100%

	$\hat{y}$
	0
	1
	1
	1

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$	0	0	0
$\vec{x}^{(3)}$	0	0	0
$\vec{x}^{(4)}$	0	0	0

Epoch 3

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$
$\vec{x}^{(1)}$	1	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	1	1	0
$\vec{x}^{(4)}$	1	1	1

$y$
0
1
1
1

$w_0$	$w_1$	$w_2$
0	1	1

$\hat{y}$
0
1
1
1

Accuracy: 100%

OR

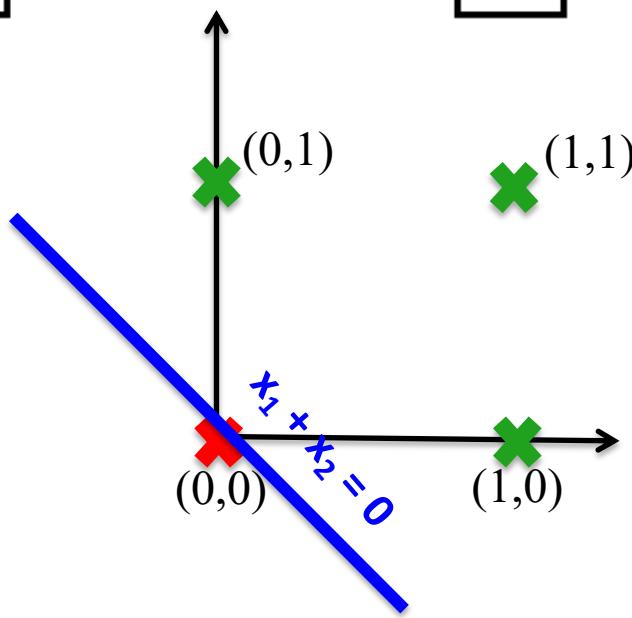
- ✖ Eticheta 1
- ✖ Eticheta 0

$$w_0x_0 + w_1x_1 + w_2x_2 > 0$$

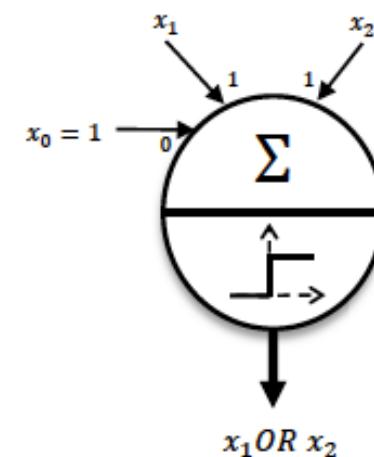
$$x_1 + x_2 > 0$$

$$w_0x_0 + w_1x_1 + w_2x_2 \leq 0$$

$$x_1 + x_2 \leq 0$$



Perceptronul a învățat  
funcția OR în 3 epoci



$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} > 0 \\ 0 & \text{otherwise} \end{cases}$$

# Limitările perceptronului

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0$$

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0 \Rightarrow w_0 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_2$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_1$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0 \Rightarrow w_0 < -w_1 - w_2$$

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0 \Rightarrow w_0 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_2$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_1$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0 \Rightarrow w_0 < -w_1 - w_2$$

$$\left. \begin{array}{l} 2w_0 \geq -w_1 - w_2 \\ 2w_0 > w_0 \end{array} \right\} \Rightarrow 2w_0 > w_0 \Rightarrow w_0 > 0$$

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0 \Rightarrow w_0 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_2$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_1$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0 \Rightarrow w_0 < -w_1 - w_2$$

$$\left. \begin{array}{l} 2w_0 \geq -w_1 - w_2 \\ 2w_0 > w_0 \end{array} \right\} \Rightarrow 2w_0 > w_0 \Rightarrow w_0 > 0$$



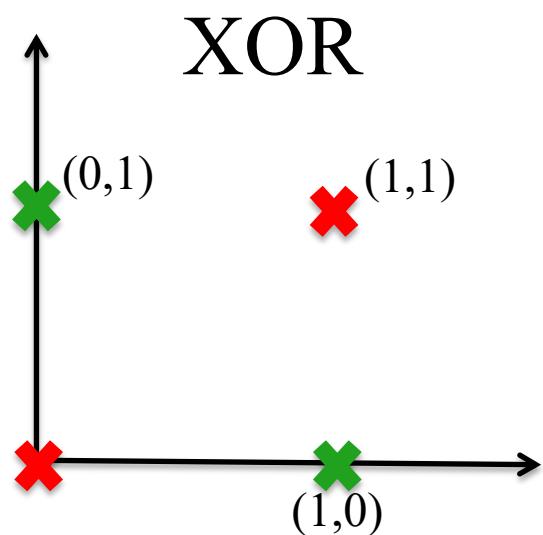
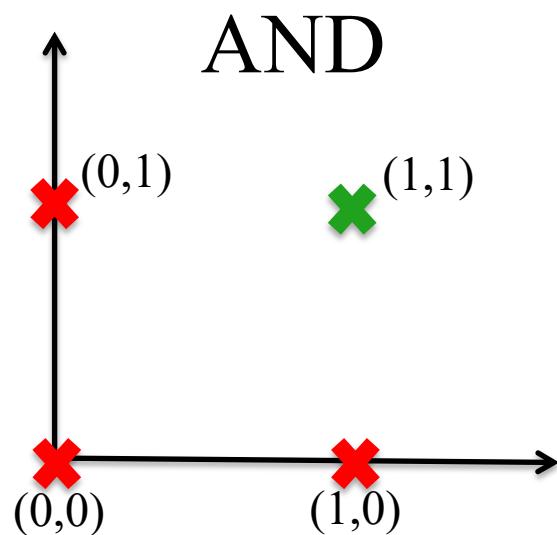
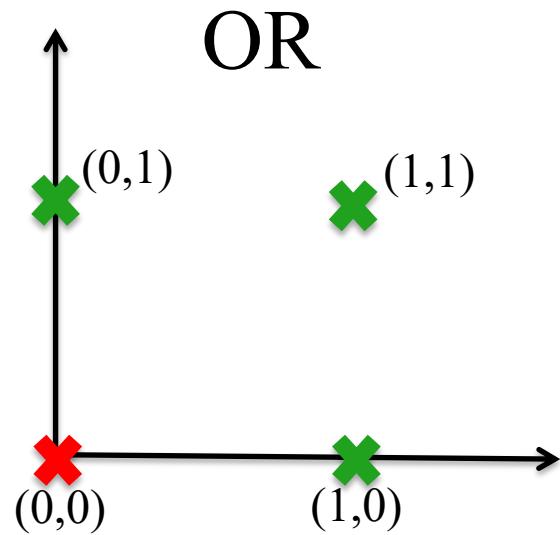
**Contradicție**

- Perceptronul nu poate învăța XOR oricât de mulți pași ar face
- Mai mult, Perceptronul poate învăță numai clase care sunt liniar separabile

# Teorema de convergență a perceptronului

- Dacă mulțimea de antrenare  $E$  este liniar separabilă cu margine  $\gamma$ , algoritmul de învățare a perceptronului este garantat că va converge într-un număr finit de pași către o soluție în care nu se fac greșeli pe mulțimea de antrenare
- Numărul de pași  $k$  satisfac relația  $k \leq \frac{R^2}{\gamma^2}$ , unde  $R$  este raza sferei din spațiul caracteristicilor care cuprinde toate exemplele de antrenare
- Teorema spune că va converge către o soluție, nu este necesar să fie o soluție bună (SVM oferă soluția de margine maximă). Dacă mulțimea de antrenare  $E$  nu este liniar separabilă algoritmul nu va converge către o soluție.

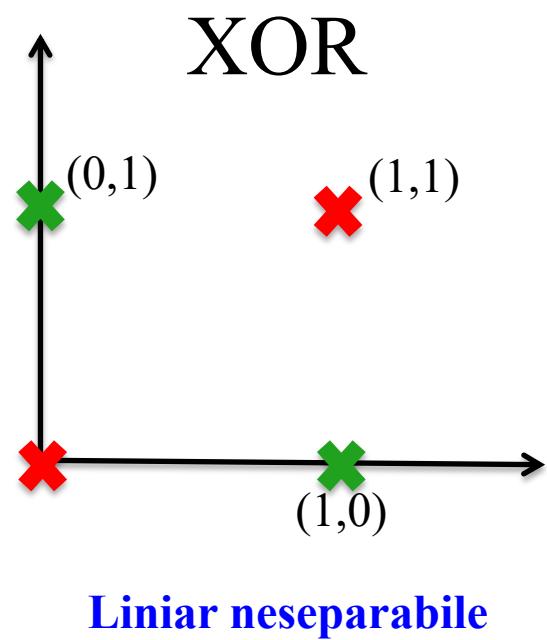
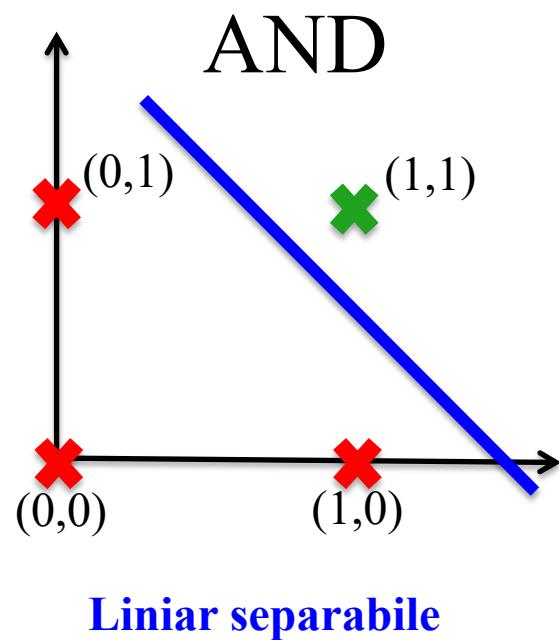
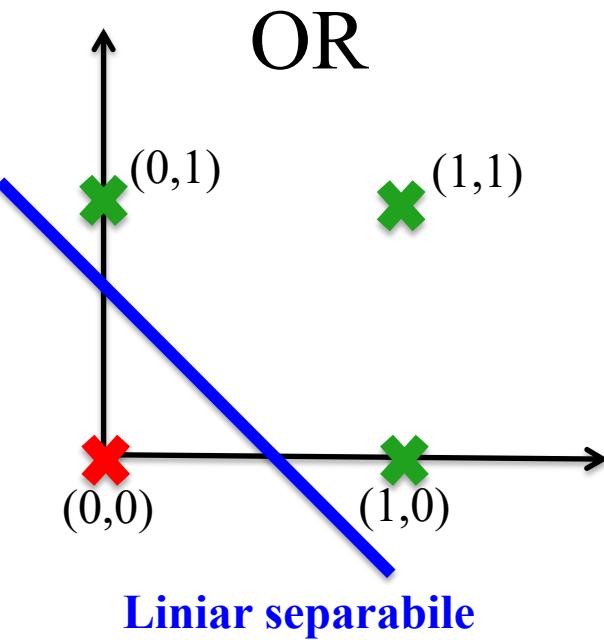
# Perspectiva geometrică



✗ Eticheta 0

✖ Eticheta 1

# Perspectiva geometrică



✗ Eticheta 0

✖ Eticheta 1

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 6

< All teams



## Inteligenta Artificiala - CTI-3-20...

Home page

Class Notebook

Assignments

Grades

Reflect

Insights

### Channels

General

Curs

Laborator



Curs

Posts

Files

Notes

+

+ New

Upload

Edit in grid view

Share

Sync

Download

Add shortcut to OneDrive

...

Curs

CTI-IA-bonus curs 2022\_2023.pdf A few seconds ago DUMITRU BOGDA...

Bibliografie October 5 DUMITRU BOGDA...

Cursuri October 6 DUMITRU BOGDA...

# Proiect – deadline mărit cu o săptămână

- sincronizare bună curs/laborator
- paletă largă de metode clasificare: kNN, Bayes, SVM, perceptron, rețele de perceptri
- număr mic de participanți până acum (aprox. 50)

# Recapitulare – cursul trecut

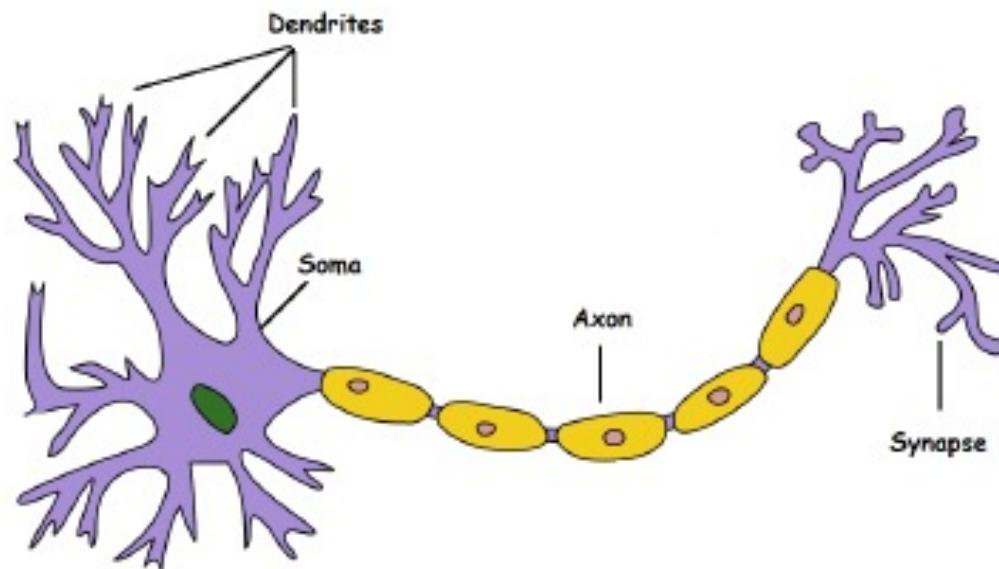
1. Mașini cu vectori support (SVMs – support vector machines)
2. Perceptronul

# Perceptronul

# Neuronul biologic

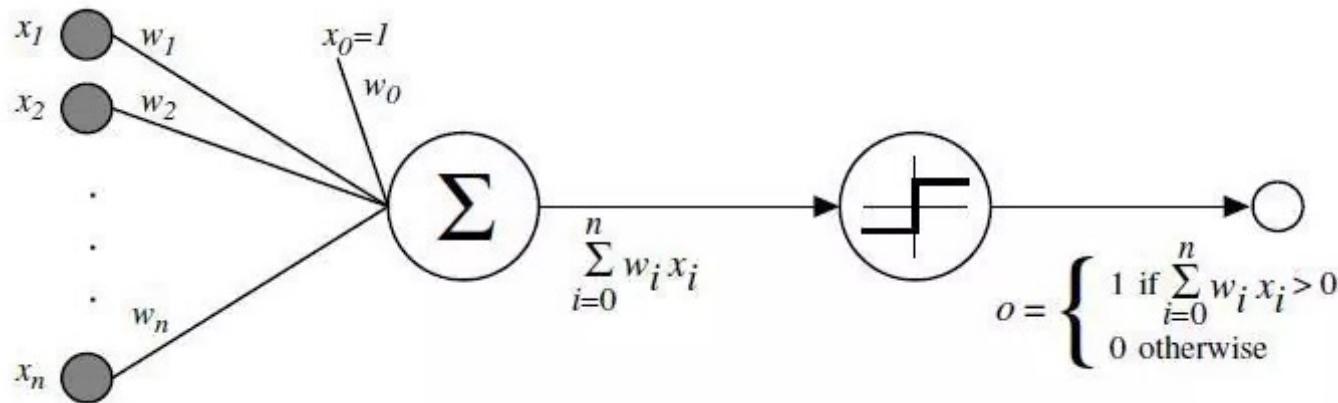
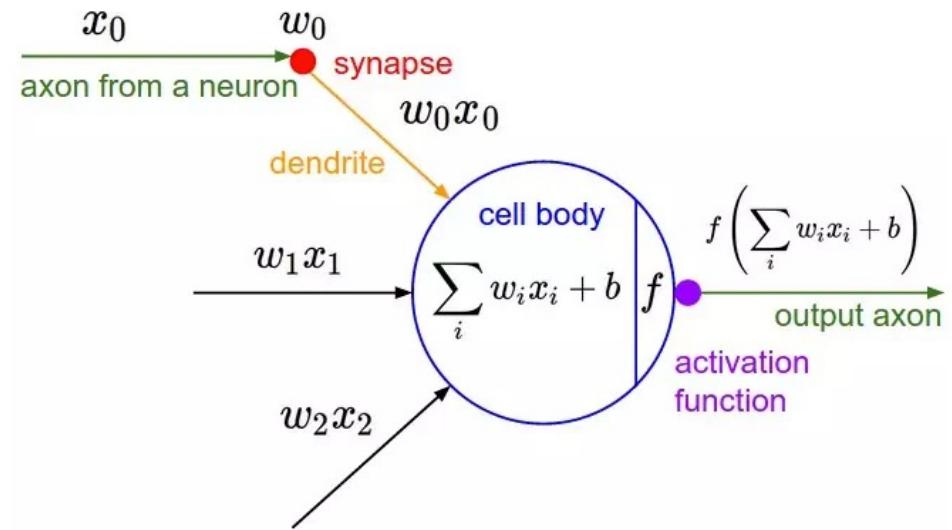
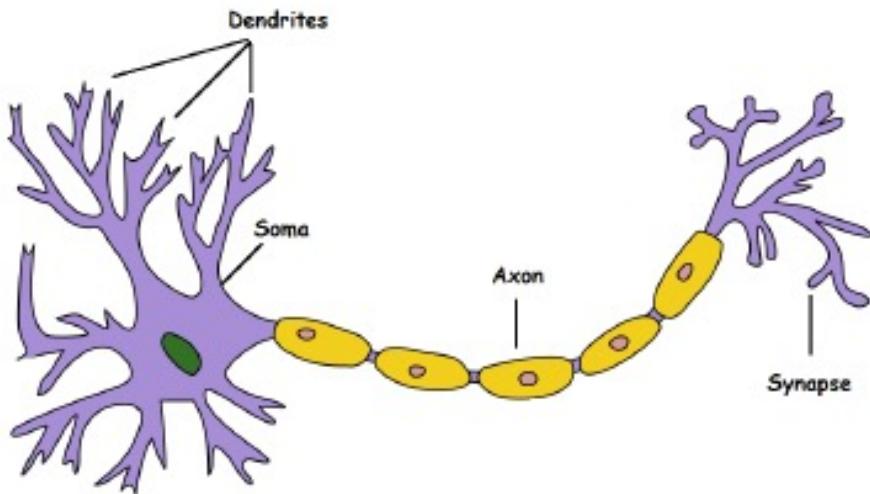
Cum funcționează un neuron:

- semnale electrice (intrări) sunt transmise prin **dendrite**
- aceste semnale duc la acumularea unui potențial electric în **corpul neuronului (soma)**
- când potențialul electric acumulat depășește un anumit prag (threshold) un semnal electric (ieșirea neuronului) este transmis prin intermediul unui **axon**
- conectarea axonului cu dendritele altor neuroni se realizează prin **sinapse**



# Perceptronul

Perceptronul este un clasificator liniar inspirat de neuronul uman.



# Algoritmul de învățare al perceptronului

- Mulțimea de exemple de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \{0,1\}^{n+1} (x_0^{(i)} = 1, \forall i), y^{(i)} \in \{0,1\}$$

- Ponderile  $w_j$  sunt inițializate cu 0 (pe acestea le învățăm)
- Pentru fiecare exemplu de antrenare  $(\vec{x}^{(i)}, y^{(i)})$  din mulțimea E:
  - dacă eticheta prezisă = eticheta reală  $\hat{y}^{(i)} == y^{(i)}$  nu facem actualizare
  - dacă  $\hat{y}^{(i)} == 0$  și  $y^{(i)} == 1$  creștem ponderile tuturor intrărilor active
  - dacă  $\hat{y}^{(i)} == 1$  și  $y^{(i)} == 0$  scădem ponderile tuturor intrărilor active
- Regula de actualizare a perceptronului

$$w_j = w_j + \Delta w_j, \text{ unde}$$

$$\Delta w_j = \eta(y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

Cantitatea prin care schimbăm ponderea  
(rata de învățare)

Setează semnul actualizării

Selectează intrările active

$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

# Algoritmul de învățare al perceptronului

```
1 def perceptron(X, y, n_epochs, η):
2     m, n = X.shape # number of samples, number of inputs
3     for j in range(n):
4         wj = 0
5     for epoch in range(n_epochs): # an “epoch” is a run through all training data.
6         for i in range(m): # a “training step” is one update of the weights.
7             ŷ(i) = unit_step_function(Σj=0n wjxj(i))      ŷ(i) = {1 if Σj=0n wjxj(i) ≥ 0
8             for j in range(n):
9                 wj += η(y(i) - ŷ(i)) · xj(i)
```

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$	$w_0$	$w_1$	$w_2$	$\hat{y}$
$\vec{x}^{(1)}$	1	0	0	0	0	0	0	0
$\vec{x}^{(2)}$	1	0	1	1	0	0	0	0
$\vec{x}^{(3)}$	1	1	0	1	0	0	0	0
$\vec{x}^{(4)}$	1	1	1	1	0	0	0	0

Accuracy: 25%

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$
$\vec{x}^{(1)}$	1	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	1	1	0
$\vec{x}^{(4)}$	1	1	1

**y**

$w_0$	$w_1$	$w_2$
1	0	1

**Accuracy: 75%**

$\hat{y}$

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	0	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	0	0	0
$\vec{x}^{(4)}$	0	0	0

## Epoch 1

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

**Accuracy: 75%**

$\hat{y}$		$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
1	$\vec{x}^{(1)}$	-1	0	0
1	$\vec{x}^{(2)}$	0	0	0
1	$\vec{x}^{(3)}$	1	1	0
1	$\vec{x}^{(4)}$	0	0	0

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$	$y$
$\vec{x}^{(1)}$	1	0	0	0
$\vec{x}^{(2)}$	1	0	1	1
$\vec{x}^{(3)}$	1	1	0	1
$\vec{x}^{(4)}$	1	1	1	1

	$w_0$	$w_1$	$w_2$
	0	1	1

Accuracy: 100%

	$\hat{y}$
	0
	1
	1
	1

	$\Delta w_0$	$\Delta w_1$	$\Delta w_2$
$\vec{x}^{(1)}$	-1	0	0
$\vec{x}^{(2)}$	0	0	0
$\vec{x}^{(3)}$	0	0	0
$\vec{x}^{(4)}$	0	0	0

Epoch 3

$$\eta = 1$$

$$\Delta w_j = (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

# Exemplu – învățarea funcției OR

	$x_0$	$x_1$	$x_2$
$\vec{x}^{(1)}$	1	0	0
$\vec{x}^{(2)}$	1	0	1
$\vec{x}^{(3)}$	1	1	0
$\vec{x}^{(4)}$	1	1	1

$y$
0
1
1
1

$w_0$	$w_1$	$w_2$
0	1	1

$\hat{y}$
0
1
1
1

Accuracy: 100%

OR

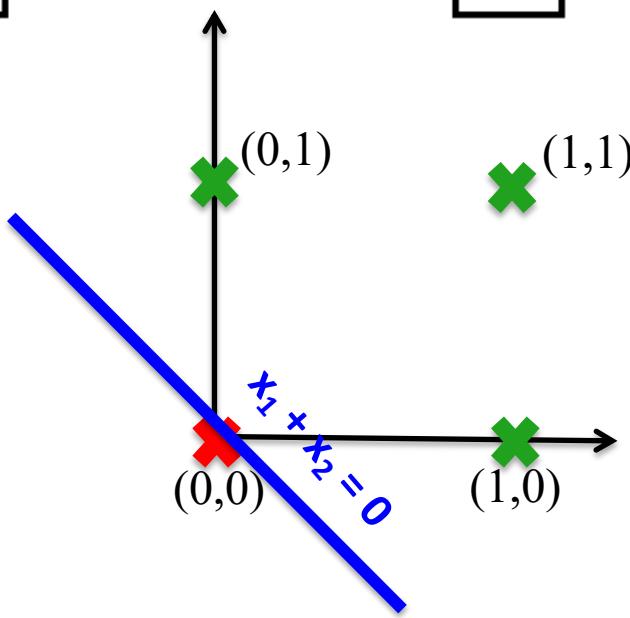
- ✖ Eticheta 1
- ✖ Eticheta 0

$$w_0x_0 + w_1x_1 + w_2x_2 > 0$$

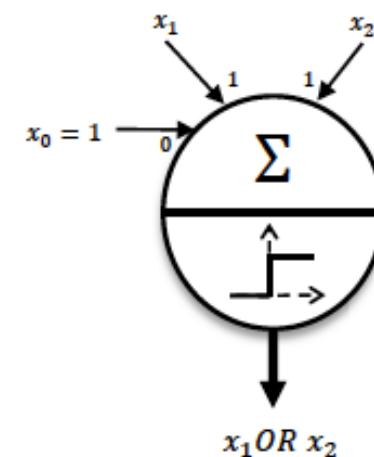
$$x_1 + x_2 > 0$$

$$w_0x_0 + w_1x_1 + w_2x_2 \leq 0$$

$$x_1 + x_2 \leq 0$$



Perceptronul a învățat  
funcția OR în 3 epoci

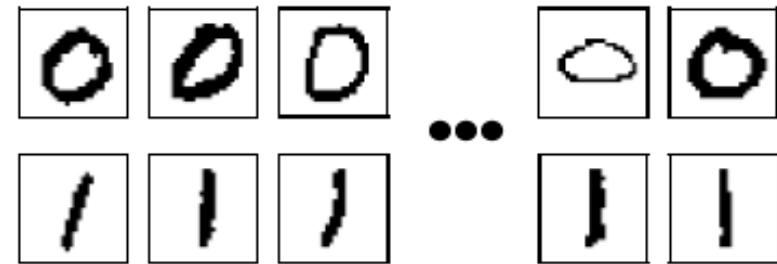
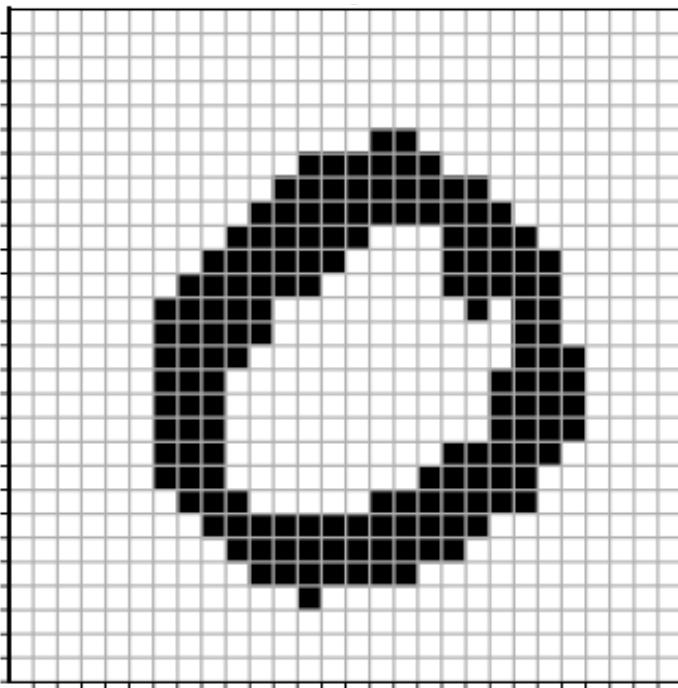


$$\hat{y}^{(i)} = \begin{cases} 1 & \text{if } \sum_{j=0}^n w_j x_j^{(i)} > 0 \\ 0 & \text{otherwise} \end{cases}$$

# Exemplu – Recunoașterea cifrelor

Putem antrena un perceptron care să discrimineze între imagini cu cifre scrise de mână cu cifrele 0 și 1?

**$28 \times 28$  valori binare**

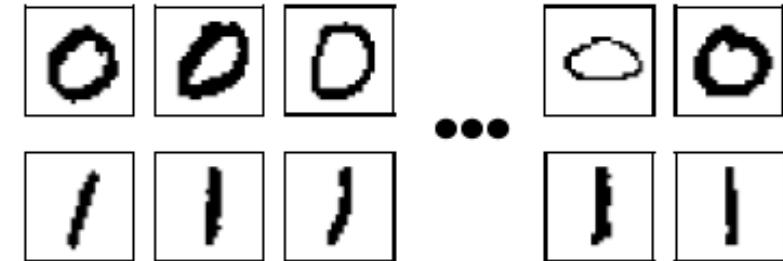
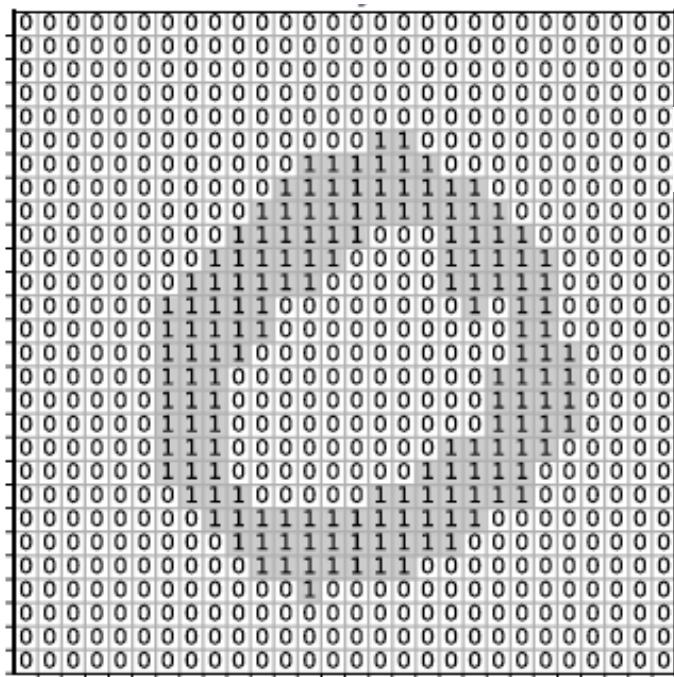


**100 de exemple pentru fiecare clasă**

# Exemplu – Recunoașterea cifrelor

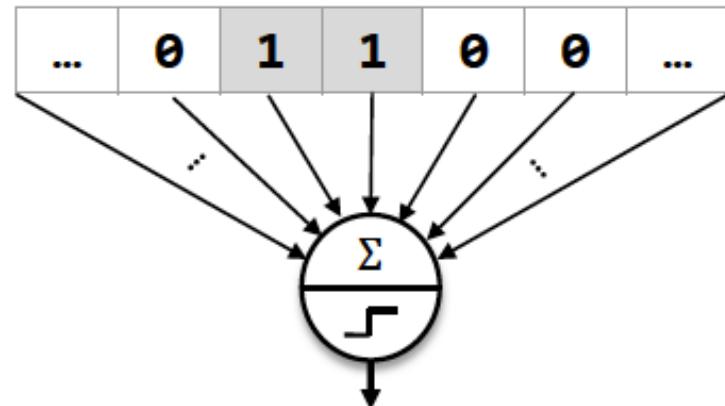
Putem antrena un perceptron care să discrimineze între imagini cu cifre scrise de mâna cu cifrele 0 și 1?

**$28 \times 28$  valori binare**



**100 de exemple pentru fiecare clasă**

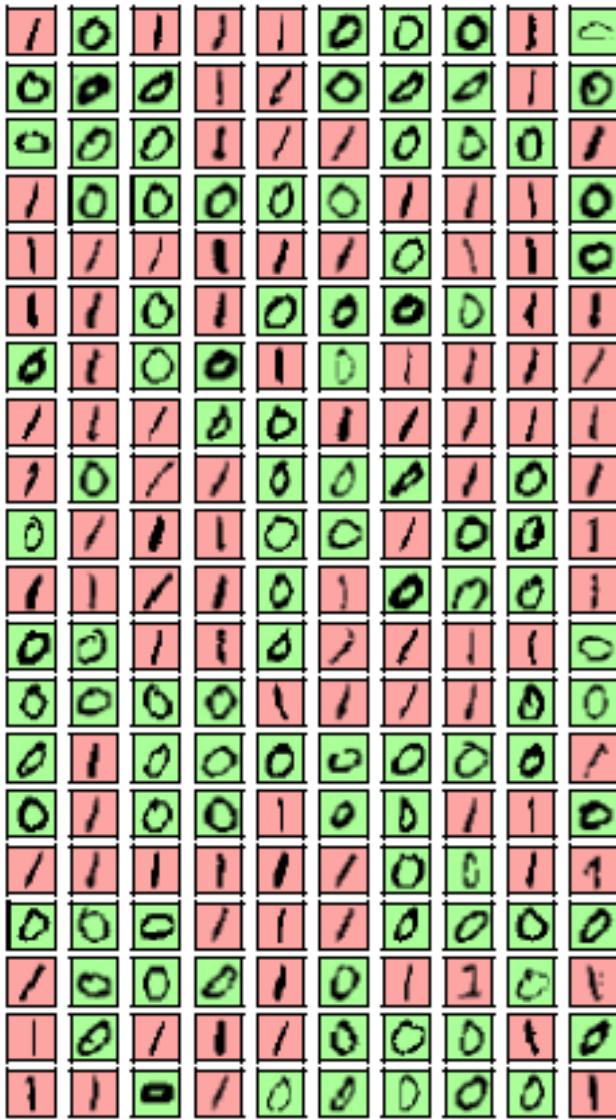
**784 de intrări**



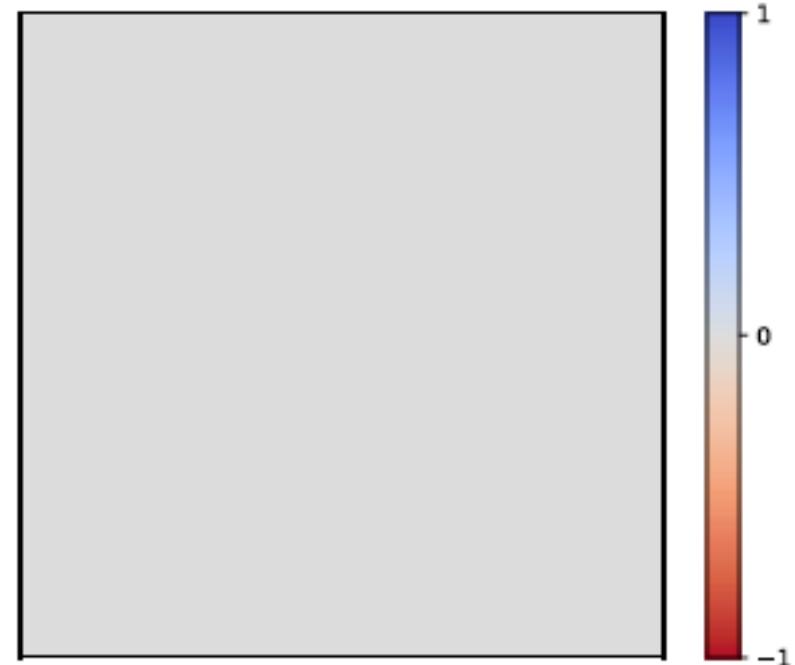
# Exemplu – Recunoașterea cifrelor

1	0	1	1	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1	0	
1	1	1	1	1	0	1	1	0	
1	1	0	1	0	0	0	1	1	
0	1	0	0	1	0	1	1	1	
1	1	1	0	1	1	1	1	1	
1	0	1	1	0	0	0	1	0	
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	0	1
0	0	1	1	0	1	1	1	0	0
0	0	0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	1	1
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	0	1	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	1	1

# Exemplu – Recunoașterea cifrelor

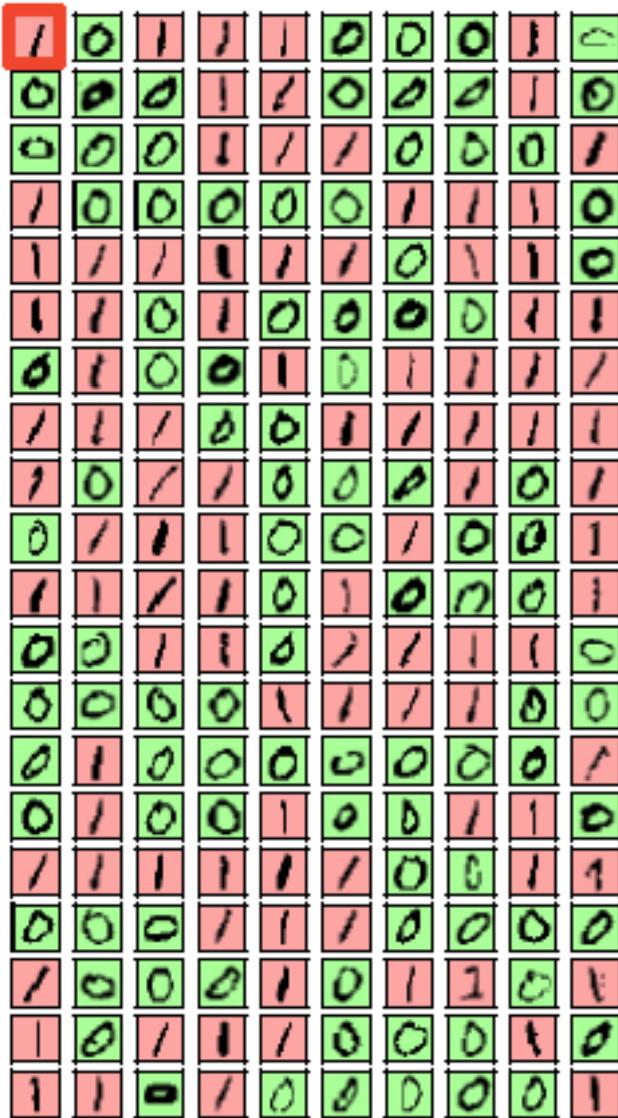


Valorile ponderilor (inițial 0)  
afișate ca o imagine  $28 \times 28$

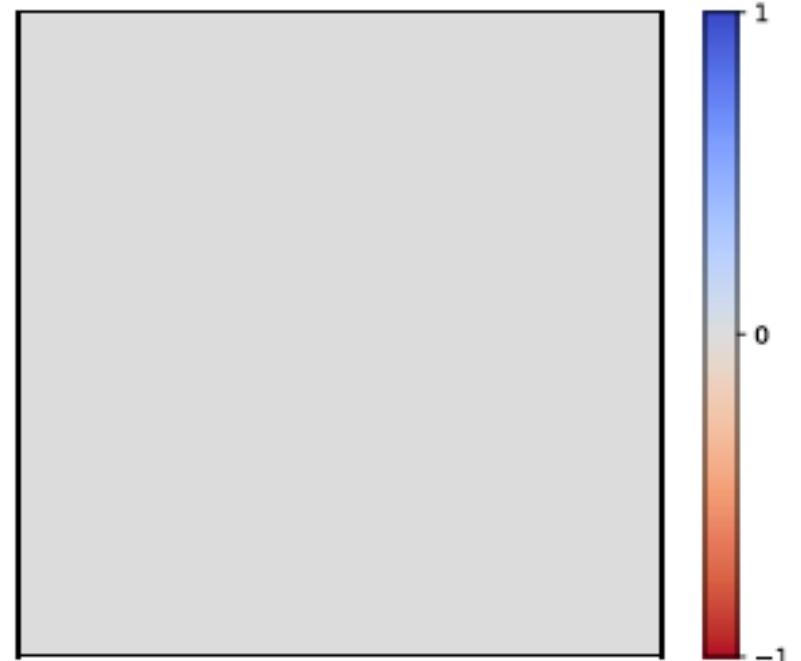


Pasul 0 de antrenare  
Acuratețe 50%

# Exemplu – Recunoașterea cifrelor



Valorile ponderilor afișate  
ca o imagine  $28 \times 28$



Pasul 1 de antrenare  
Acuratețe 50%

# Exemplu – Recunoașterea cifrelor

1	0	1	1	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1	0
1	1	1	1	1	1	0	1	1	0
1	1	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	0	1
0	0	1	1	0	1	1	1	1	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	1	1
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	0	1	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	0	1

## Regula de actualizare a perceptronului

$$w_j = w_j + \Delta w_j, \text{ unde}$$

$$\Delta w_j = \eta(y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

Cantitatea prin care schimbăm ponderea (rata de învățare)

Setează semnul actualizării

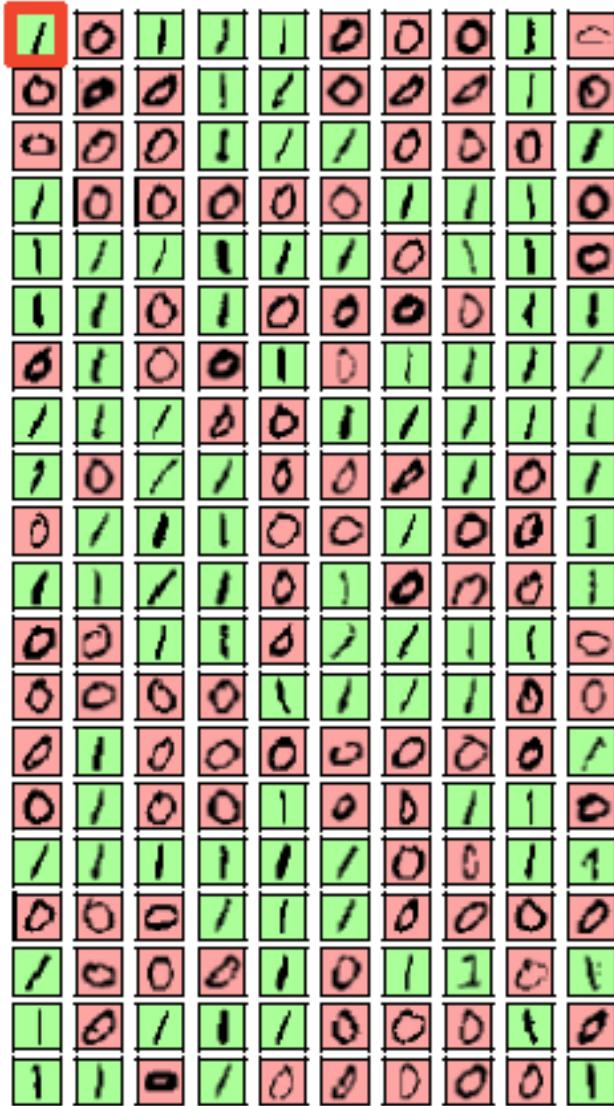
Selectează intrările active

Pentru exemple de antrenare misclasificate:

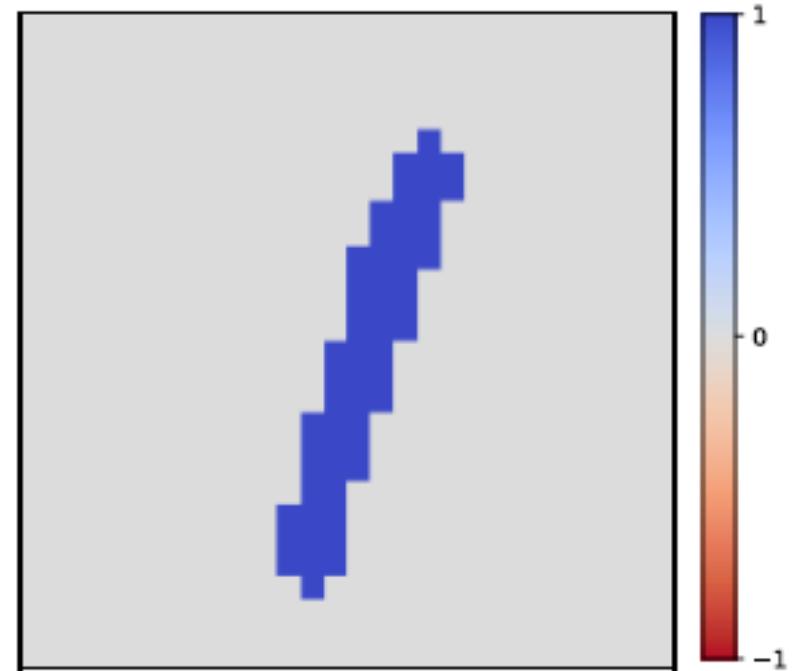
-daca  $y^{(i)}=1$  atunci adaug  $\eta * (1-0)*1$  pentru ponderile corespunzătoare pixelilor negri (cei din exemplul de antrenare curent cu cifra 1)

-daca  $y^{(i)}=0$  atunci adaug  $\eta * (0-1)*1$  pentru ponderile corespunzătoare pixelilor negri (cei din exemplul de antrenare curent cu cifra 0)

# Exemplu – Recunoașterea cifrelor



Valorile ponderilor afișate  
ca o imagine  $28 \times 28$

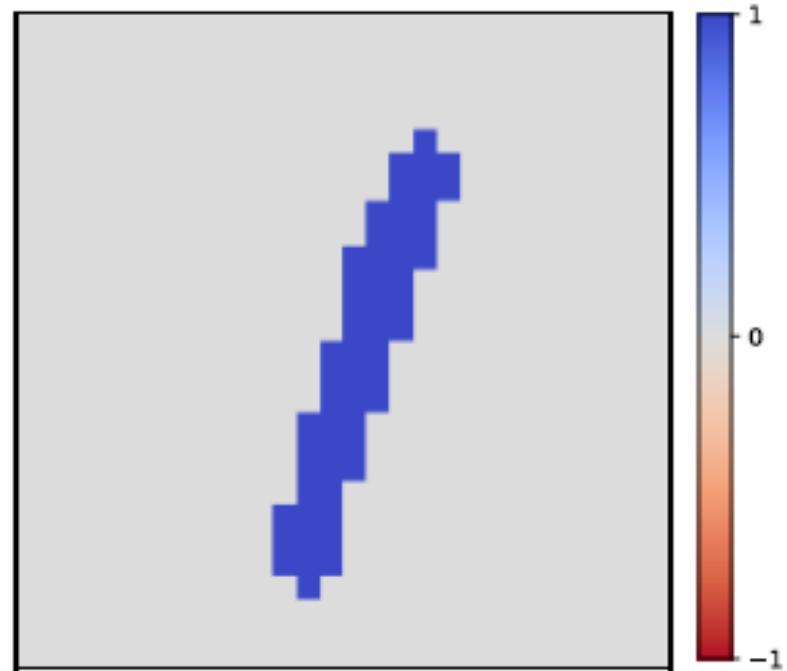


Pasul 1 de antrenare  
Acuratețe 50%

# Exemplu – Recunoașterea cifrelor

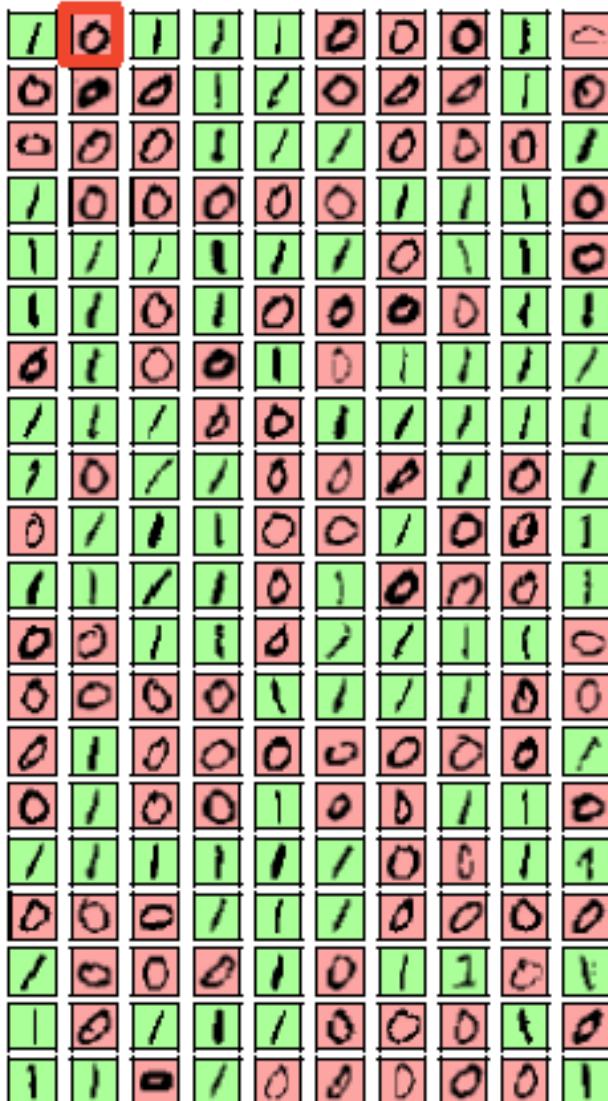
1	0	1	1	1	0	0	0	1	-1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1	0
1	1	1	1	1	1	0	1	1	0
1	1	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	0	1
0	0	1	1	0	1	1	1	0	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	1	1
0	0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	0	1	1	0	0	0	1	0	0
1	1	0	1	0	0	0	0	0	1

Valorile ponderilor afișate ca o imagine  $28 \times 28$



Pasul 2 de antrenare  
Acuratețe 50%

# Exemplu – Recunoașterea cifrelor



## Regula de actualizare a perceptronului

$$w_j = w_j + \Delta w_j, \text{ unde}$$

$$\Delta w_j = \eta(y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)}$$

Cantitatea prin care  
schimbăm ponderea  
(rata de învățare)

Setează semnul actualizării

Selectează intrările active

Pentru exemple de antrenare misclasificate:

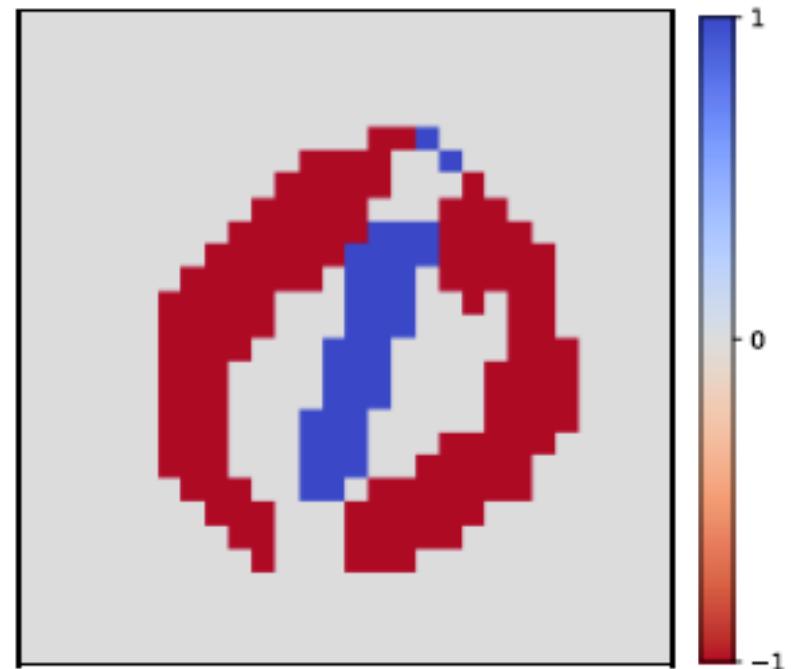
-daca  $y^{(i)}=1$  atunci adaug  $\eta*(1-0)*1$  pentru ponderile corespunzatoare pixelilor negri (cei din exemplul de antrenare curent cu cifra 1)

-daca  $y^{(i)}=0$  atunci adaug  $\eta*(0-1)*1$  pentru ponderile corespunzatoare pixelilor negri (cei din exemplul de antrenare curent cu cifra 0)

# Exemplu – Recunoașterea cifrelor

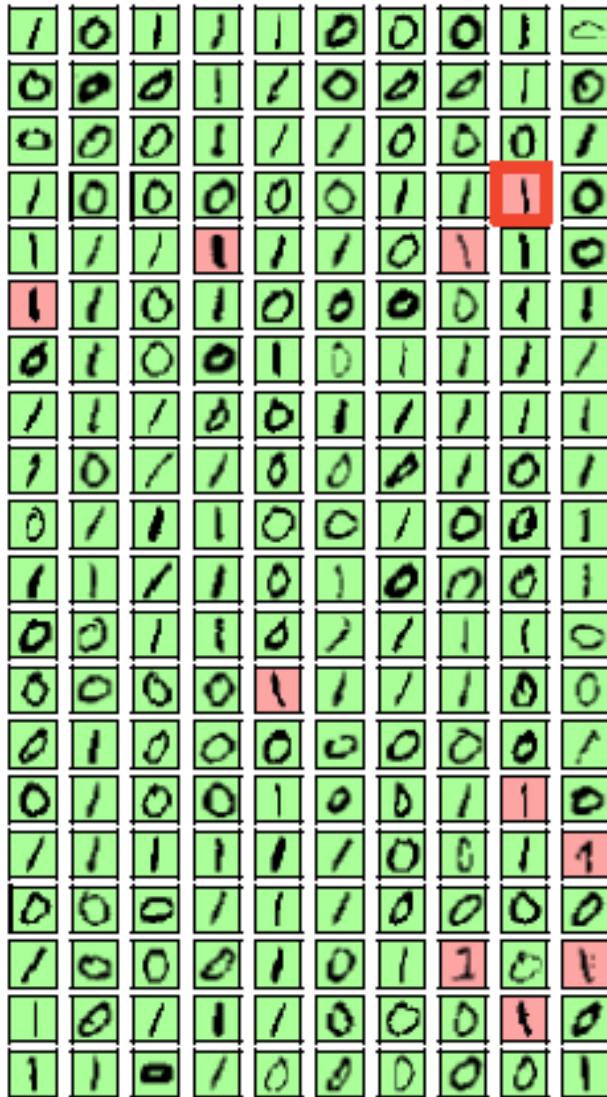
1	0	1	1	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1	0
1	1	1	1	1	0	1	1	0	0
1	1	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	1	0
0	0	1	1	0	1	1	1	1	0
0	0	0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	1	0
0	1	0	0	1	0	0	1	1	0
1	1	1	1	1	0	0	1	1	0
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	0	1	1	1	0	0	1	0	0
1	1	0	1	0	0	0	0	0	1

Valorile ponderilor afișate ca o imagine  $28 \times 28$

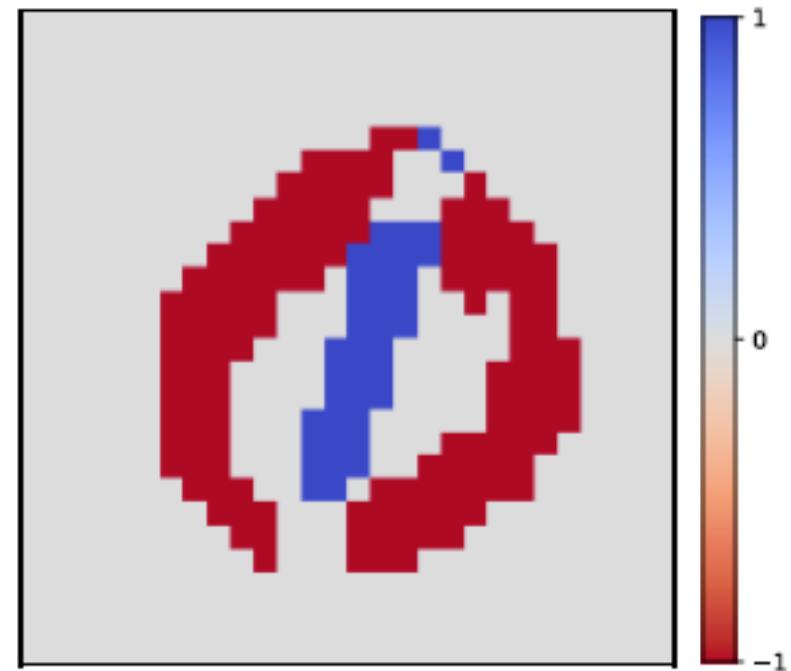


Pasul 2 de antrenare  
Acuratețe 95%

# Exemplu – Recunoașterea cifrelor

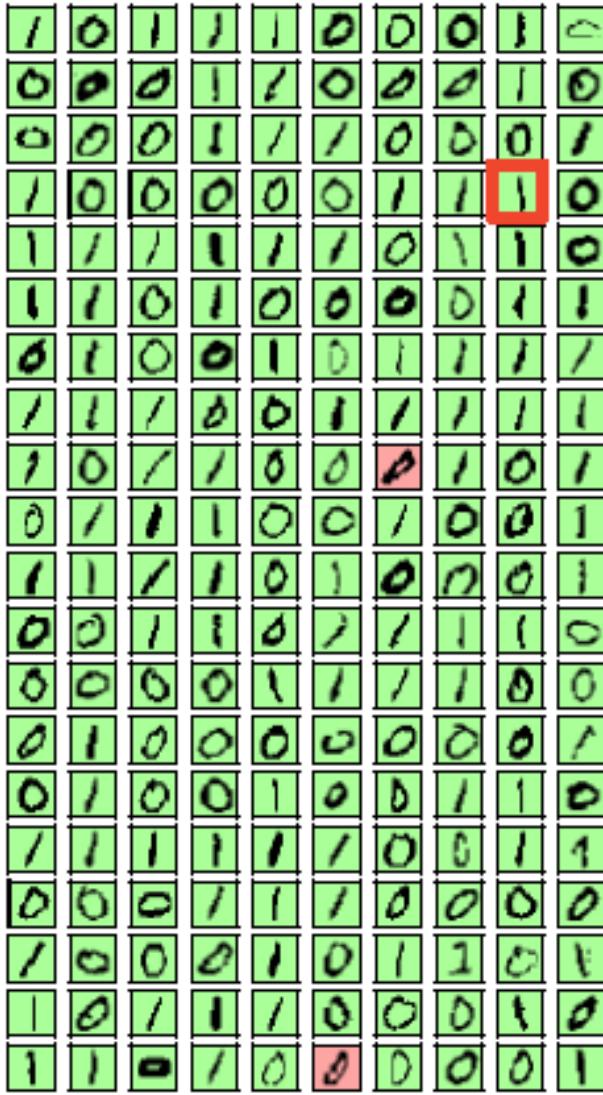


Valorile ponderilor afișate ca o imagine  $28 \times 28$

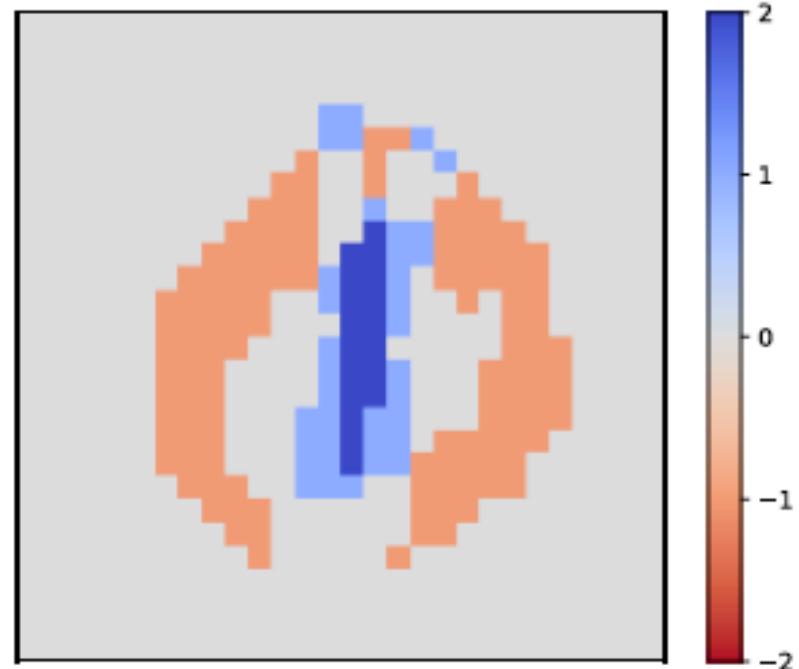


Pasul 39 de antrenare  
Acuratețe 95%

# Exemplu – Recunoașterea cifrelor

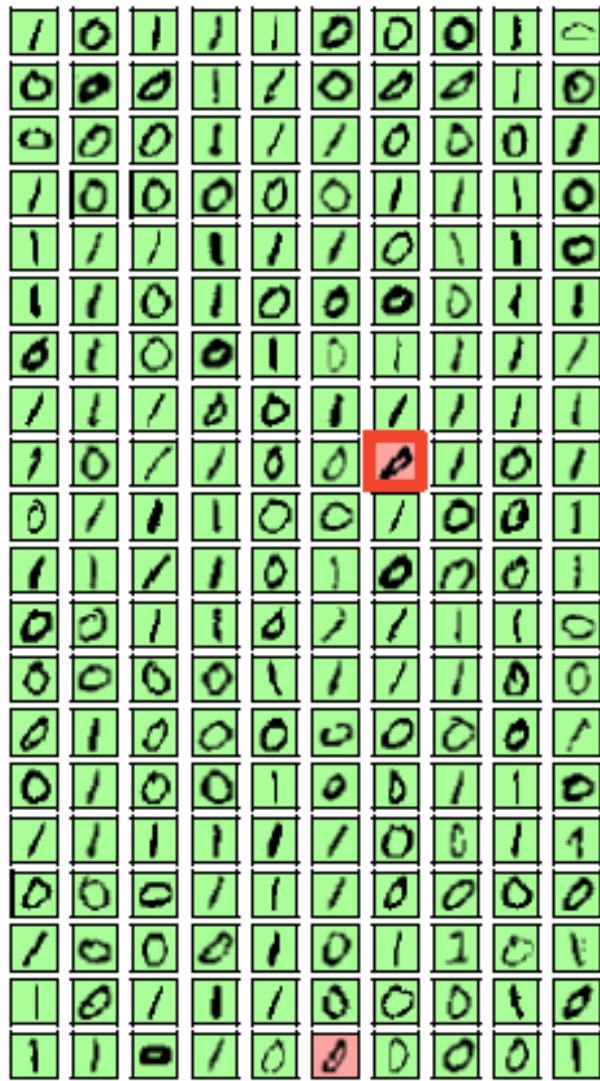


Valorile ponderilor afișate ca o imagine  $28 \times 28$

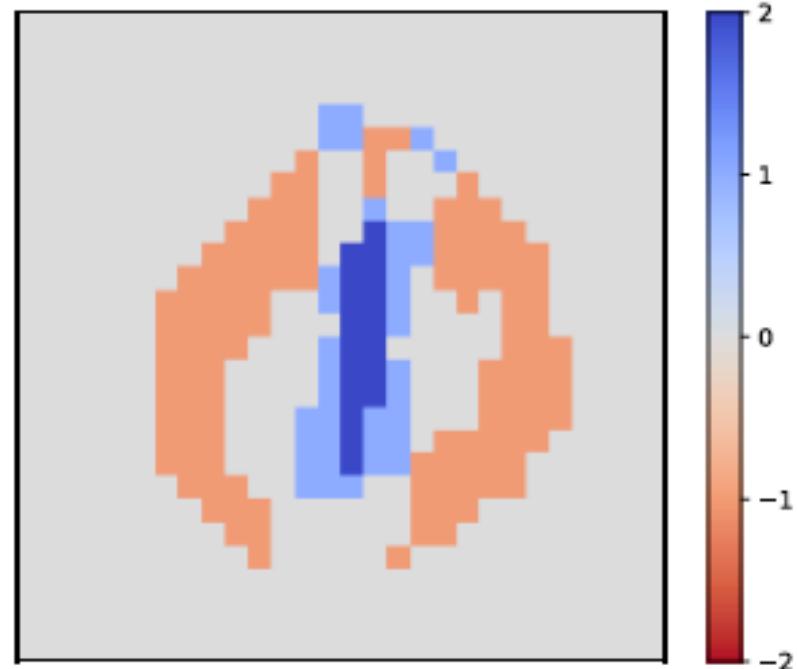


Pasul 39 de antrenare  
Acuratețe 99%

# Exemplu – Recunoașterea cifrelor



Valorile ponderilor afișate ca o imagine  $28 \times 28$

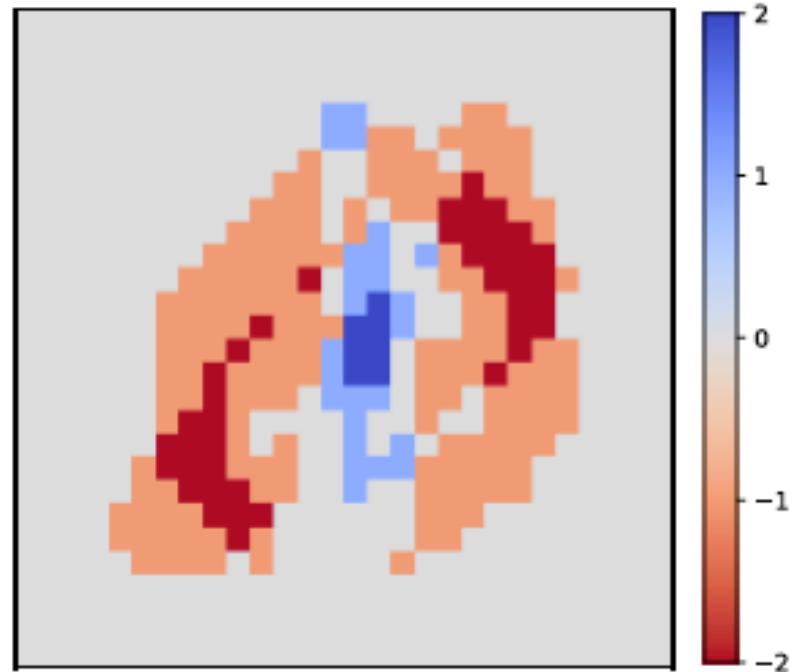


Pasul 87 de antrenare  
Acuratețe 99%

# Exemplu – Recunoașterea cifrelor

1	0	1	1	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1	1	0
1	1	1	1	1	0	1	1	0	0
1	1	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1	1
1	0	1	1	0	0	1	0	1	1
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	0	1
0	0	1	1	0	1	1	1	0	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	0	1	1	1	0
1	1	1	1	1	1	0	0	1	1
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	1	0	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	0	1

Valorile ponderilor afișate ca o imagine  $28 \times 28$

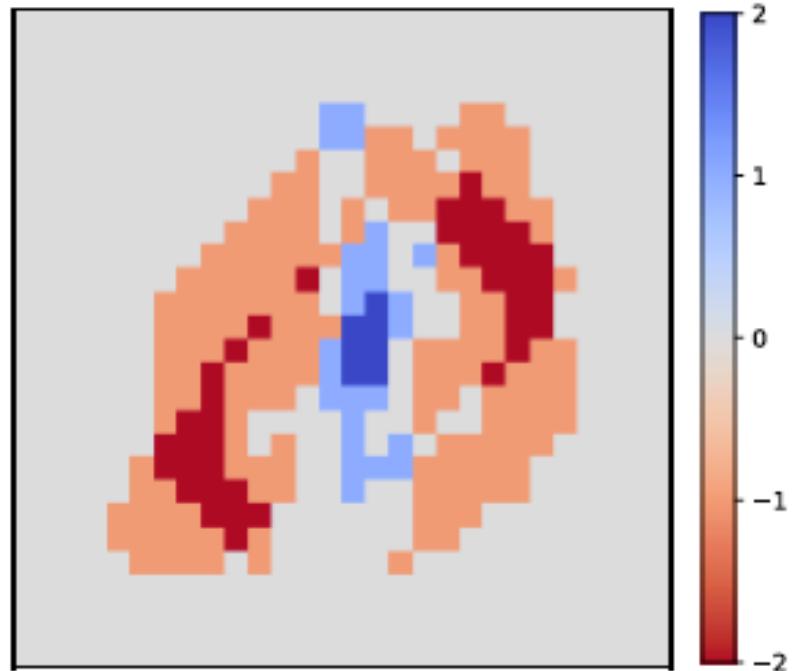


Pasul 87 de antrenare  
Acuratețe 88%

# Exemplu – Recunoașterea cifrelor

1	0	1	1	1	0	0	0	1	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1	0
1	1	1	1	1	0	1	1	0	1
1	1	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	0	1
0	0	1	1	0	1	1	1	0	0
0	0	0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	1	1
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	0	1	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	0	1

Valorile ponderilor afișate ca o imagine  $28 \times 28$

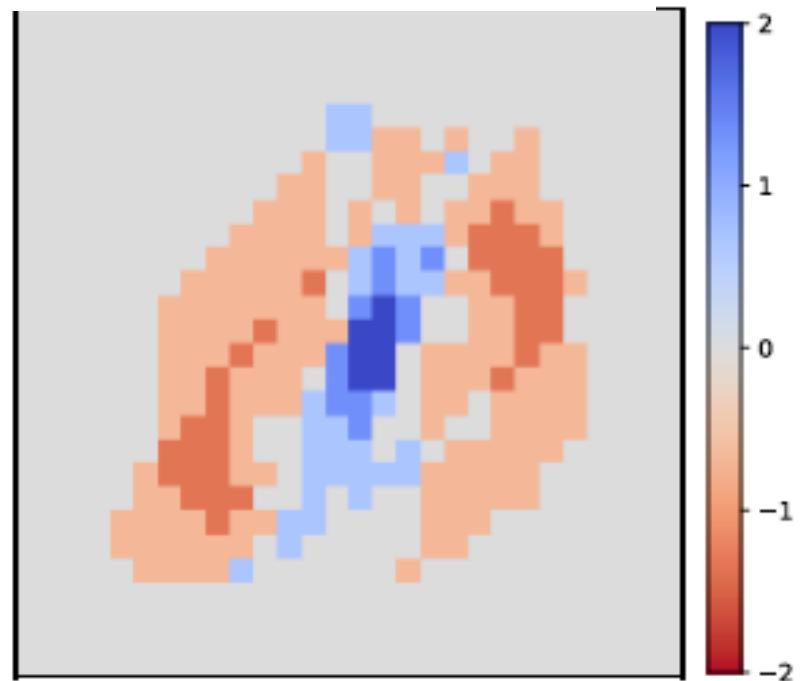


Pasul 92 de antrenare  
Acuratețe 88%

# Exemplu – Recunoașterea cifrelor

1	0	1	1	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1	0
1	1	1	1	1	1	0	1	1	0
1	1	0	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	1	1
1	0	1	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	0	1
1	1	1	1	0	1	0	0	0	1
0	0	1	1	0	1	1	1	1	0
0	0	0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	1	1
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1	0	1
1	0	1	1	1	0	0	0	1	0
1	1	0	1	0	0	0	0	0	1

Valorile ponderilor afișate ca o imagine  $28 \times 28$

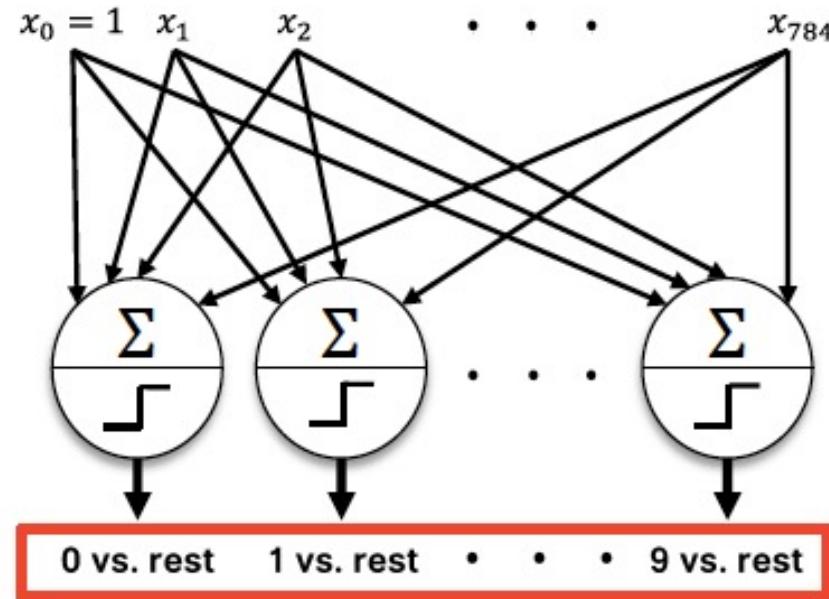


Pasul 92 de antrenare  
Acuratețe 100%

# Exemplu – Recunoașterea cifrelor

100 samples of each digit

8	0	3	6	1	0	0	4	1	7
3	2	4	9	5	1	1	9	2	8
1	5	8	1	2	0	5	8	3	0
3	5	4	0	6	9	7	5	5	2
2	7	0	0	8	8	6	1	8	4
9	8	6	1	3	3	3	2	6	4
6	5	7	7	8	7	3	3	6	5
4	6	7	7	4	4	6	2	3	6
5	5	5	4	3	1	5	5	1	2
2	3	9	7	5	0	6	2	4	2
4	6	7	4	5	0	8	7	3	2
0	2	9	4	7	1	6	3	2	1
6	9	8	7	9	2	6	3	8	8
3	4	4	9	6	5	1	9	3	7
5	8	4	0	5	3	8	3	1	1
8	2	0	3	5	5	6	7	4	1
3	7	6	1	7	9	6	2	5	1
8	6	1	7	9	5	2	2	5	4
7	9	5	0	3	1	4	5	2	4
5	2	4	8	8	5	8	✓	8	1



reprezentare one-hot  
(o componentă 1, restul 0)

O retea poate prezice clase multiple folosind un neuron separat pentru fiecare clasă  
Pe problema dată atinge ușor acuratețe de ~ 99%

# Perceptronul în IA

Extrase din ziarul The New York Times, 8 iulie, 1958

## NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

- <http://jcblackmon.com/wp-content/uploads/2018/01/MBC-Rosenblatt-Perceptron-NYT-article.jpg.pdf>

# Limitările perceptronului

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0$$

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0 \Rightarrow w_0 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_2$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_1$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0 \Rightarrow w_0 < -w_1 - w_2$$

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0 \Rightarrow w_0 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_2$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_1$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0 \Rightarrow w_0 < -w_1 - w_2$$

$$\left. \begin{aligned} 2w_0 &\geq -w_1 - w_2 \\ 2w_0 &> w_0 \end{aligned} \right\} \Rightarrow w_0 > 0$$

# Limitările perceptronului

- Perceptronul învață funcțiile booleene AND/OR și reușește să rezolve probleme de recunoaștere a cifrelor. Probleme simple sau grele?
- Considerăm funcția XOR (exclusiv OR):

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Pentru ca un Perceptron să învețe funcția XOR e nevoie ca:

$$w_0 + 0 \cdot w_1 + 0 \cdot w_2 < 0 \Rightarrow w_0 < 0$$

$$w_0 + 0 \cdot w_1 + 1 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_2$$

$$w_0 + 1 \cdot w_1 + 0 \cdot w_2 \geq 0 \Rightarrow w_0 \geq -w_1$$

$$w_0 + 1 \cdot w_1 + 1 \cdot w_2 < 0 \Rightarrow w_0 < -w_1 - w_2$$

$$\left. \begin{array}{l} 2w_0 \geq -w_1 - w_2 \\ 2w_0 > w_0 \end{array} \right\} \Rightarrow 2w_0 > w_0 \Rightarrow w_0 > 0$$



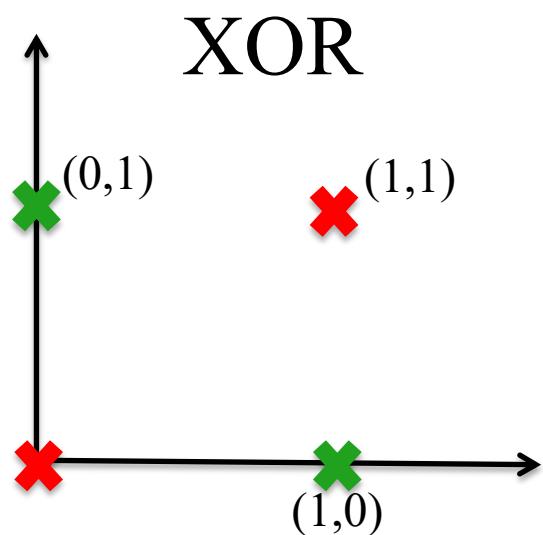
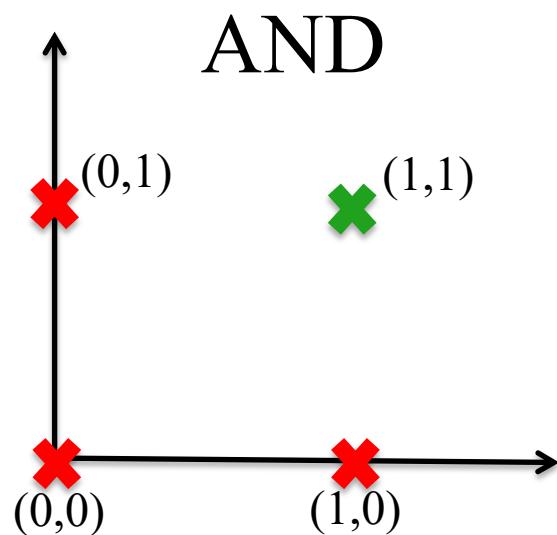
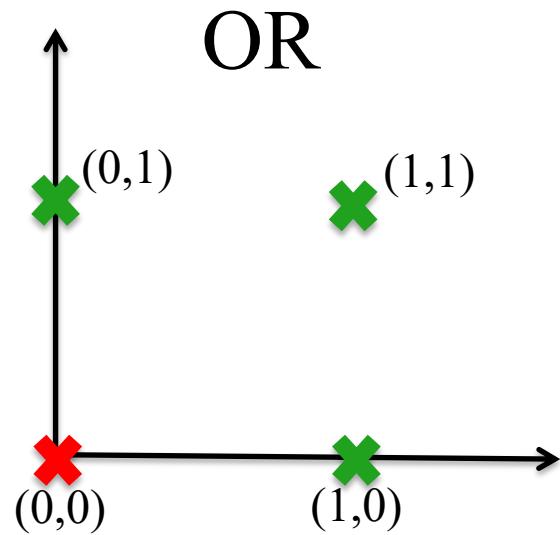
**Contradicție**

- Perceptronul nu poate învăța XOR oricât de mulți pași ar face
- Mai mult, Perceptronul poate învăță numai clase care sunt liniar separabile

# Teorema de convergență a perceptronului

- Dacă mulțimea de antrenare  $E$  este liniar separabilă cu margine  $\gamma$ , algoritmul de învățare a perceptronului este garantat că va converge într-un număr finit de pași către o soluție în care nu se fac greșeli pe mulțimea de antrenare
- Numărul de pași  $k$  satisfac relația  $k \leq \frac{R^2}{\gamma^2}$ , unde  $R$  este raza sferei din spațiul caracteristicilor care cuprinde toate exemplele de antrenare
- Teorema spune că va converge către o soluție, nu este necesar să fie o soluție bună (SVM oferă soluția de margine maximă). Dacă mulțimea de antrenare  $E$  nu este liniar separabilă algoritmul nu va converge către o soluție.

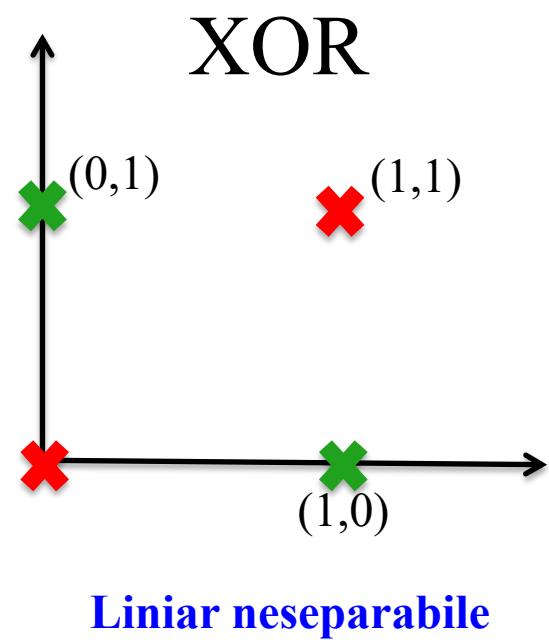
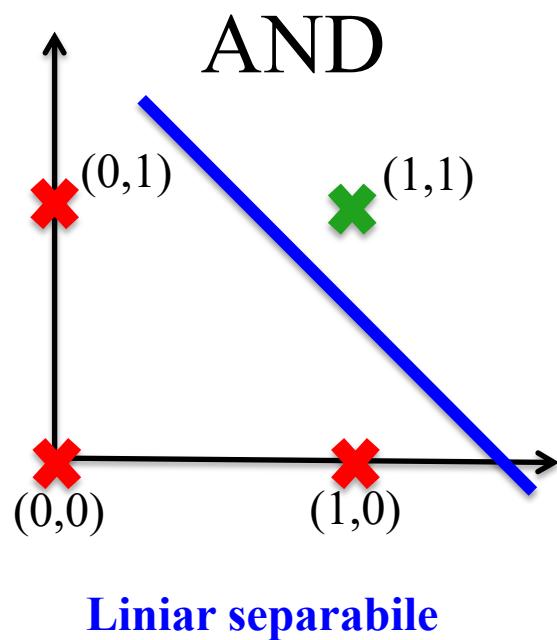
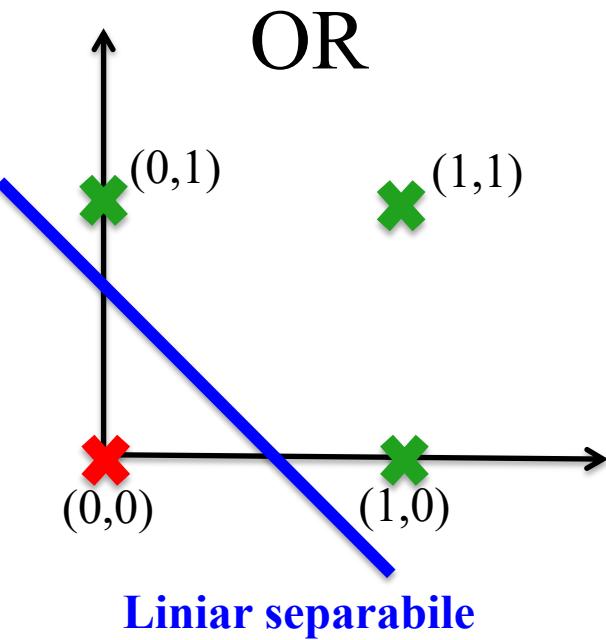
# Perspectiva geometrică



✗ Eticheta 0

✖ Eticheta 1

# Perspectiva geometrică

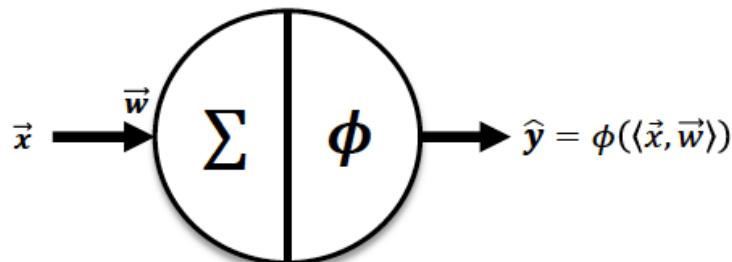
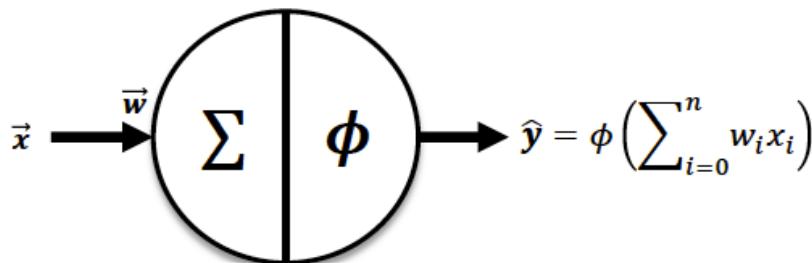
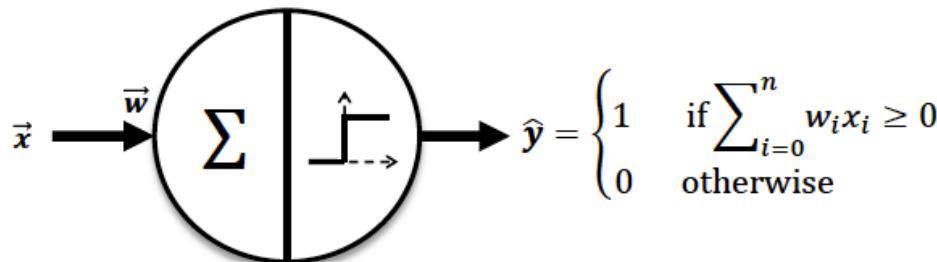
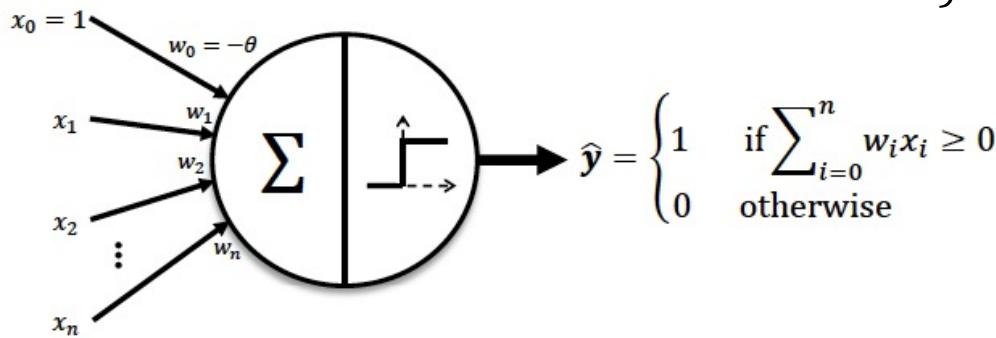


✗ Eticheta 0

✖ Eticheta 1

# Alte reguli de învățare pentru perceptron

# Notății



Intrările  $x_i$  și ponderile  $w_i$   
reprezentate sub formă de vectori

Funcția hardlim notată cu  $\Phi$  și  
numită funcție de activare

Suma intrărilor ponderate  
reprezentată ca produs scalar

# Regula de învățare delta

- Ieșirea perceptronului este:  $\hat{y} = \phi(\langle \vec{x}, \vec{w} \rangle)$
- Măsur diferența dintre ce voiam să obțin și ceea ce furnizează perceptronul la ieșire folosind o funcție de eroare E
- Există multe posibilități de alegere a funcției de eroare, spre exemplu pot alege E ca fiind eroarea pătratică:

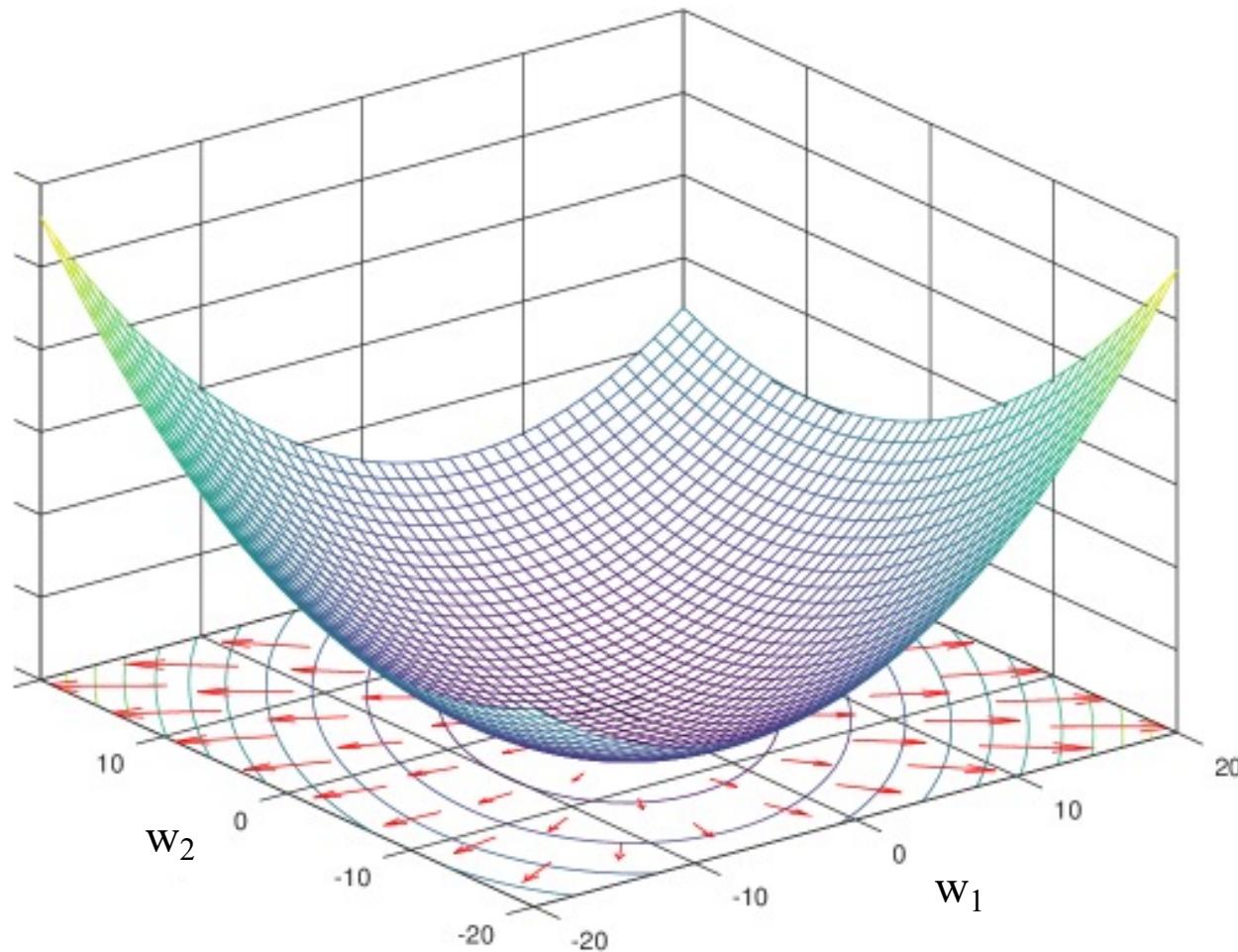
$$E(\vec{x}) = \frac{1}{2} (\vec{y} - \hat{\vec{y}})^2$$

Funcția de eroare E ia valori mici când cele două etichete iau valori apropiate și ia valori mari altfel

- Pot folosi metoda coborârii pe gradient (gradient descent – Cauchy 1847) pentru găsirea minimului unei funcții prin actualizarea setului de ponderi actual în direcția inversă a gradientului

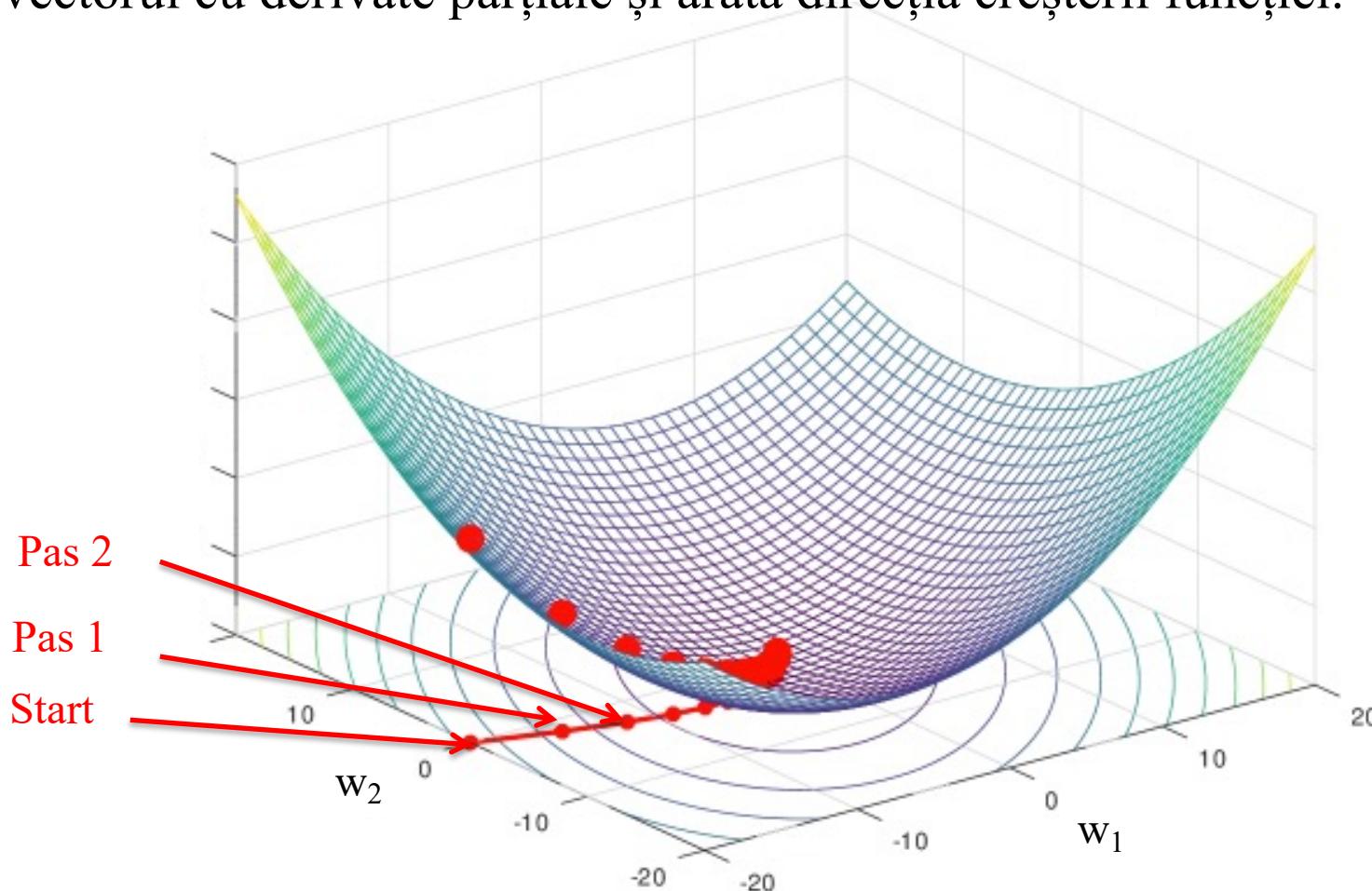
# Suprafață descrisă de funcția de eroare E

Functia de eroare pătratică E descrie un paraboloid în  $R^n$



# Algoritmul de coborâre pe gradient

Algoritm iterativ, la fiecare pas o ia în direcția inversă a gradientului pentru minimizarea valorii funcției E. Gradientul unei funcții într-un punct este vectorul cu derivate parțiale și arată direcția creșterii funcției.



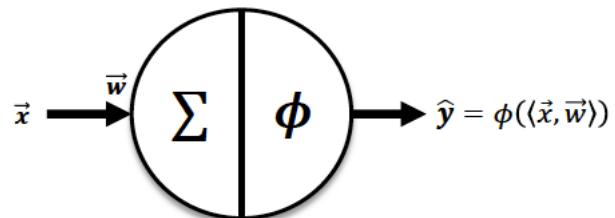
# Regula de învățare delta

- Ieșirea perceptronului este:  $\hat{y} = \phi(\langle \vec{x}, \vec{w} \rangle)$
- Aleg E ca fiind eroarea pătratică:  $E(\vec{x}) = \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})^2$
- Pot folosi metoda coborârii pe gradient (gradient descent – Cauchy 1847) pentru găsirea minimului funcției E prin actualizarea setului de ponderi actual  $\vec{w}$  în direcția inversă a gradientului

$$\frac{\partial E(\vec{x})}{\partial w_j} = \frac{\partial E(\vec{x})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_j} = \frac{\partial E(\vec{x})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \langle \vec{x}, \vec{w} \rangle} \cdot \frac{\partial \langle \vec{x}, \vec{w} \rangle}{\partial w_j}$$

$$\frac{\partial E(\vec{x})}{\partial \hat{y}} = -(y - \hat{y})$$

$$\frac{\partial \hat{y}}{\partial \langle \vec{x}, \vec{w} \rangle} = \phi'(\langle \vec{x}, \vec{w} \rangle)$$



$$\frac{\partial \langle \vec{x}, \vec{w} \rangle}{\partial w_j} = \frac{\partial (\sum_{j=0}^n w_j x_j)}{\partial w_j} = x_j$$

# Regula de învățare delta

- Ieșirea perceptronului este:  $\hat{y} = \phi(\langle \vec{x}, \vec{w} \rangle)$
- Aleg E ca fiind eroarea pătratică:  $E(\vec{x}) = \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})^2$
- Pot folosi metoda coborârii pe gradient (gradient descent – Cauchy 1847) pentru găsirea minimului funcției E prin actualizarea setului de ponderi actual  $\vec{w}$  în direcția inversă a gradientului

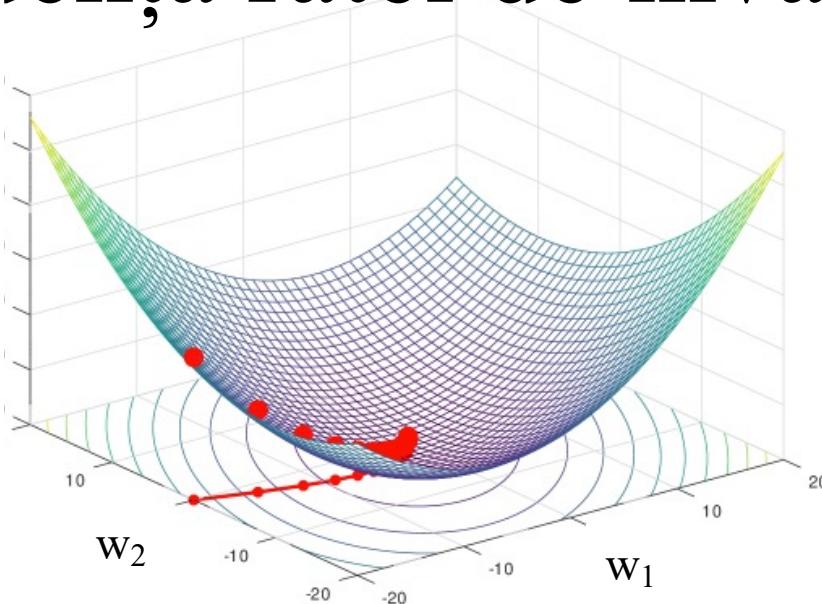
$$\frac{\partial E(\vec{x})}{\partial w_j} = \frac{\partial E(\vec{x})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_j} = \frac{\partial E(\vec{x})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \langle \vec{x}, \vec{w} \rangle} \cdot \frac{\partial \langle \vec{x}, \vec{w} \rangle}{\partial w_j} = -(y - \hat{y}) \cdot \phi'(\langle \vec{x}, \vec{w} \rangle) \cdot x_j$$

Regula de învățare delta

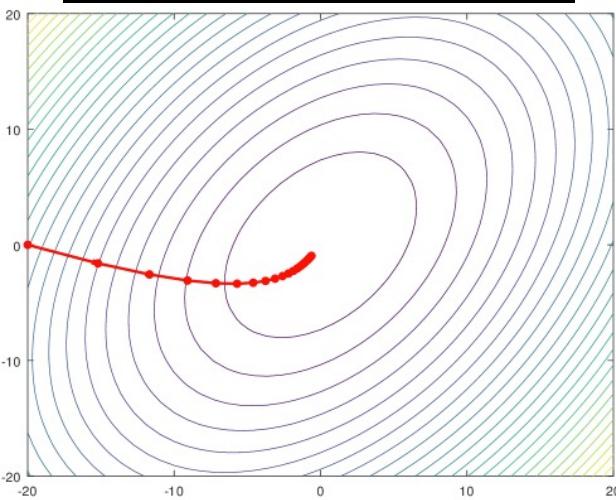
$$\Delta w_j = \eta \cdot (y - \hat{y}) \cdot \phi'(\langle \vec{x}, \vec{w} \rangle) \cdot x_j$$

↑  
rata de învățare

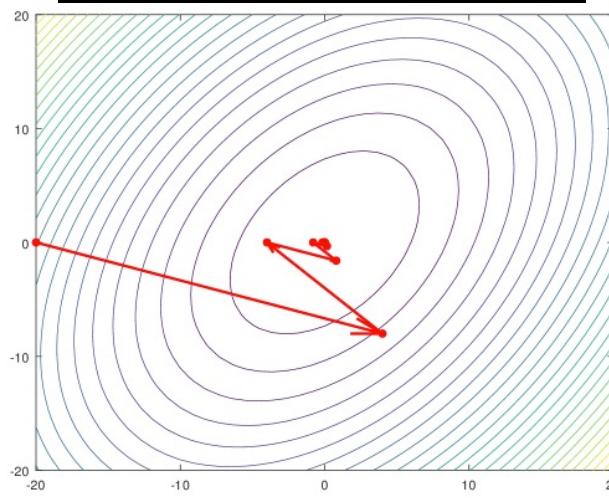
# Influența ratei de învățare $\eta$



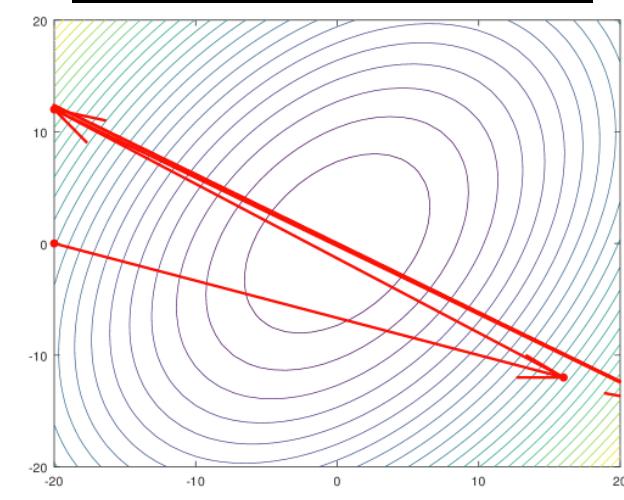
Rată de învățare prea mică  
(algoritmul converge încet)



Rată de învățare bună  
(viteză și convergență)



Rată de învățare mare  
(algoritm diverge)

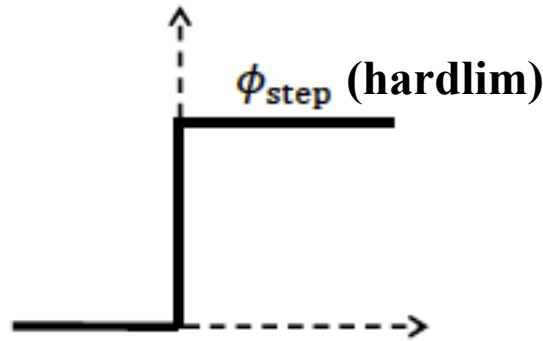


# Regula de învățare delta

- Regula de învățare delta:

$$w_j = w_j + \Delta w_j, \text{ unde}$$

$$\Delta w_j = \eta \cdot (y - \hat{y}) \cdot \phi'(\langle \vec{x}, \vec{w} \rangle) \cdot x_j$$



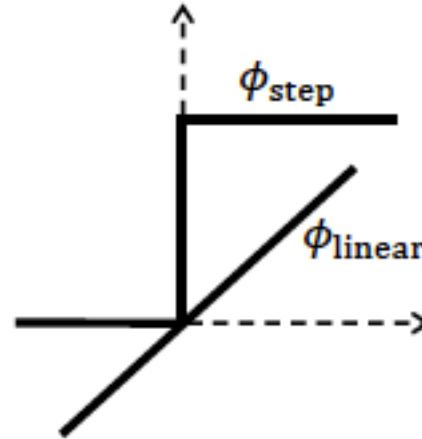
- Funcția  $\Phi_{\text{step}} = \text{hardlim}$  (step function) nu este derivabilă în punctul 0 și în rest are derivata 0 (nu se va face nicio actualizare)

# Regula de învățare adaline delta

- Regula de învățare delta:

$$w_j = w_j + \Delta w_j, \text{ unde}$$

$$\Delta w_j = \eta \cdot (y - \hat{y}) \cdot \phi'(\langle \vec{x}, \vec{w} \rangle) \cdot x_j$$



- ADALINE (ADaptive Linear NEuron) este o variantă a Perceptronului propusă de Bernard Widrow în 1960, care folosește o funcție liniară  $\Phi_{\text{linear}}(x) = x$  de activare la antrenare și una hardlim (step) la testare. Derivata  $\Phi_{\text{linear}}'(x) = 1$ .
- Obținem regula de învățare adaline delta:

$$w_j = w_j + \Delta w_j, \text{ unde } \Delta w_j = \eta \cdot (y - \hat{y}) \cdot x_j$$

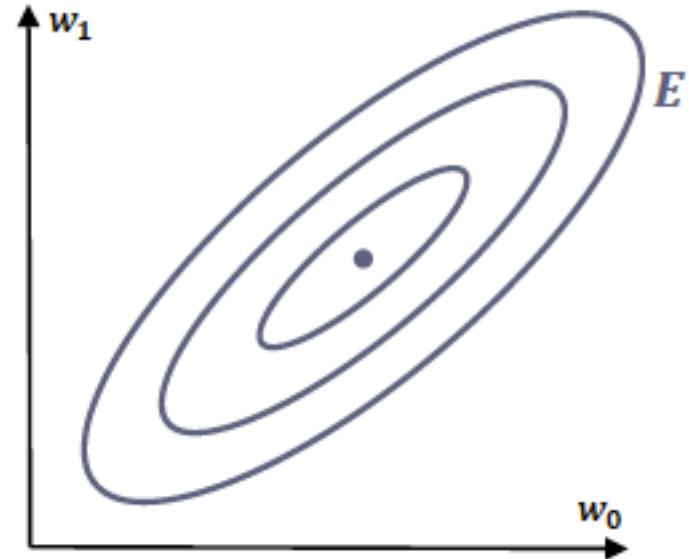
# Algoritmul de învățare al perceptronului

```
1 def perceptron(X, y, n_epochs, η):  
2     m, n = X.shape # number of samples, number of inputs  
3     for j in range(n):  
4         wj = 0  
5     for epoch in range(n_epochs): # an “epoch” is a run through all training data.  
6         for i in range(m): # a “training step” is one update of the weights.  
7             ŷ(i) = unit_step_function(Σj=0n wjxj(i))      — ŷ(i) = {  
8                 1   if Σj=0n wjxj(i) ≥ 0  
9                 0   otherwise  
          for j in range(n):  
              wj += η(y(i) - ŷ(i)) · xj(i)
```

Similar cu regula de  
învățare adaline delta  $w_j = w_j + \Delta w_j$ , unde  $\Delta w_j = \eta \cdot (y - \hat{y}) \cdot x_j$

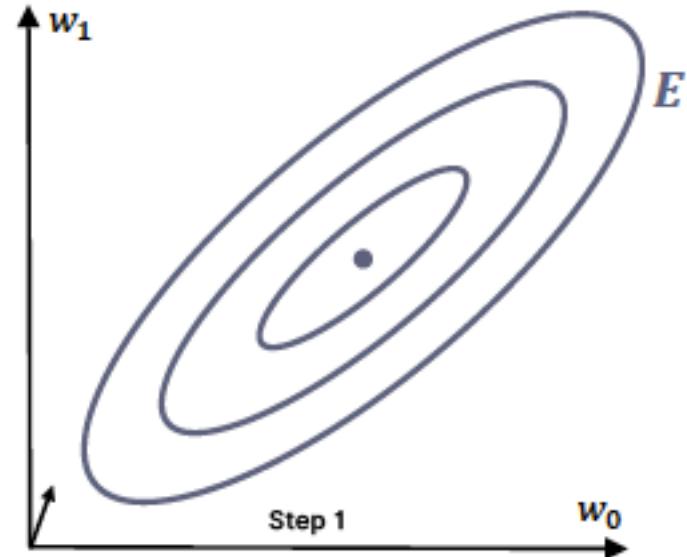
# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare



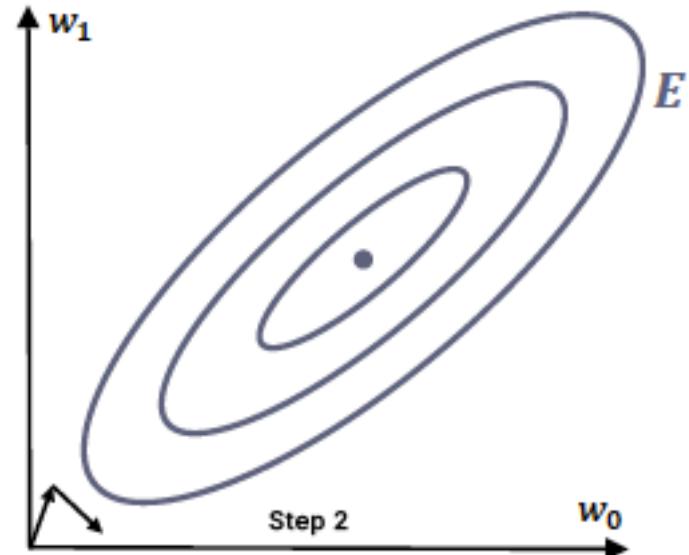
# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare



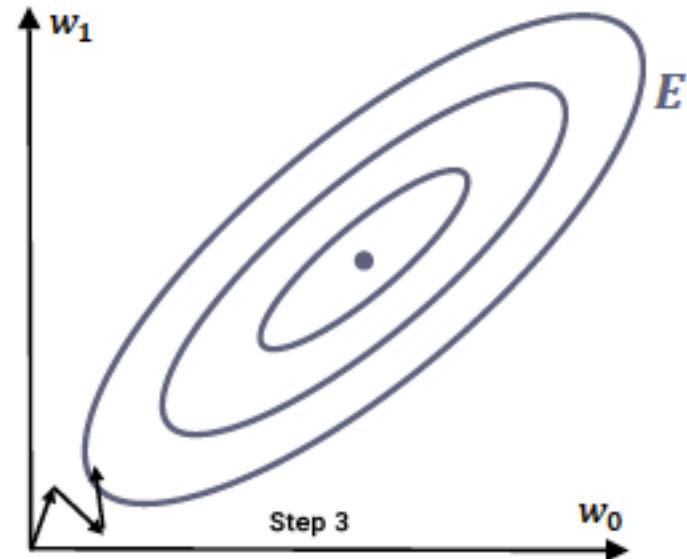
# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare



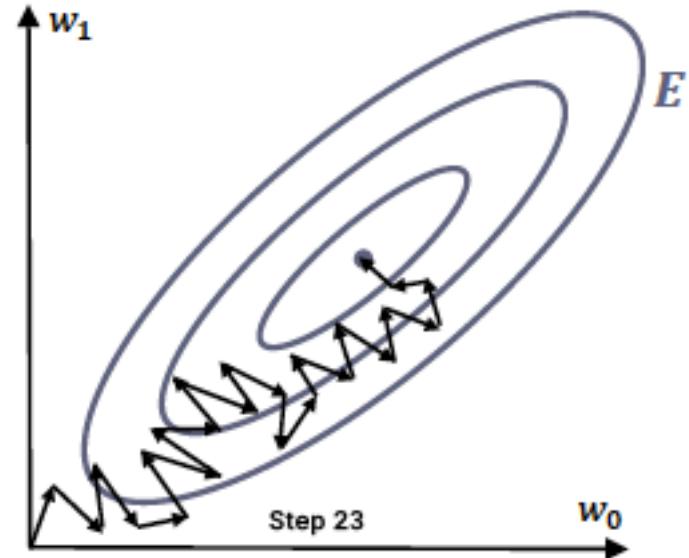
# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare



# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare

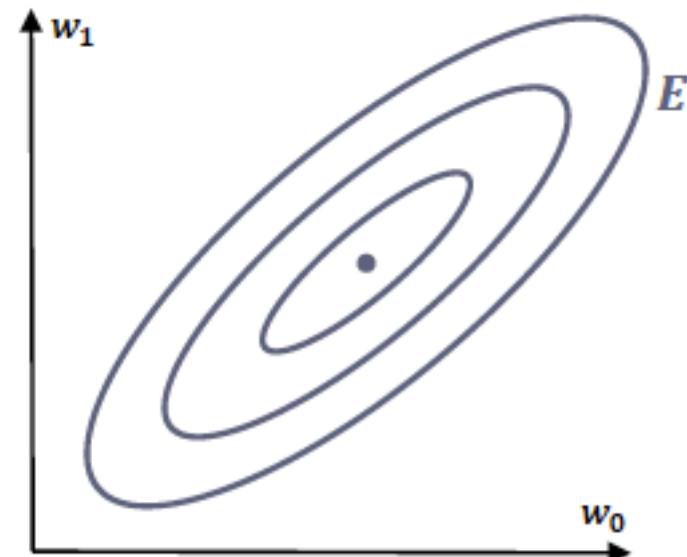
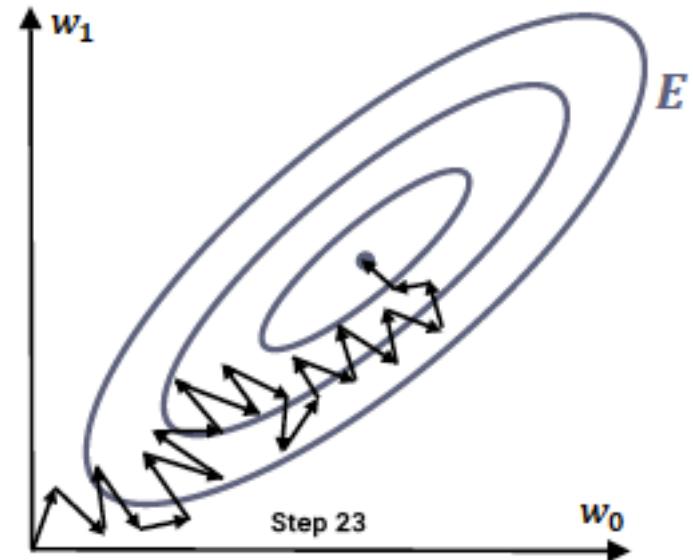


# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare
- Învățarea de tip batch înseamnă actualizarea ponderilor în funcție de mai multe exemple de antrenare pe baza mediei gradienților:

$$\Delta w_j = \frac{\eta}{b} \sum_{i_b=0}^{b-1} (y^{(i+i_b)} - \hat{y}^{(i+i_b)}) \cdot x_j^{(i+i_b)}$$

b – numărul de exemple în batch

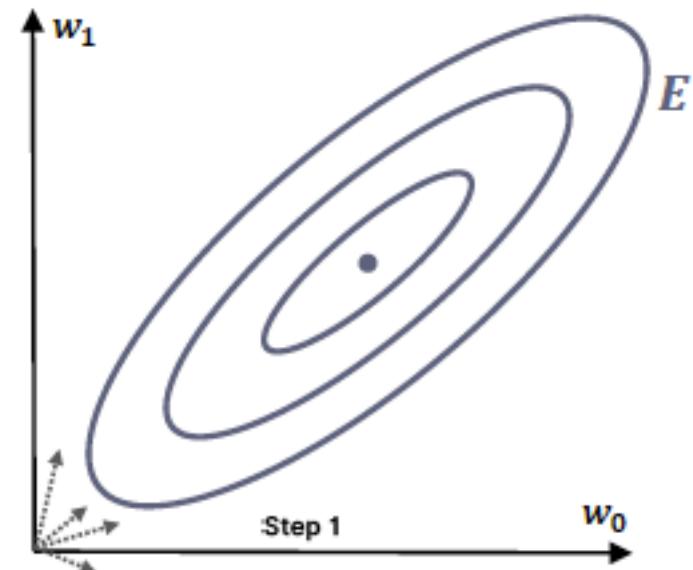
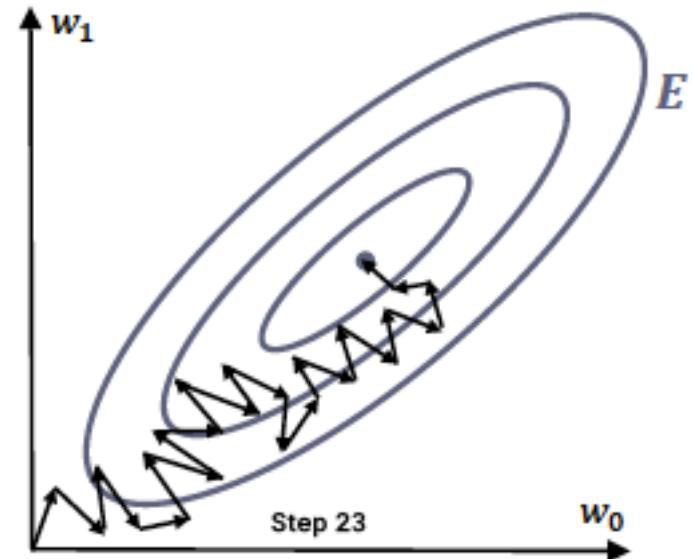


# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare
- Învățarea de tip batch înseamnă actualizarea ponderilor în funcție de mai multe exemple de antrenare pe baza mediei gradienților:

$$\Delta w_j = \frac{\eta}{b} \sum_{i_b=0}^{b-1} (y^{(i+i_b)} - \hat{y}^{(i+i_b)}) \cdot x_j^{(i+i_b)}$$

$b$  – numărul de exemple în batch

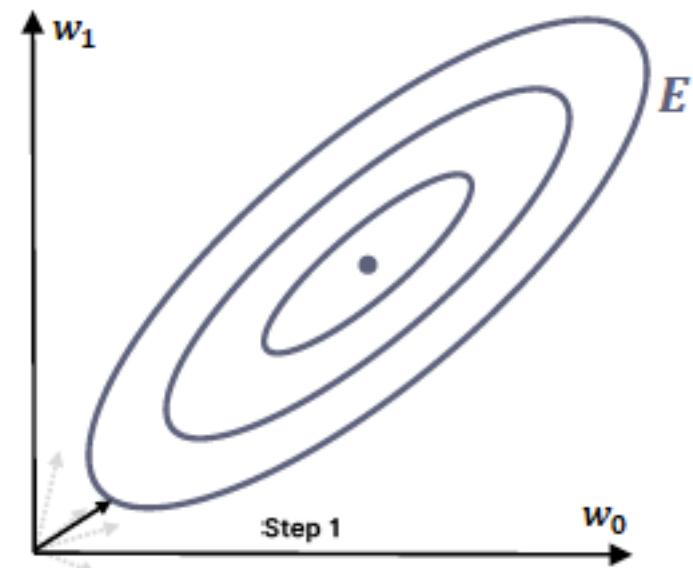
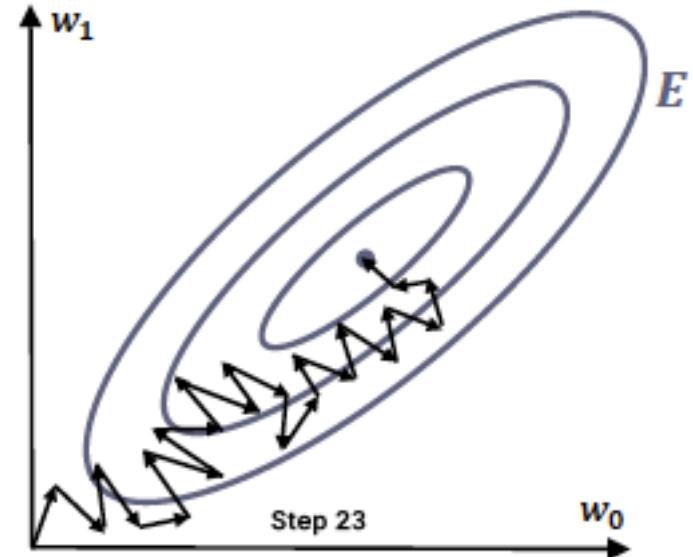


# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare
- Învățarea de tip batch înseamnă actualizarea ponderilor în funcție de mai multe exemple de antrenare pe baza mediei gradienților:

$$\Delta w_j = \frac{\eta}{b} \sum_{i_b=0}^{b-1} (y^{(i+i_b)} - \hat{y}^{(i+i_b)}) \cdot x_j^{(i+i_b)}$$

$b$  – numărul de exemple în batch

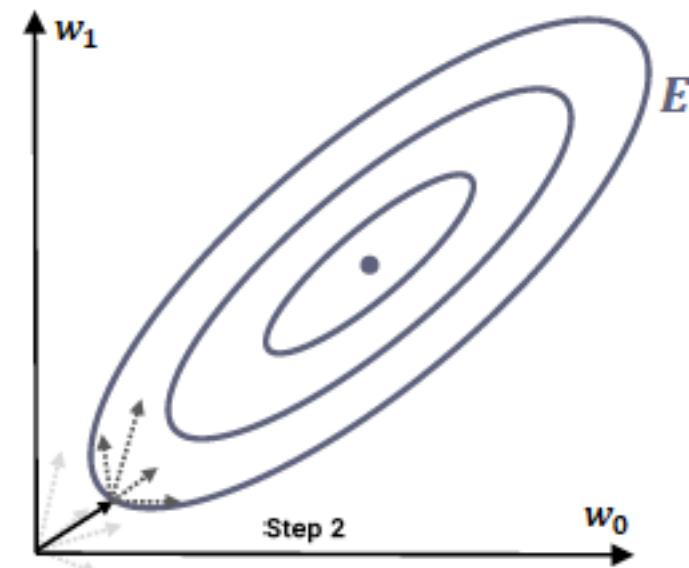
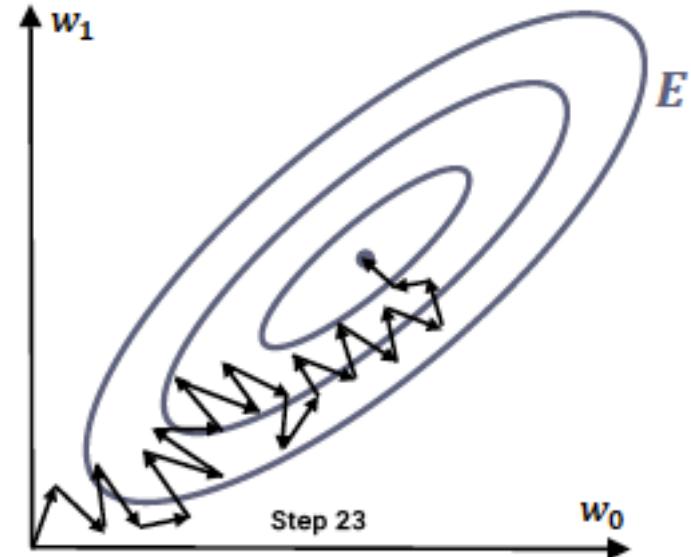


# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare
- Învățarea de tip batch înseamnă actualizarea ponderilor în funcție de mai multe exemple de antrenare pe baza mediei gradienților:

$$\Delta w_j = \frac{\eta}{b} \sum_{i_b=0}^{b-1} (\mathbf{y}^{(i+i_b)} - \hat{\mathbf{y}}^{(i+i_b)}) \cdot x_j^{(i+i_b)}$$

$b$  – numărul de exemple în batch

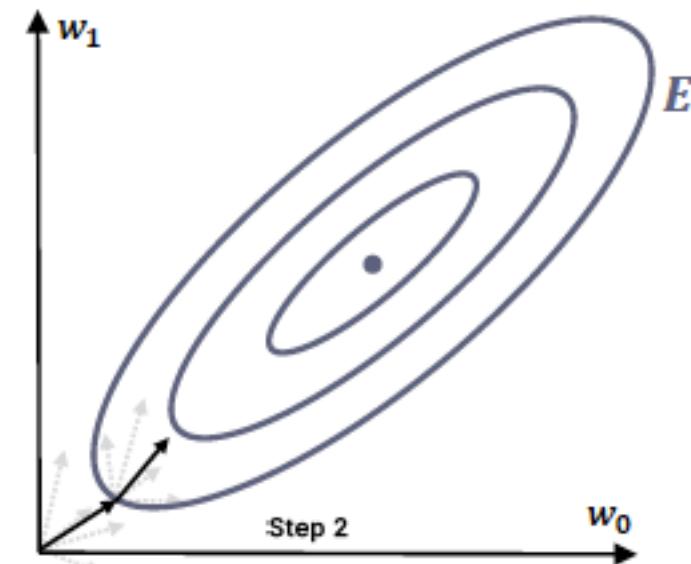
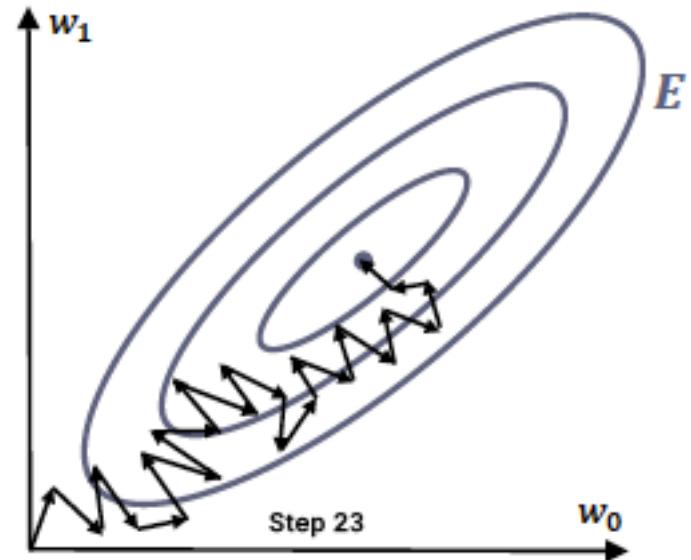


# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare
- Învățarea de tip batch înseamnă actualizarea ponderilor în funcție de mai multe exemple de antrenare pe baza mediei gradienților:

$$\Delta w_j = \frac{\eta}{b} \sum_{i_b=0}^{b-1} (y^{(i+i_b)} - \hat{y}^{(i+i_b)}) \cdot x_j^{(i+i_b)}$$

$b$  – numărul de exemple în batch

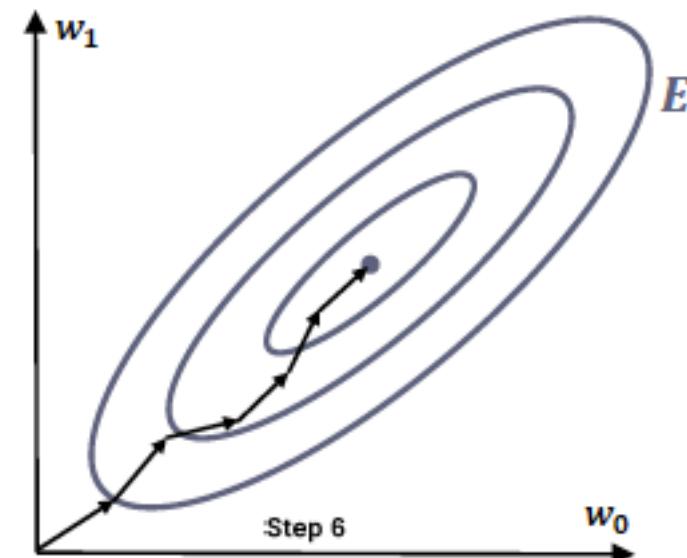
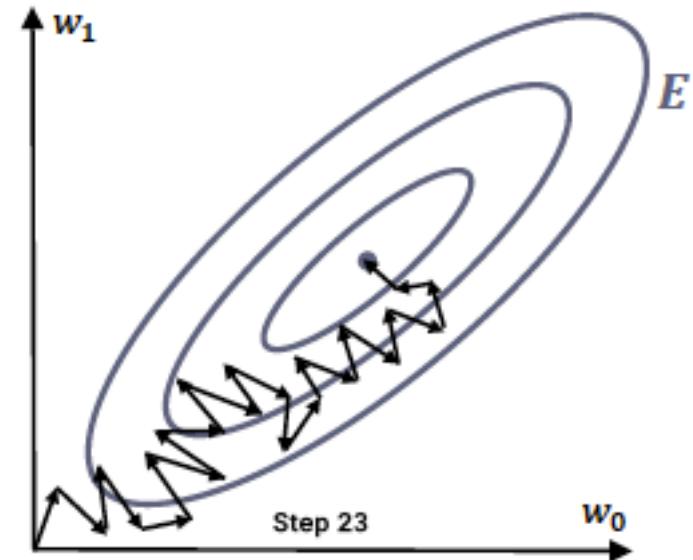


# Învățarea de tip batch

- Vrem să găsim un set de ponderi  $\vec{w}$  care să minimizeze eroarea pe toate exemplele de antrenare
- În algoritmii precedenți de învățare modific ponderile după fiecare exemplu misclasificat
- Actualizarea după fiecare exemplu misclasificat în direcția dată de exemplul curent s-ar putea să nu fie cea mai bună idee raportată la întreaga mulțime de antrenare
- Învățarea de tip batch înseamnă actualizarea ponderilor în funcție de mai multe exemple de antrenare pe baza mediei gradienților:

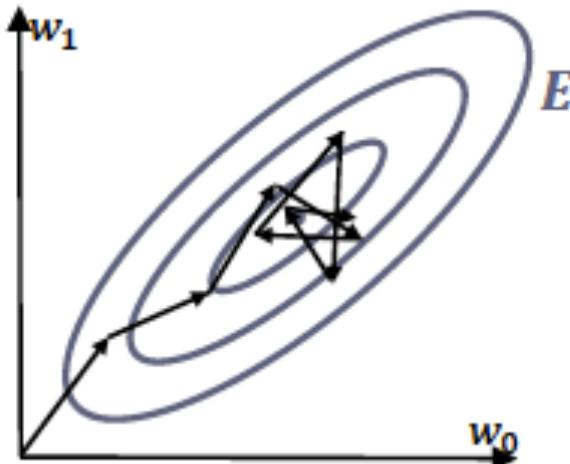
$$\Delta w_j = \frac{\eta}{b} \sum_{i_b=0}^{b-1} (y^{(i+i_b)} - \hat{y}^{(i+i_b)}) \cdot x_j^{(i+i_b)}$$

$b$  – numărul de exemple în batch

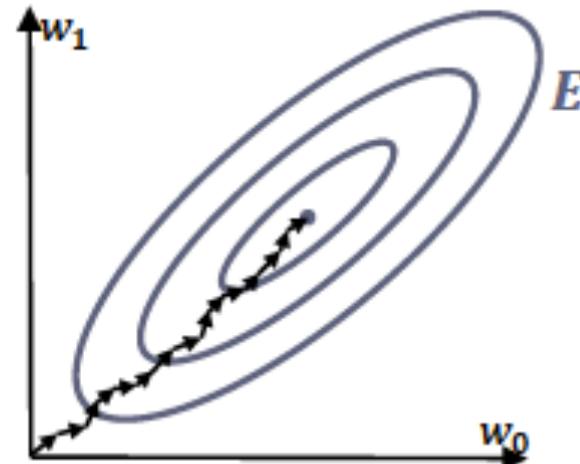


# Influența ratei de învățare

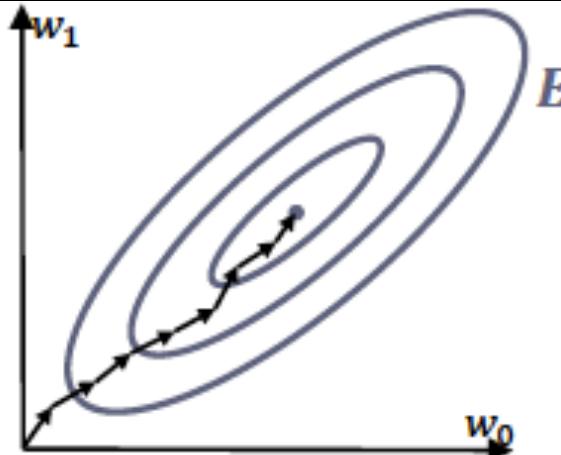
Rată de învățare prea mare  
(este posibil ca algoritmul să nu conveargă)



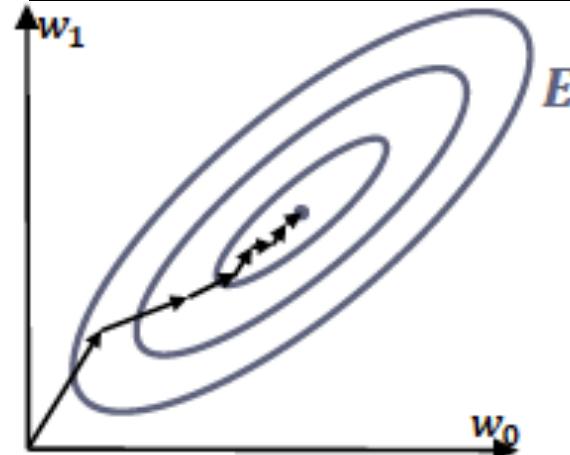
Rată de învățare prea mică  
(algoritmul converge foarte încet)



Rată de învățare bună  
(compromis între viteză și convergență)



Rată de învățare descrescătoare  
(în timp rata de învățare scade)



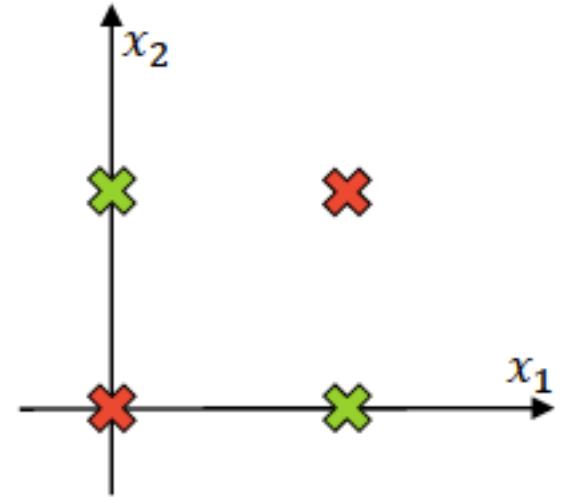
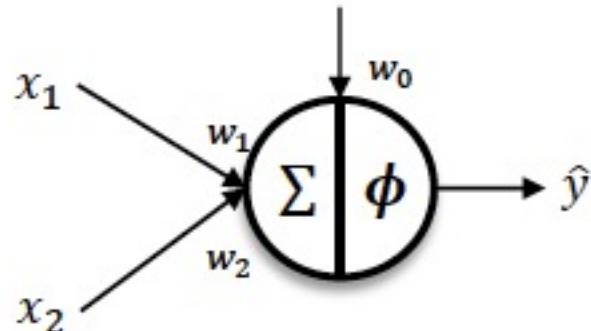
# Retele feedforward multistrat de perceptroni (Multilayer perceptrons)

# Funcția XOR

- Un singur perceptron nu poate învăța funcția XOR încărcat nu este liniar separabilă

XOR

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



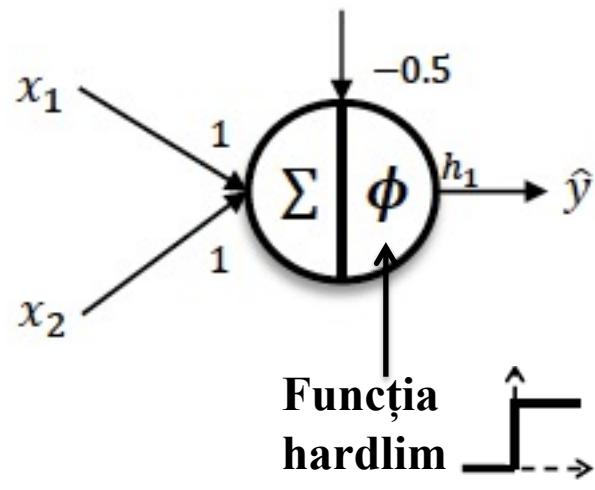
✗ Eticheta 0

✗ Eticheta 1

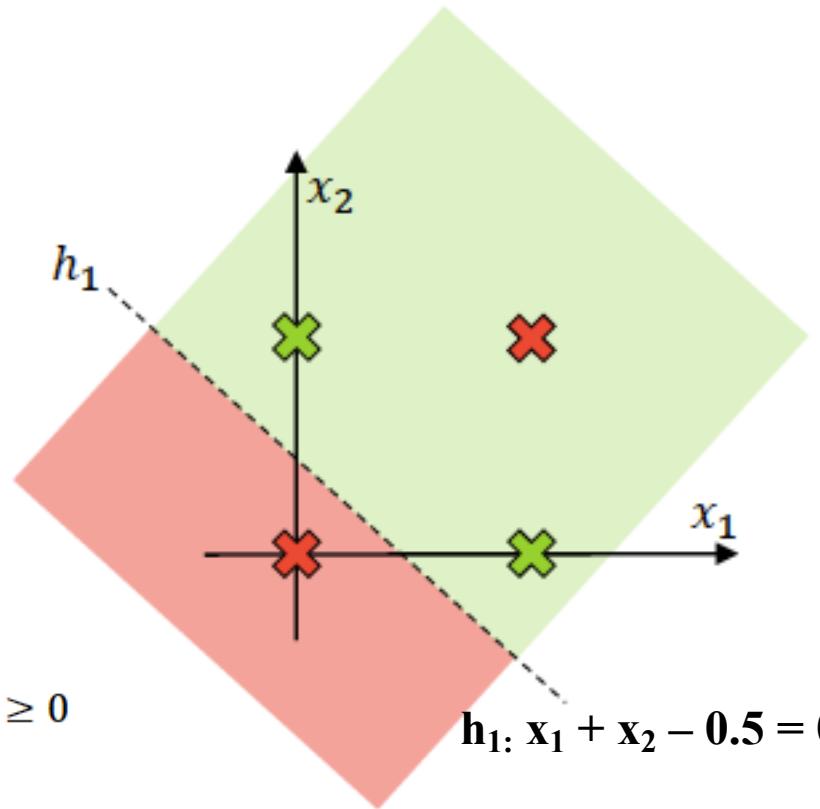
# Funcția XOR

- Un singur perceptron nu poate învăța funcția XOR încărcăt nu este liniar separabilă
- Putem depăși această limitare combinând ieșirile mai multor perceptroni

XOR		
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



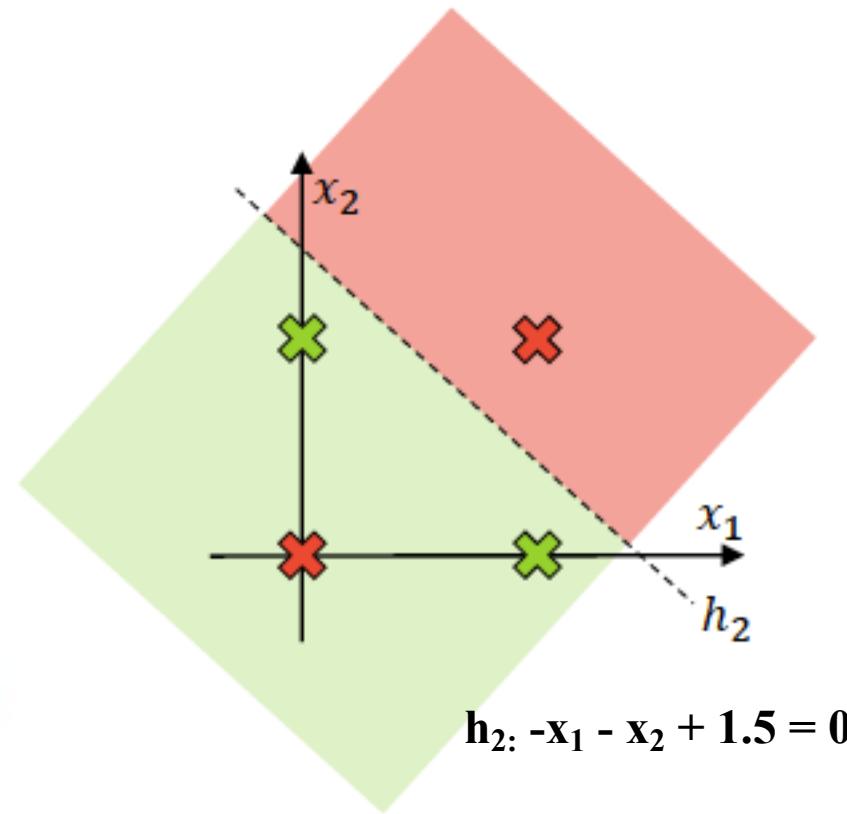
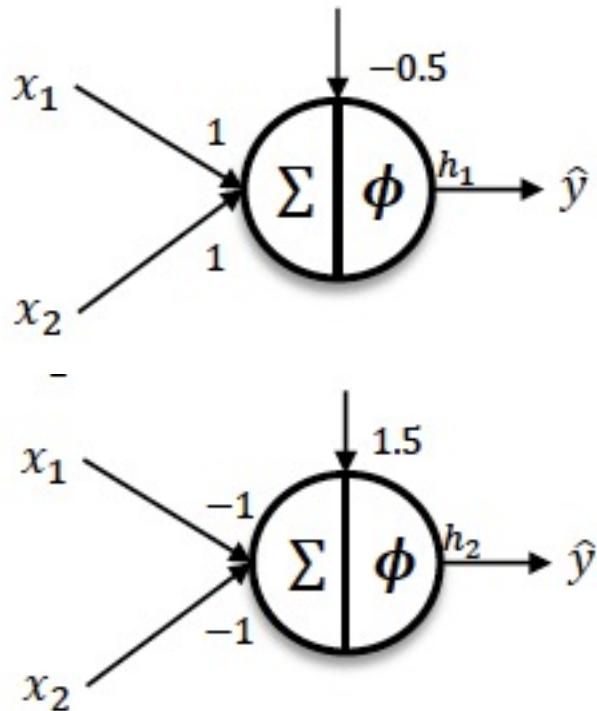
$$\hat{y} = \begin{cases} 1 & \text{if } \sum_{i=0}^n w_i x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



# Funcția XOR

- Un singur perceptron nu poate învăța funcția XOR încărcat nu este liniar separabilă
- Putem depăși această limitare combinând ieșirile mai multor perceptri

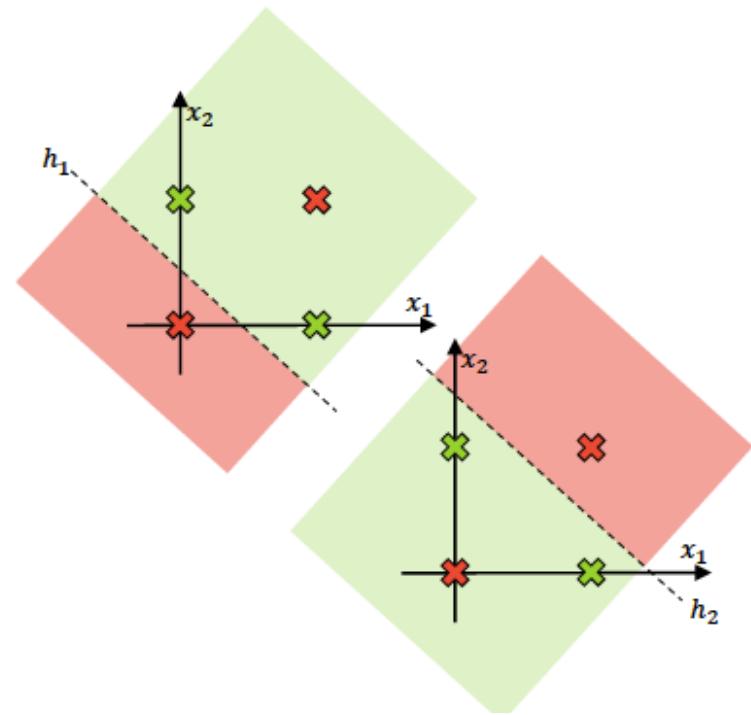
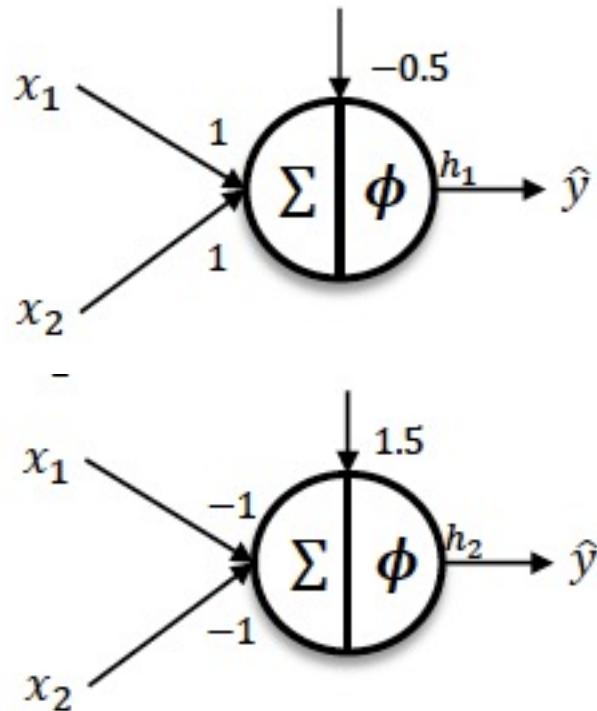
XOR		
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



# Funcția XOR

- Un singur perceptron nu poate învăța funcția XOR încărcat nu este liniar separabilă
- Putem depăși această limitare combinând ieșirile mai multor perceptri

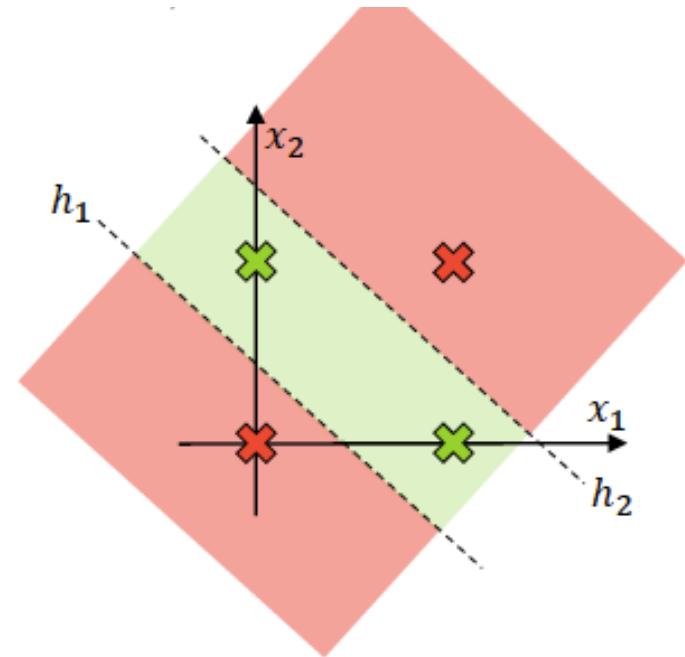
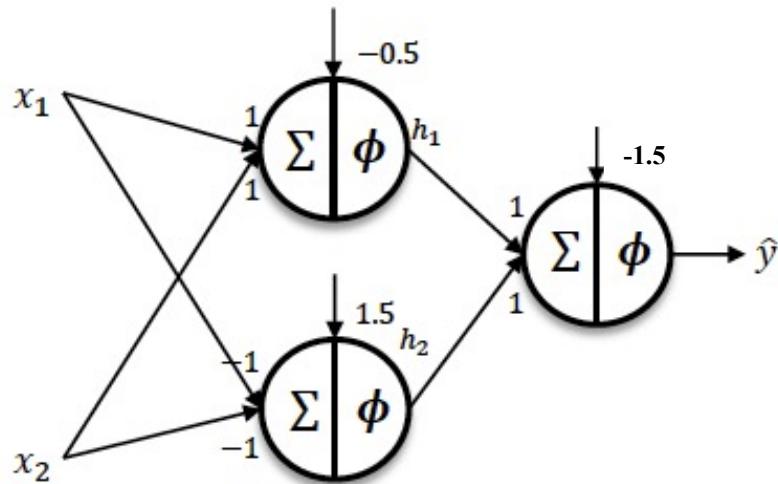
XOR		
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



# Funcția XOR

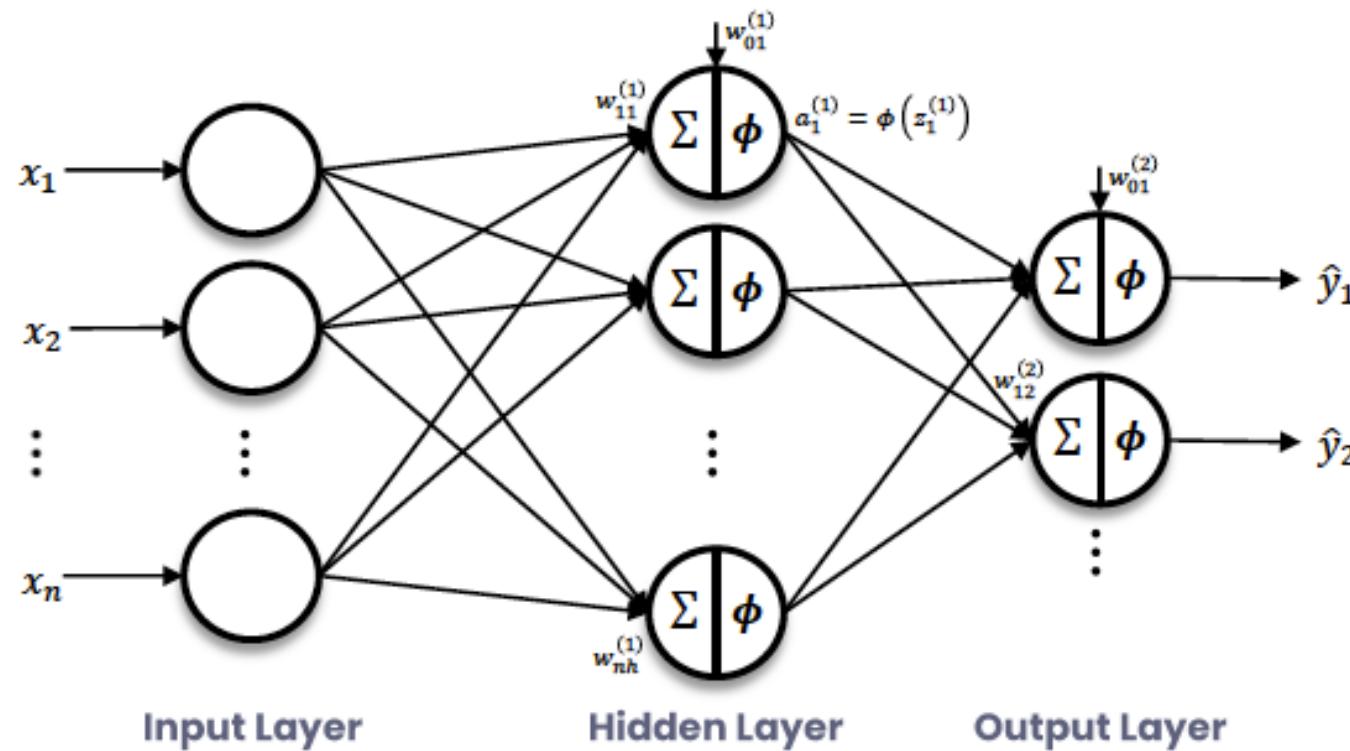
- Un singur perceptron nu poate învăța funcția XOR încărcat nu este liniar separabilă
- Putem depăși această limitare combinând ieșirile mai multor perceptroni formând o rețea de perceptroni

XOR		
$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



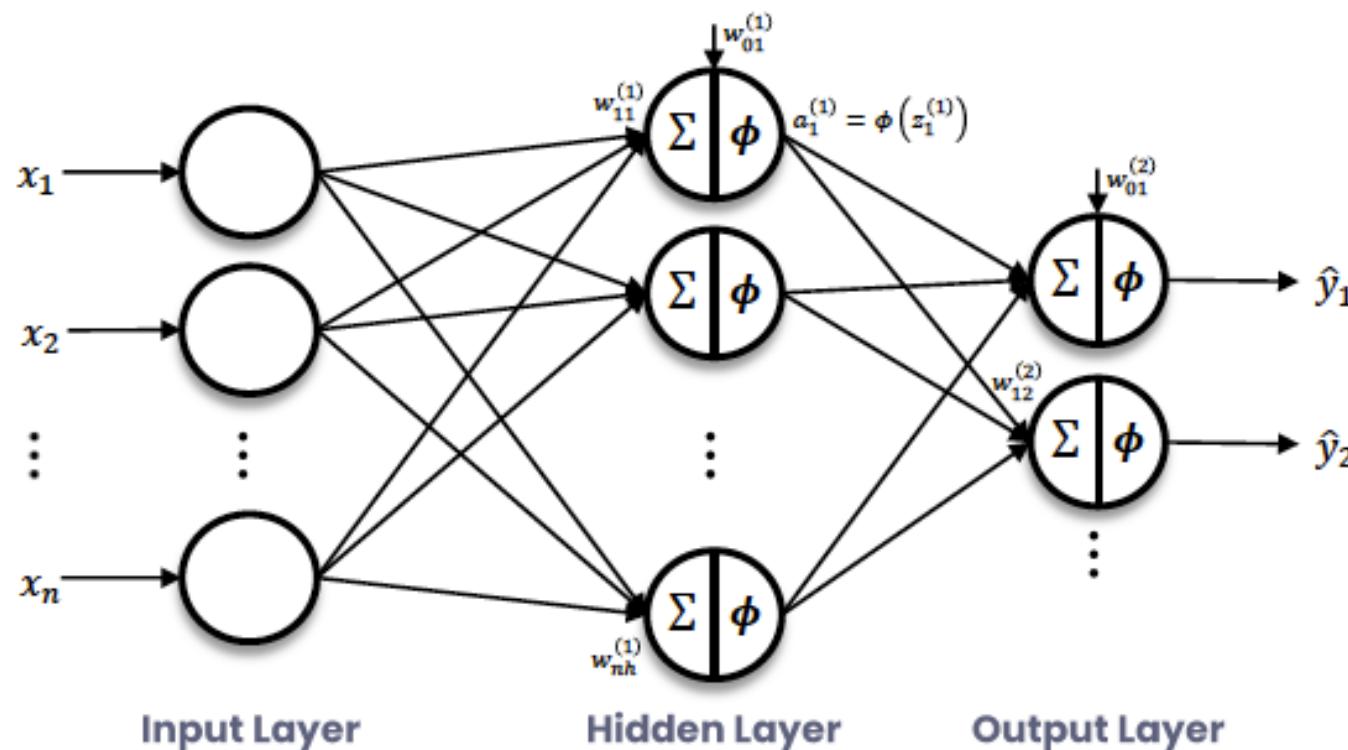
# Rețele feedforward multistrat de neuroni

- O rețea feedforward multistrat de neuroni (perceptroni) este o rețea de neuroni grupați pe straturi (layere), în care propagarea informației se realizează numai dinspre intrare spre ieșire (de la stânga la dreapta). Rețeaua are un strat de intrare (input layer), unul sau mai multe straturi ascunse (hidden layers) și un strat de ieșire (output layer).



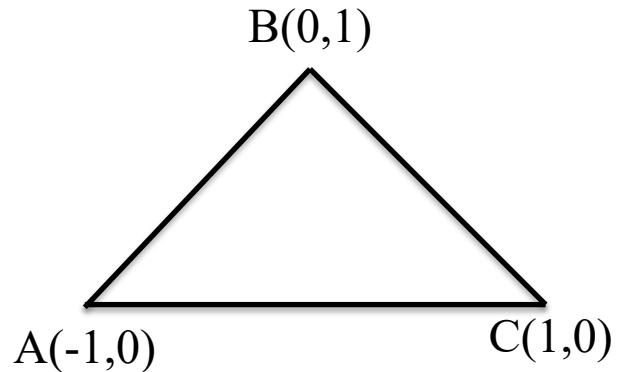
# Rețele feedforward multistrat de neuroni

- Toți neuronii din rețea, cu excepția celor din stratul de intrare aplică o funcție de activare sumei ponderate ale intrărilor.
- Fiecare pereche de neuroni din două straturi consecutive are o pondere asociată



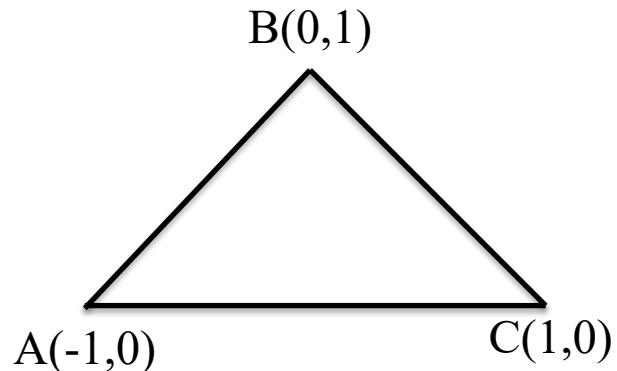
# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



$$AB: x_1 - x_2 + 1 = 0$$

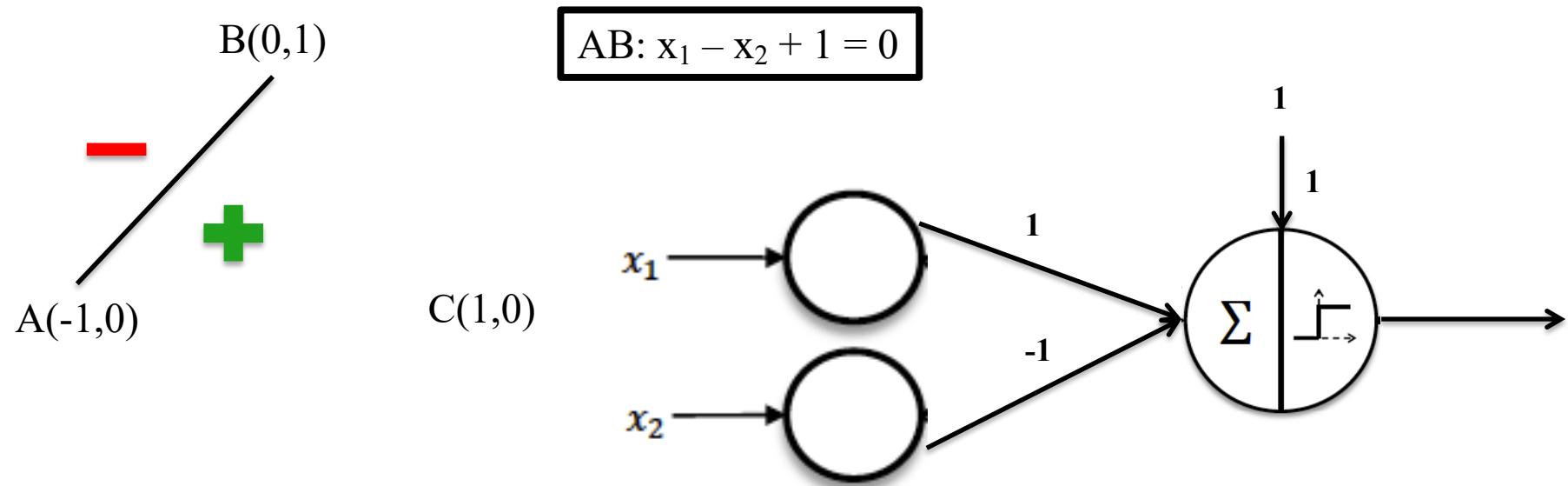
$$BC: x_1 + x_2 - 1 = 0$$

$$AC: x_2 = 0$$

Construiesc o rețea cu 3 perceptroni pe primul strat care implementează fiecare ecuația unei drepte și cu coeficienții setați astfel încât punctele din interiorul și pe frontieră triunghiului se află în semiplanul care primește valoarea 1 (în cazul funcției de activare hardlim)

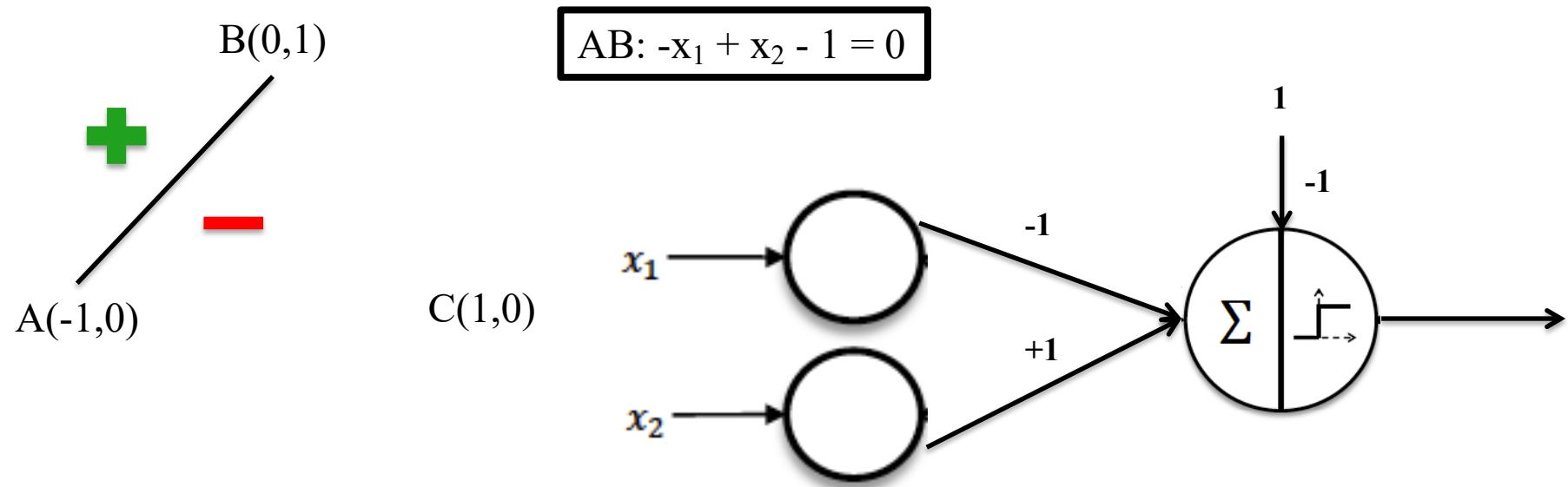
# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



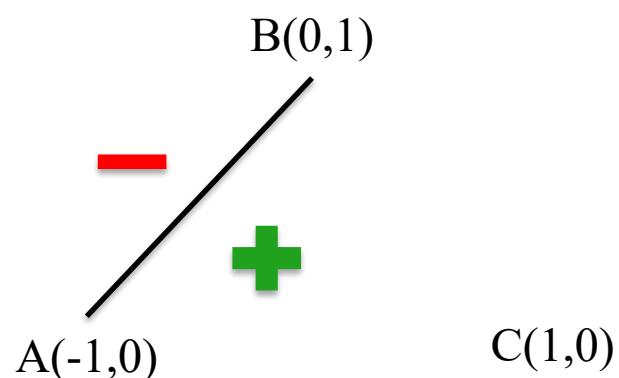
# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



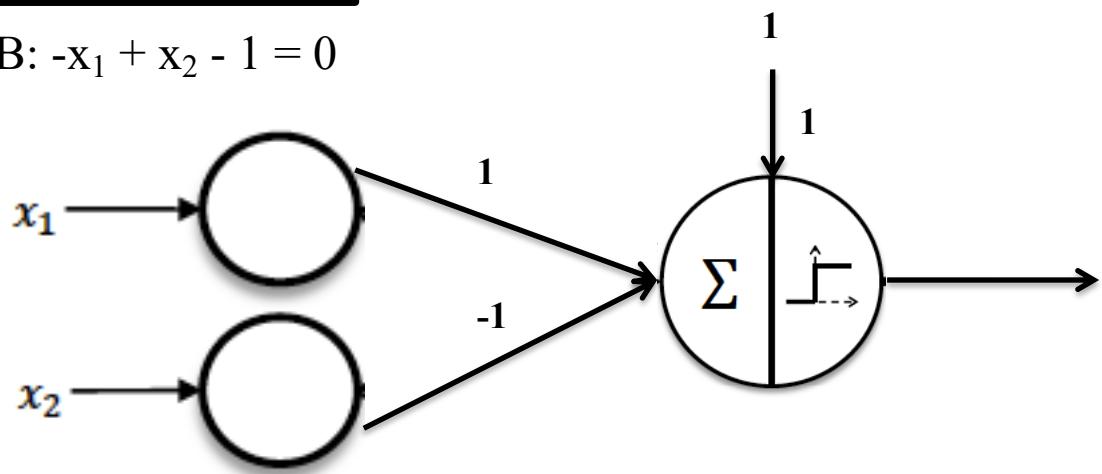
# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



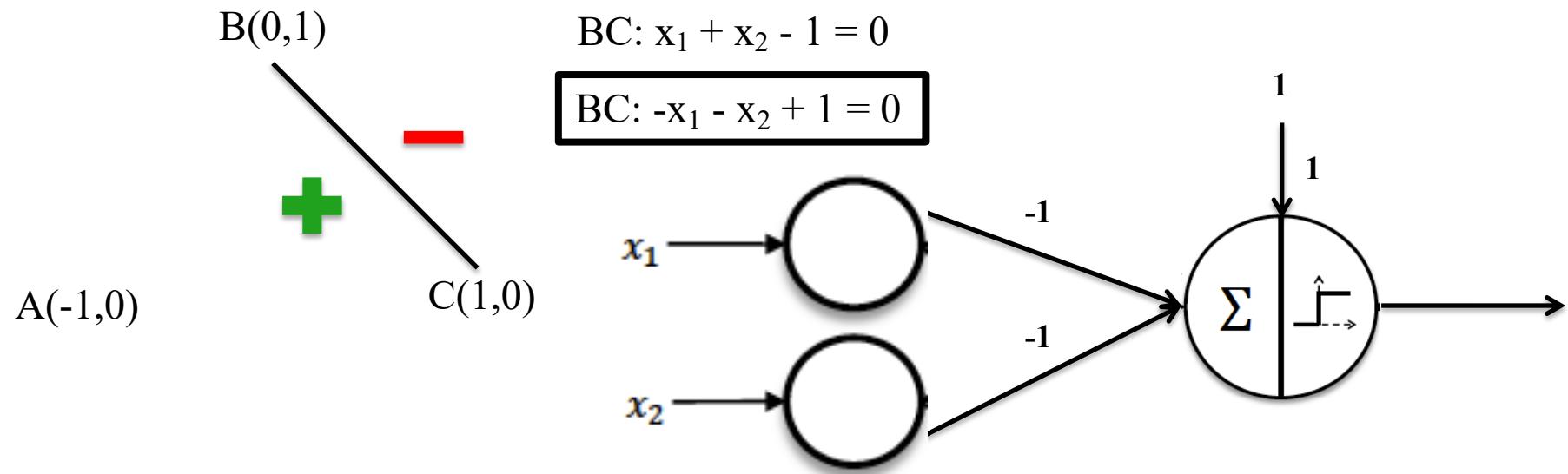
$$\boxed{AB: x_1 - x_2 + 1 = 0}$$

$$AB: -x_1 + x_2 - 1 = 0$$



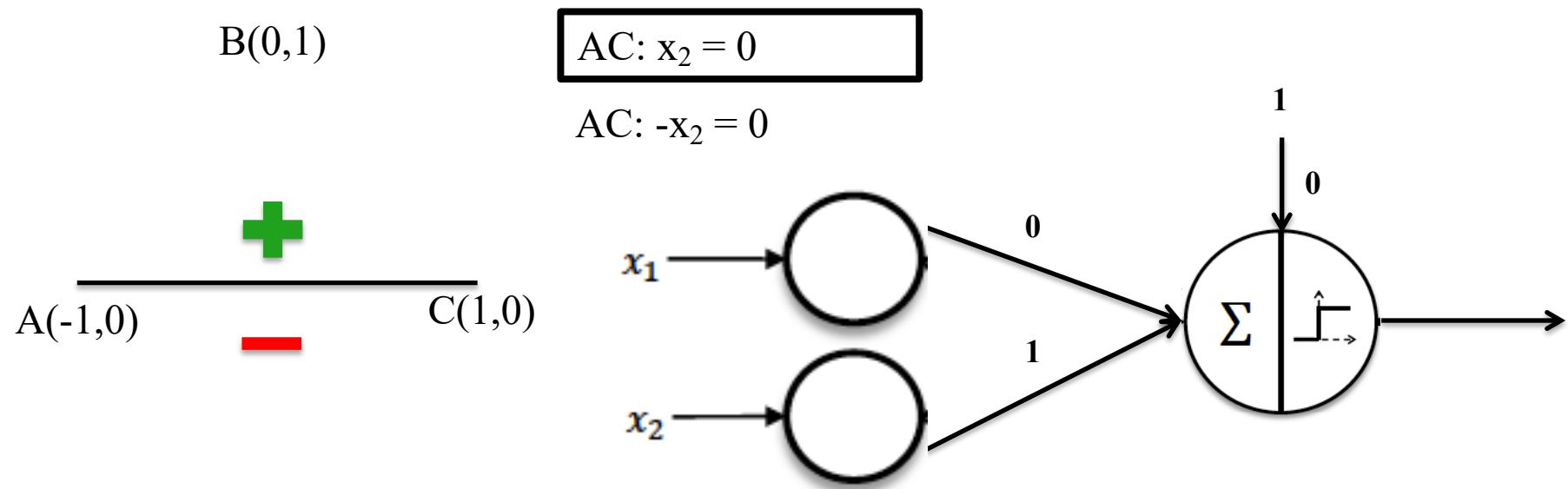
# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



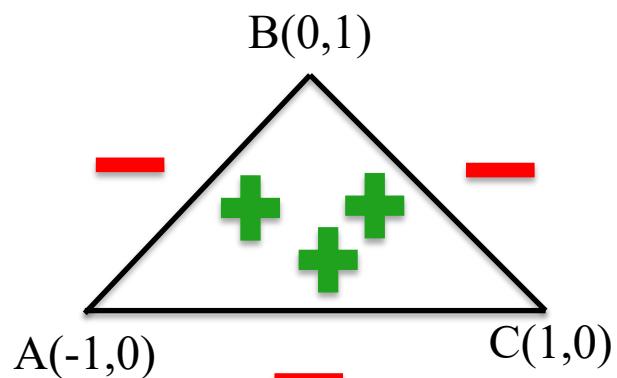
# Rețele feedforward multistrat de neuroni setate manual

Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



# Rețele feedforward multistrat de neuroni setate manual

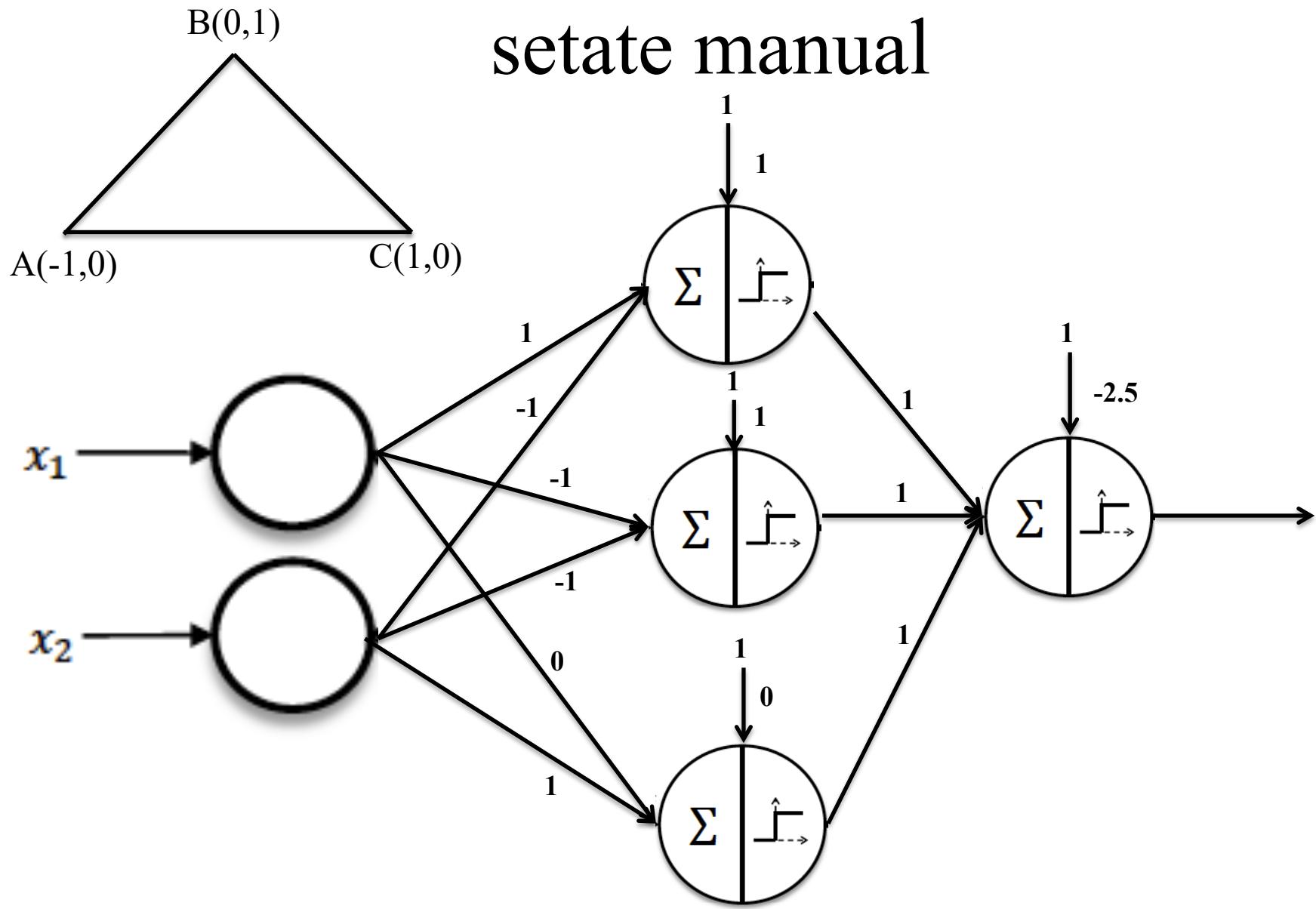
Construiți o rețea care să implementeze funcția indicator a triunghiului ABC, cu vârfurile având coordonatele A(-1,0), B(0,1), C(1,0). Funcția indicator a unui triunghi ia valoarea 1 pentru punctele din triunghi (interior + frontieră) și 0 altfel (exterior).



Construiesc o rețea cu 3 perceptri pe primul strat care implementează fiecare ecuația unei drepte și cu coeficienții setați astfel încât punctele din interiorul și pe frontieră triunghiului se află în semiplanul care primește valoarea 1 (în cazul funcției de activare hardlim)

Pe ultimul strat pun un perceptron care face AND, un punct primește eticheta 1 dacă primește 1 de la toți cei trei perceptri.

# Rețele feedforward multistrat de neuroni setate manual



# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 7

# Sistem de notare proiect Kaggle

- proiectul valorează 1.5 puncte din nota finală
- partea de concurs Kaggle = 1 punct
  - locul 1 = 1 punct
  - locul 2 = 0.99 puncte
  - ...
  - locul 50 = 0.51 puncte
  - locul 51+ = 0.5 puncte (cât timp aveți o performanță > baseline)
- partea de documentație + prezentare = 0.5 puncte
- important: pentru fiecare submisie să știți ce ați făcut (cod Python, parametri, etc). La final vă veți alege 2 submisii care credeți voi că sunt cele mai bune (pot fi același model cu parametri diferiți). Vă bifăți cele 2 submisii în platforma Kaggle înainte de deadline.

# Documentație proiect - pdf

- descrieți în detaliu (1-2 pagini) **2 modele diferite** folosite (kNN, Naïve Bayes, SVM, perceptron, rețea neuronală):
  - ce caracteristici folosiți;
  - care sunt parametri, hiperparametri modelului;
  - cum antrenați parametri/hiperparametri;
  - cât durează antrenarea;
  - ce performanță ati obținut pe cele 40% de date din setul de date de test public pe Kaggle;
- **pentru un singur model prezentați rezultatele în urma antrenării în maniera 5 fold cross-validation + matricea de confuzie asociată**

# Predare proiect

- predarea proiectului înseamnă trimiterea documentației și a codului Python pentru fiecare submisie
- trimiteți la adresa de email: [ub.fmi.cti.ia@gmail.com](mailto:ub.fmi.cti.ia@gmail.com) un email până luni, 21 noiembrie, ora 23:59 cu următoarele fișiere:
  - un fișier pdf cu documentația voastră
  - două fișiere python cu codul pentru submisiile voastre
  - respectați formatul de mai jos



361\_Alexe\_Bogdan\_documentatie.pdf



361\_Alexe\_Bogdan\_submisie1\_cod.py



361\_Alexe\_Bogdan\_submisie2\_cod.py

# Prezentare proiect

- este individuală, are loc în săptămâna 8 (21-25 noiembrie)
- constă într-o discuție cu Alexandra/Sergiu/Bogdan de maxim 10 minute
  - prezentarea voastră 3-5 minute (ce modele ați folosit la cele 2 submisii)
  - 3-5 minute întrebări din partea noastră
- vom face o programare pe care o vom afișa luni, după concurs, în TEAMS:
  - grupa 361: marti 12-14-16
  - grupa 362: miercuri 8-10-12
  - grupa 363: marți 10-12-14
  - grupa 364: miercuri 10-12-14
- dacă nu puteți veni la grupa voastră sau doriți să prezentați într-un anumit interval orar vă rog să îmi scrieți mie pe email până duminică seara

# Recapitulare – cursul trecut

1. Alte reguli de învățare pentru perceptron
2. Rețele feedforward multistrat de perceptri

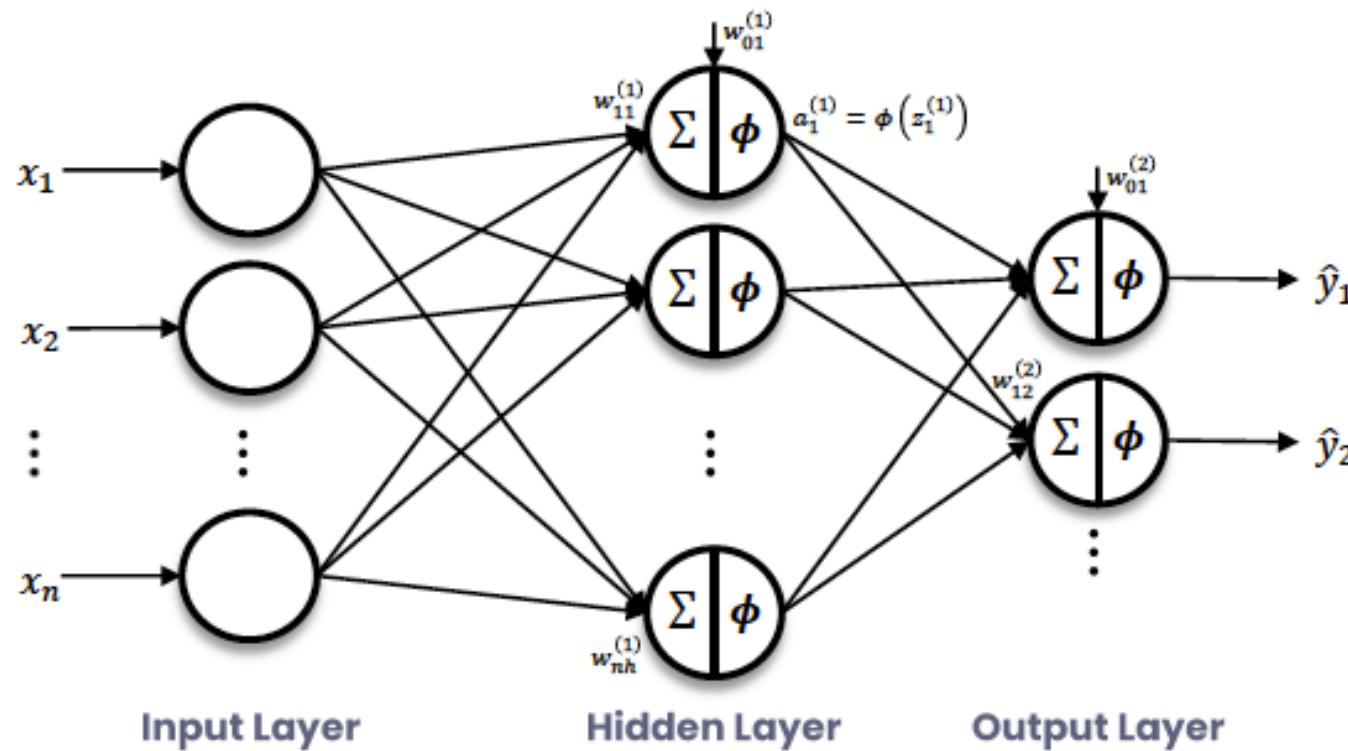
# Cuprinsul cursului de azi

1. Rețele feedforward multistrat de perceptri
2. Regresia liniară simplă și multiplă

# Retele feedforward multistrat de perceptroni (Multilayer perceptrons)

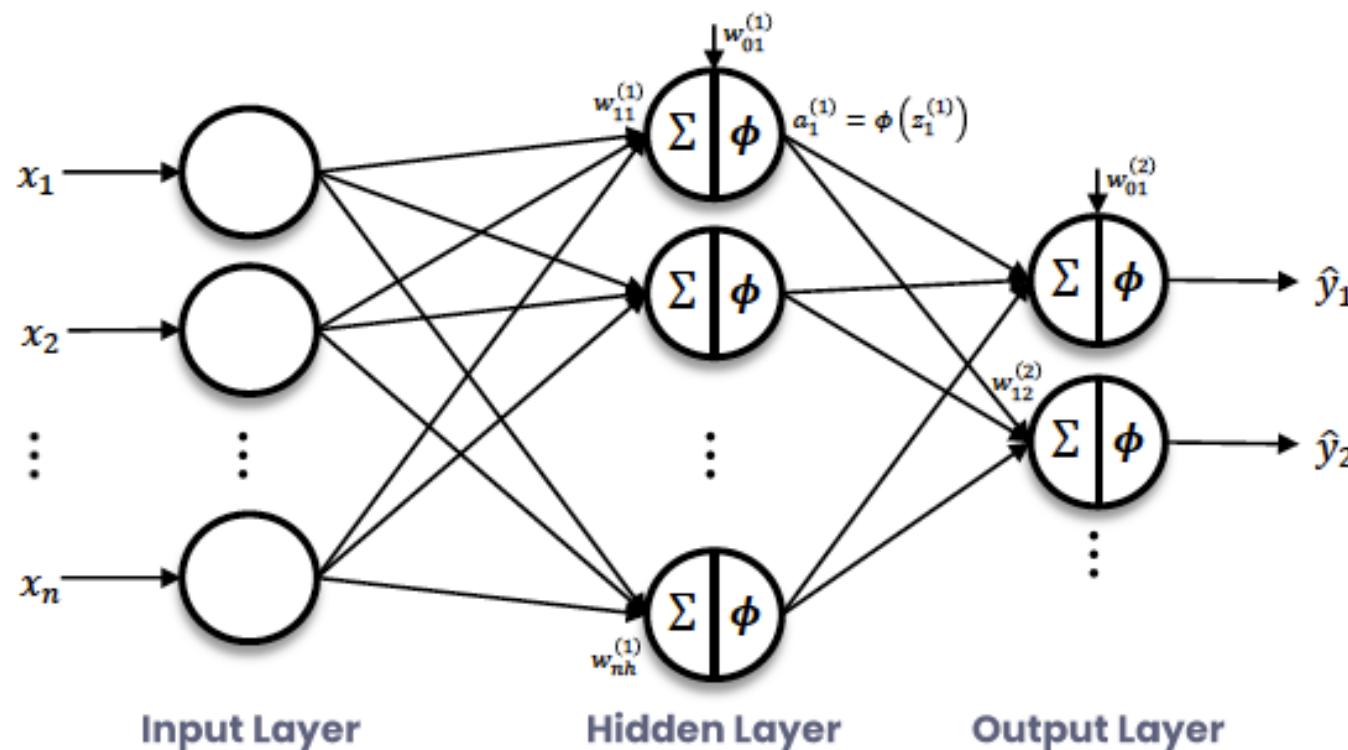
# Rețele feedforward multistrat de neuroni

- O rețea feedforward multistrat de neuroni (perceptroni) este o rețea de neuroni grupați pe straturi (layere), în care propagarea informației se realizează numai dinspre intrare spre ieșire (de la stânga la dreapta). Rețeaua are un strat de intrare (input layer), unul sau mai multe straturi ascunse (hidden layers) și un strat de ieșire (output layer).



# Rețele feedforward multistrat de neuroni

- Toți neuronii din rețea, cu excepția celor din stratul de intrare aplică o funcție de activare sumei ponderate ale intrărilor.
- Fiecare pereche de neuroni din două straturi consecutive are o pondere asociată



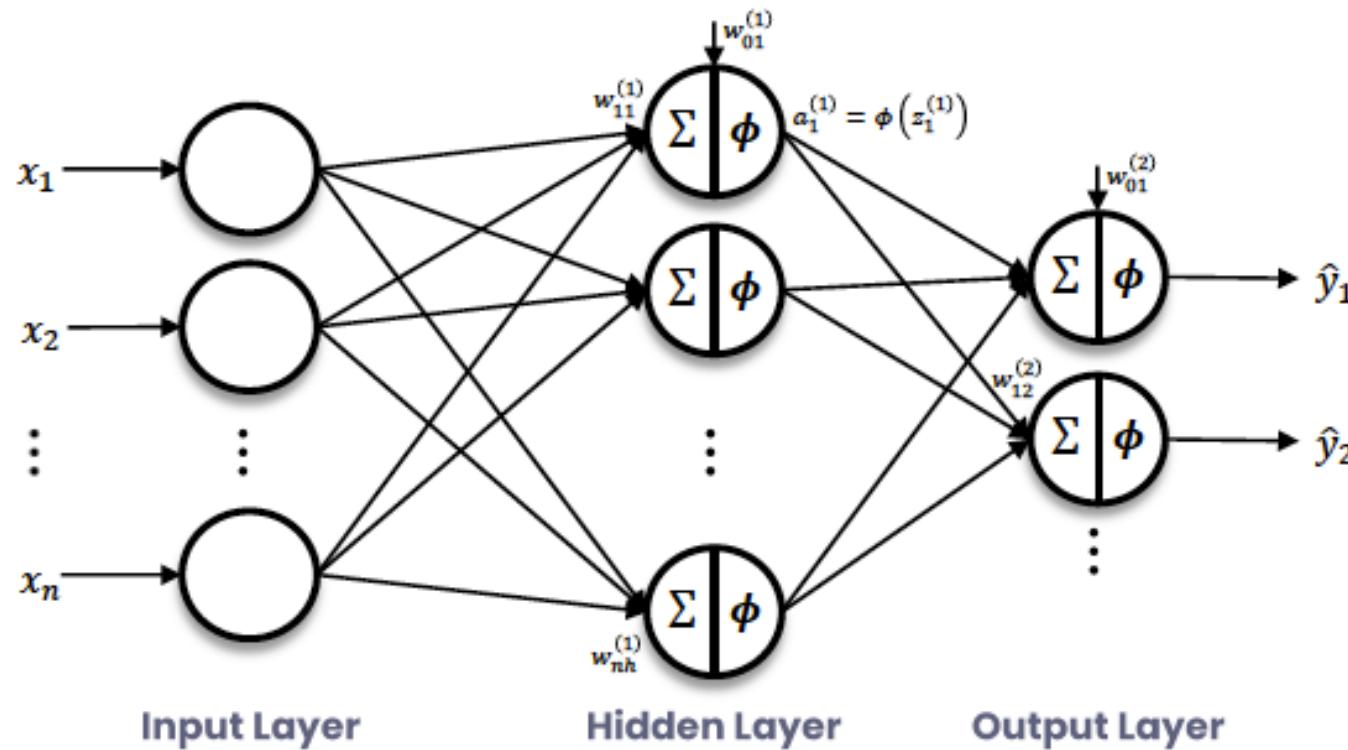
# Rețele feedforward multistrat de neuroni

$w_{ij}^{(l)}$  este ponderea neuronului  $i$  de pe stratul  $l - 1$  către neuronul  $j$  pe stratul  $l$ .

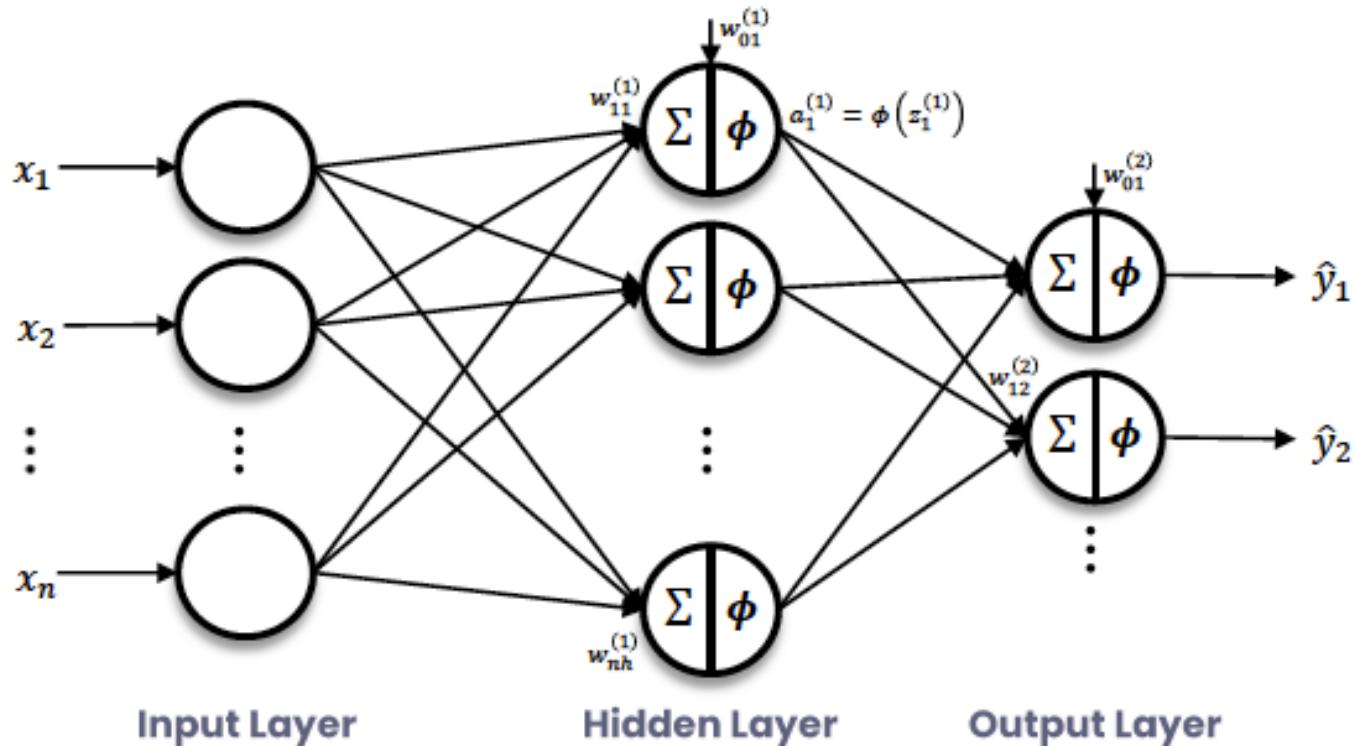
$w_{0j}^{(l)}$  este bias-ul (deplasarea) neuronului  $j$  de pe stratul  $l$ .

$z_j^{(l)}$  este ieșirea neuronului  $j$  de pe stratul  $l$  după însumarea intrărilor ponderate de la toți ceilalți neuroni.

$a_j^{(l)}$  este ieșirea neuronului  $j$  de pe stratul  $l$  după aplicarea funcției de activare  $\phi$  ieșirii  $z_j^{(l)}$ .



# Rețele feedforward multistrat de neuroni reprezentate în format matriceal



# Rețele feedforward multistrat de neuroni reprezentate în format matriceal

$$\vec{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad W^{(1)} = \begin{bmatrix} w_{01}^{(1)} & w_{11}^{(1)} & \cdots & w_{n1}^{(1)} \\ w_{02}^{(1)} & w_{12}^{(1)} & \cdots & w_{n2}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{0h}^{(1)} & w_{1h}^{(1)} & \cdots & w_{nh}^{(1)} \end{bmatrix}_{h \times n+1} \quad W^{(1)}\vec{x} = \begin{bmatrix} \sum_{i=0}^n x_i w_{i1}^{(1)} \\ \sum_{i=0}^n x_i w_{i2}^{(1)} \\ \vdots \\ \sum_{i=0}^n x_i w_{ih}^{(1)} \end{bmatrix} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ \vdots \\ z_h^{(1)} \end{bmatrix} = \vec{z}^{(1)}$$

$$\phi(\vec{z}^{(1)}) = \begin{bmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_h^{(1)} \end{bmatrix} \quad \vec{a}^{(1)} = \begin{bmatrix} 1 \\ a_1^{(1)} \\ \vdots \\ a_h^{(1)} \end{bmatrix} \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_o \end{bmatrix} = \phi(W^{(2)}\phi(W^{(1)}x))$$



# Antrenarea unei rețele feedforward multistrat de neuroni

- O rețea se antrenează folosind algoritmul de backpropagation = propagarea erorii înapoi
- Varianta cea mai folosită este utilizarea unui algoritm stochastic de coborâre pe gradient (stochastic gradient descent)
  - “stochastic” întrucât gradientul se calculează în funcție de un exemplu sau o mulțime redusă de exemple (batch) și nu în raport cu întreaga mulțime
- Trebuie să calculăm gradientul funcției de eroare  $E$  în raport cu fiecare pondere din rețea și să facem actualizările corespunzătoare:

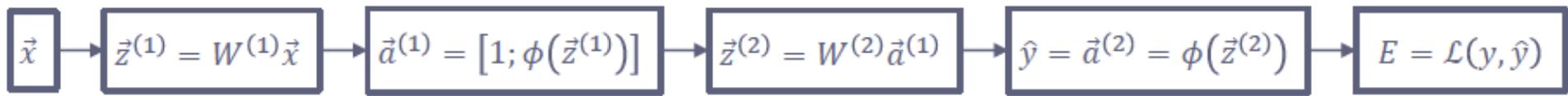
$$E(\vec{x}) = \mathcal{L}(y, \hat{y})$$

$$\Delta w_{ij}^{(l)} = -\eta \frac{\partial E}{\partial w_{ij}^{(l)}}$$

Funcția de eroare  $E$  ia valori mici când cele două etichete iau valori apropiate și ia valori mari altfel

- În format matriceal putem scrie:  $\Delta W^{(l)} = -\eta \frac{\partial E}{\partial W^{(l)}}$

# Regula de înlăntuire a derivatelor



- Trebuie să calculăm gradientul funcției de eroare E în raport cu fiecare pondere din rețea și să facem actualizările corespunzătoare
- Funcția eroare E depinde de toate ponderile din rețea:

$$E(\vec{x}) = \mathcal{L}\left(y, \phi\left(W^{(2)}\phi\left(W^{(1)}\vec{x}\right)\right)\right)$$

- Putem alege orice pondere  $w_{ij}^{(l)}$  și calcula prin regula de înlăntuire derivata ei corepunzătoare  $\frac{\partial E}{\partial w_{ij}^{(l)}}$ .
- În format matriceal putem scrie:

$$\frac{\partial E}{\partial W^{(1)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \vec{z}^{(2)}} \frac{\partial \vec{z}^{(2)}}{\partial \vec{a}^{(1)}} \frac{\partial \vec{a}^{(1)}}{\partial \vec{z}^{(1)}} \frac{\partial \vec{z}^{(1)}}{\partial W^{(1)}}$$

$$\frac{\partial E}{\partial W^{(2)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \vec{z}^{(2)}} \frac{\partial \vec{z}^{(2)}}{\partial W^{(2)}}$$

# Alegerea funcției de activare

- Funcția hardlim (funcția de activare a perceptronului) nu este derivabilă în 0 iar în toate celelalte puncte derivata sa este nulă, deci orice actualizare este nulă.
- Funcții uzuale de activare:
  - funcția identitate  $f(x) = x$ ;
  - funcția logistică  $f(x) = 1/(1+e^{-x})$ ;
  - funcția tangentă hiperbolică  $f(x) = \tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ ;
  - funcția relu  $f(x) = \max(0, x)$ .

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) \underset{x=0}{\begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Arctan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU) [2]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

# Alegerea funcției de eroare E

- Funcția pătratică de eroare (folosită de Adaline)

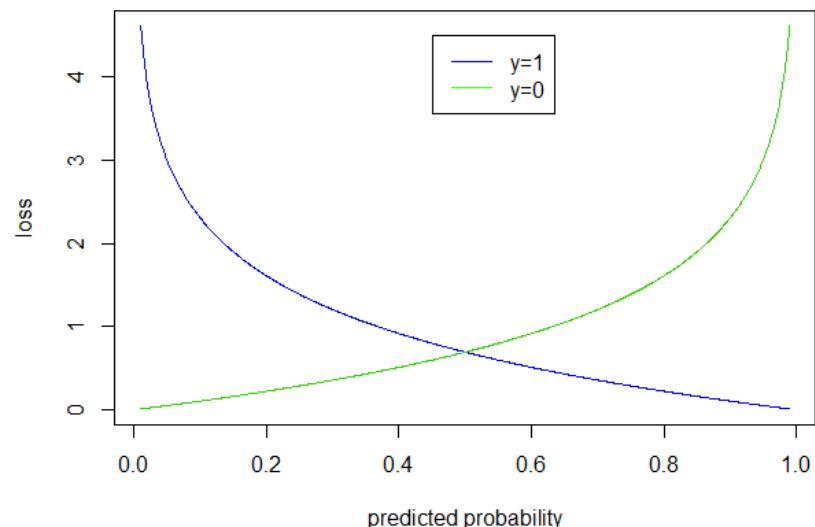
$$\mathcal{L}(y, \hat{y}) = \frac{1}{2} (y - \hat{y})^2$$

$$\mathcal{L}'(y, \hat{y}) = -(y - \hat{y})$$

- În practică, se folosește funcția de eroare bazată pe cross-entropie:

$$\mathcal{L}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

$$\mathcal{L}'(y, \hat{y}) = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}}$$



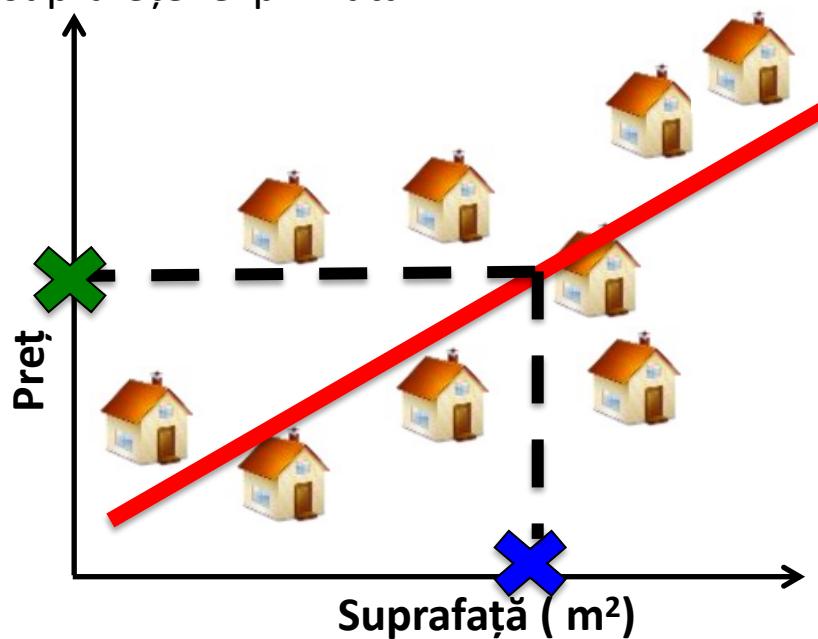
# Regresia liniară simplă și multiplă

# Cursul 2: Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$

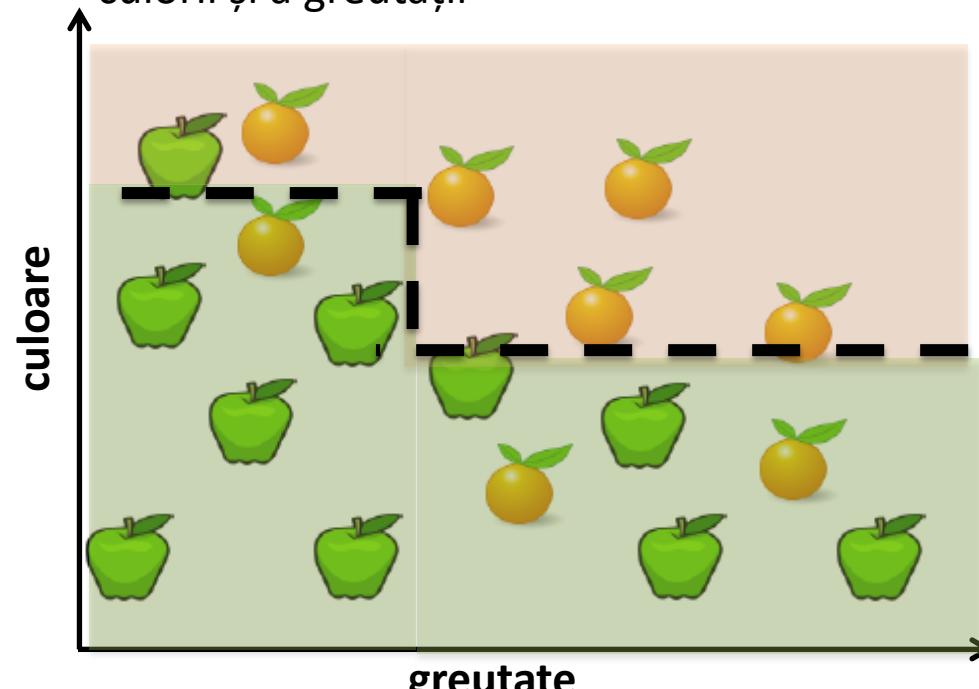


Regresie liniară

Regresie pe baza kNN

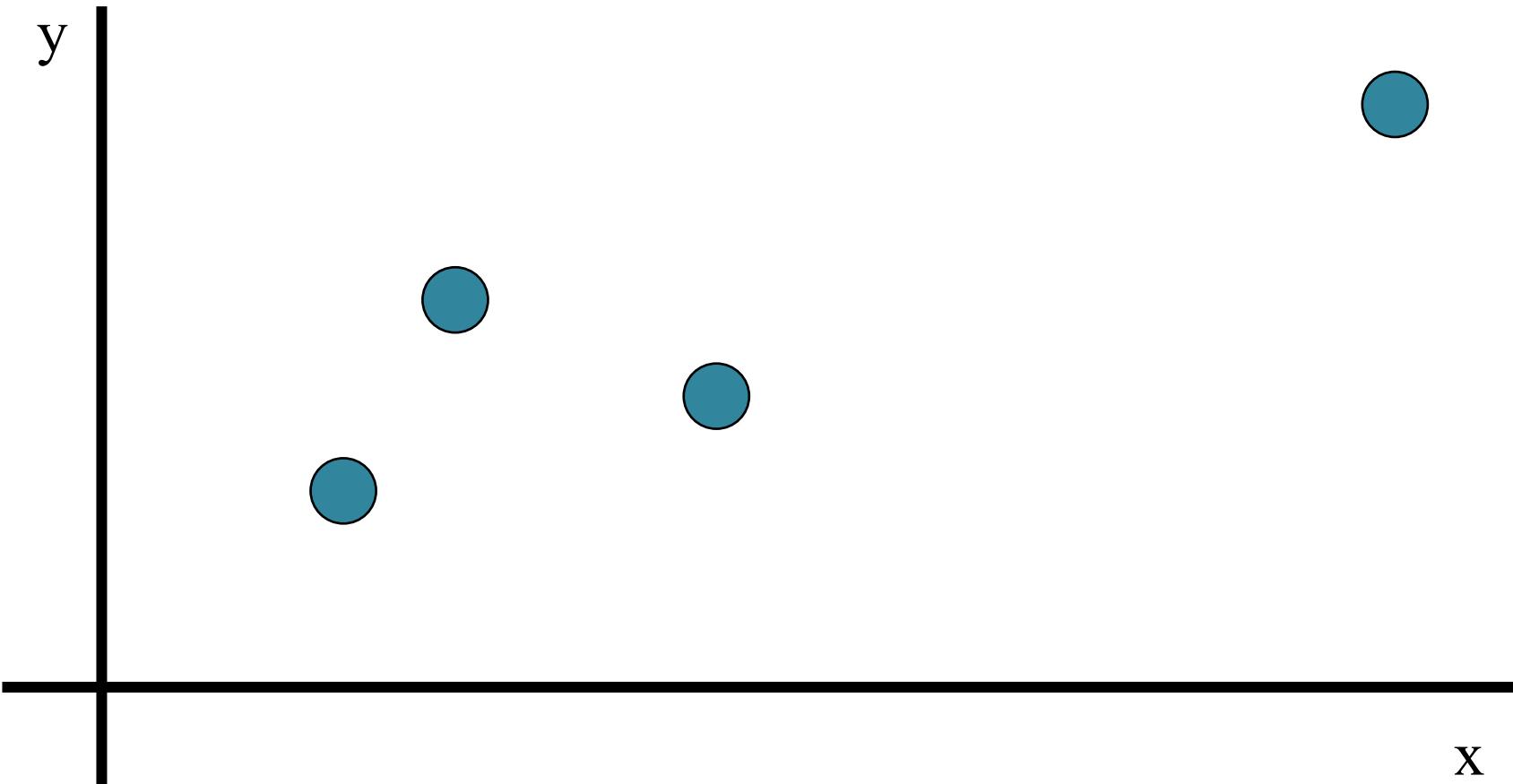
**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății

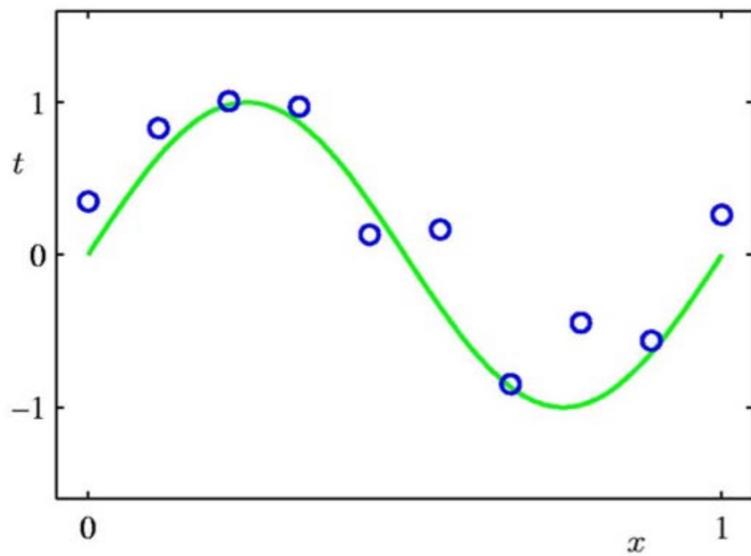


Cei mai apropiati k vecini (kNN),  
SVMs, Rețele neuronale

# Cursul 3: Modelul celor mai apropiati k-vecini pentru probleme de regresie



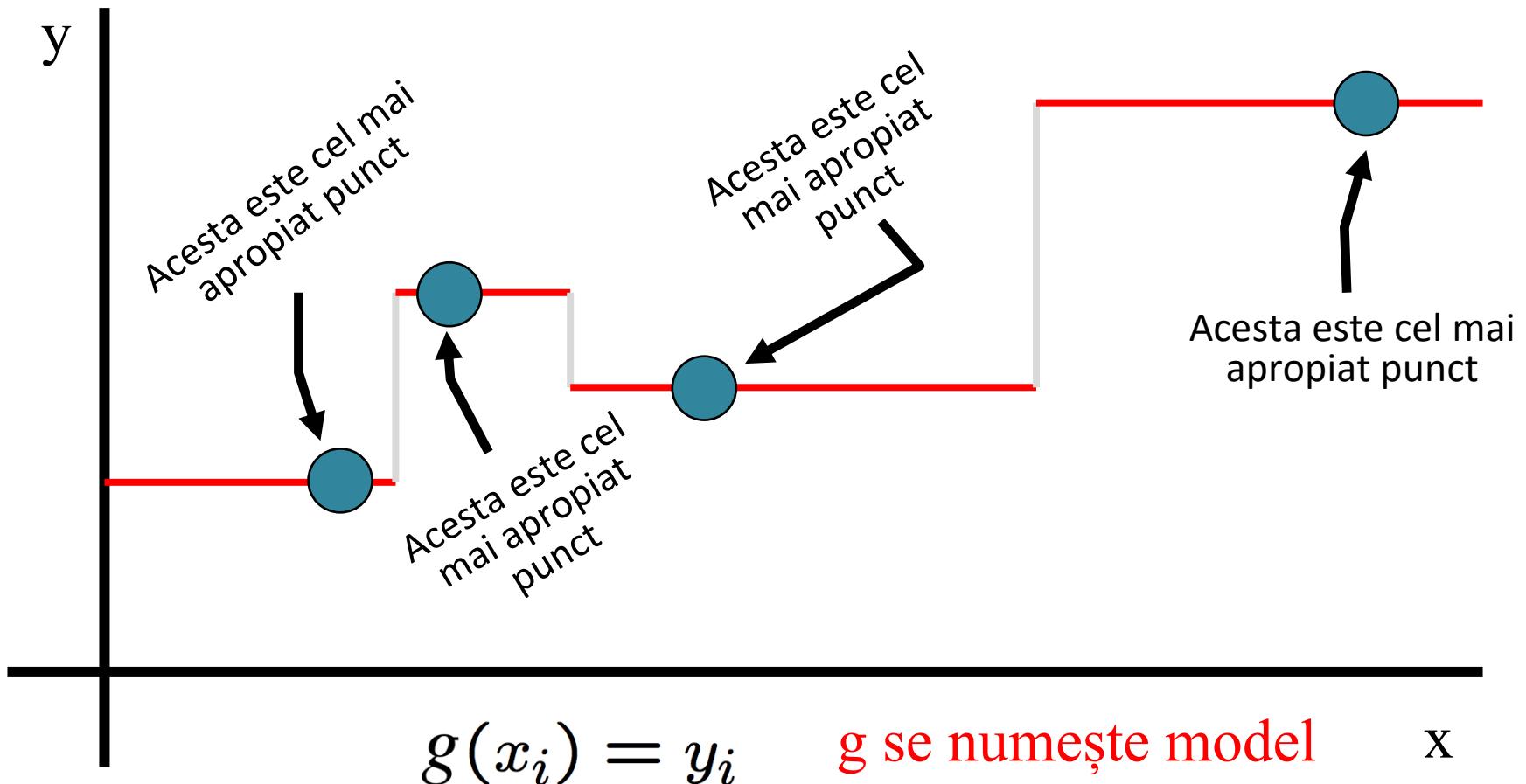
# Regresie din exemple etichetate



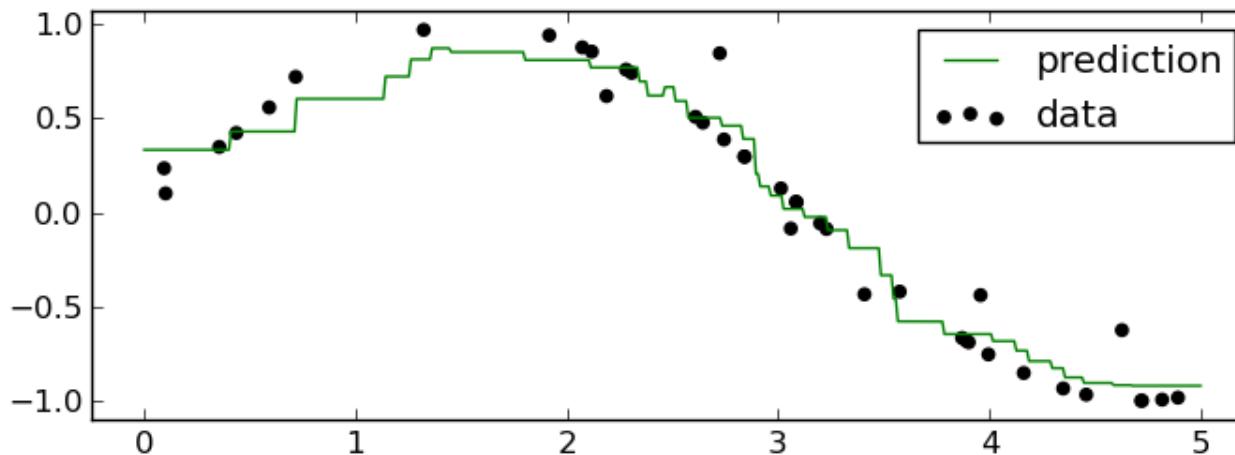
- Presupunem că avem un set de  $N$  exemple de antrenare:  
 $(x_1, \dots, x_N)$  și  $(y_1, \dots, y_N)$ ,  $x_i, y_i \in \mathbb{R}$
- Problema regresiei constă în estimarea funcției  $g(x)$  a.î.:  
$$g(x_i) = y_i \quad \text{g se numește model}$$

# Modelul celor mai apropiati k-vecini pentru probleme de regresie

**K = 1**



# Modelul celor mai apropiati k-vecini pentru probleme de regresie



Algoritmul de regresie bazat pe cei mai apropiati k-vecini:

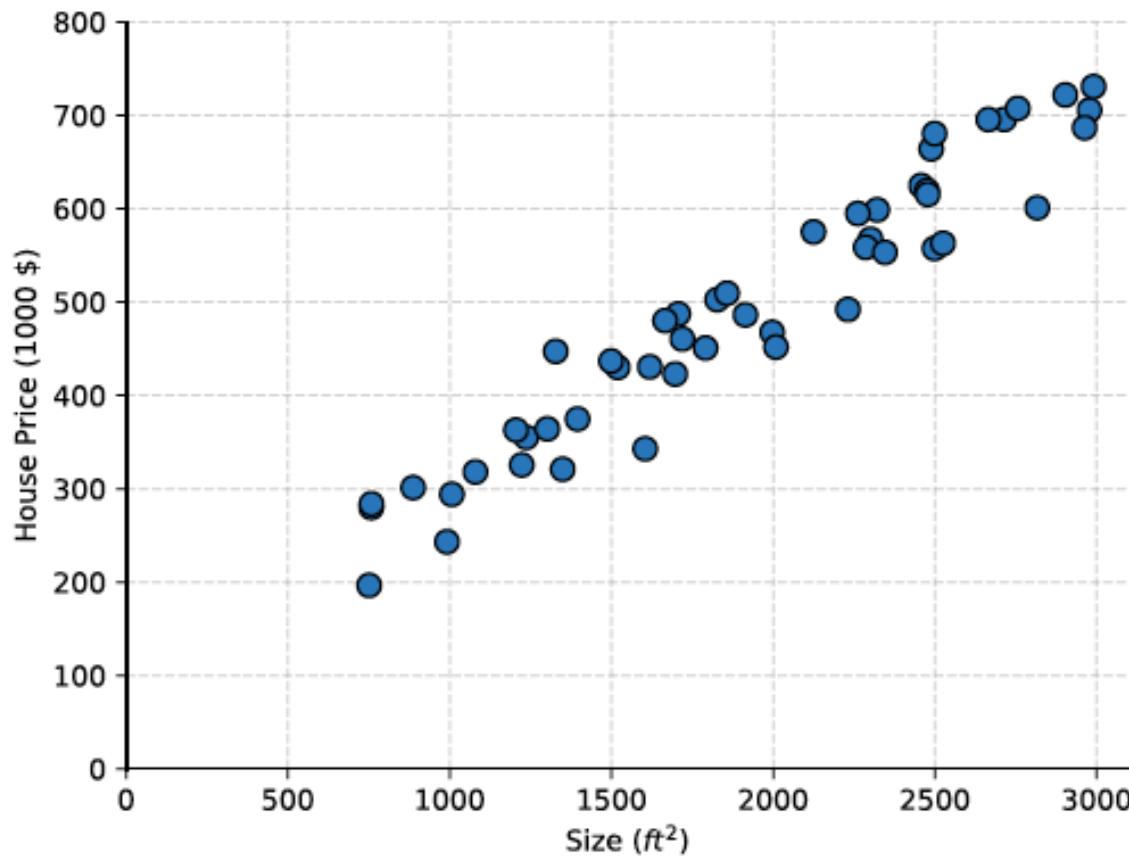
- 1) Pentru fiecare exemplu de test  $x$ , găsim cei mai apropiati  $k$  vecini și etichetele lor
- 2) Predictia este media etichetelor celor  $k$  vecini

$$f(x) = \frac{1}{K} \sum_{i=1}^K y_i$$

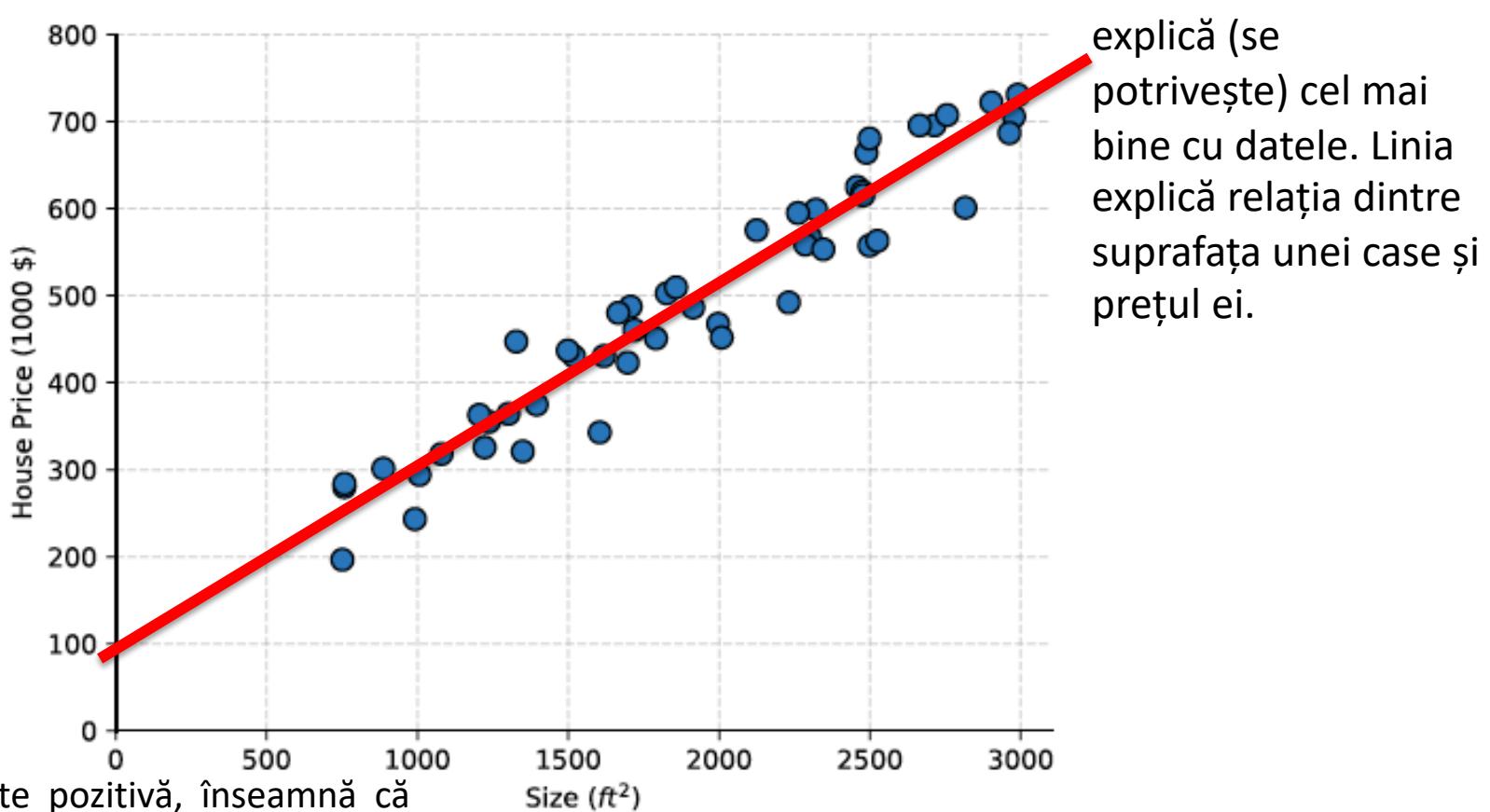
# Regresia liniară - obiective

- Stabilirea unei relații (liniare) între o variabilă dependentă și una sau mai multe variabile independente
  - variabila dependentă se mai numește etichetă, răspuns, output
  - variabilele independente se mai numesc caracteristici, trăsături, attribute, predictori
  - dacă avem o singură variabilă independentă atunci vorbim despre **regresie liniară simplă**
  - dacă avem mai multe variabile independente atunci vorbim despre **regresie liniară multiplă**
- Realizarea de predicții
  - folosește relația liniară obținută mai înainte pentru realizarea de predicții pe date noi

# Regresia liniară simplă



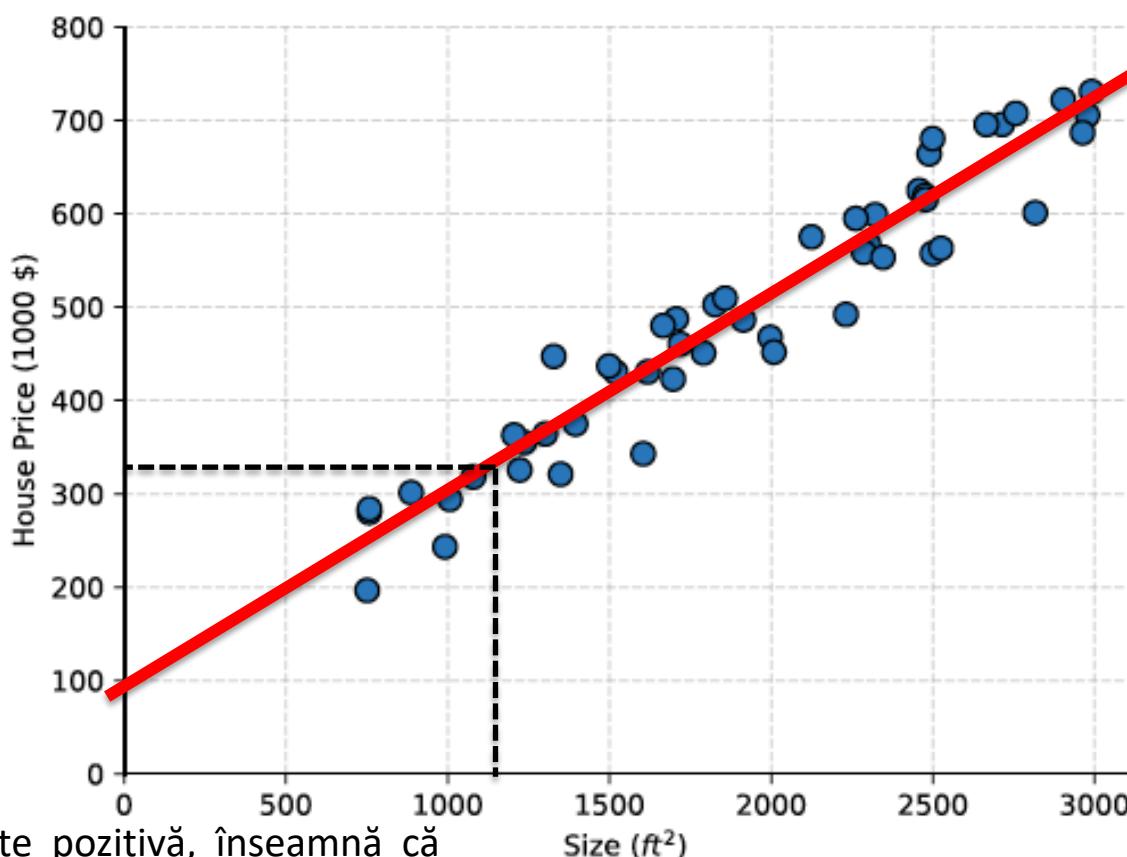
# Regresia liniară simplă



Găsim o linie care explică (se potrivește) cel mai bine cu datele. Linia explică relația dintre suprafața unei case și prețul ei.

Panta dreptei este pozitivă, înseamnă că avem o corelație pozitivă între suprafață și preț (prețul crește pe măsura ce suprafața crește).

# Regresia liniară simplă



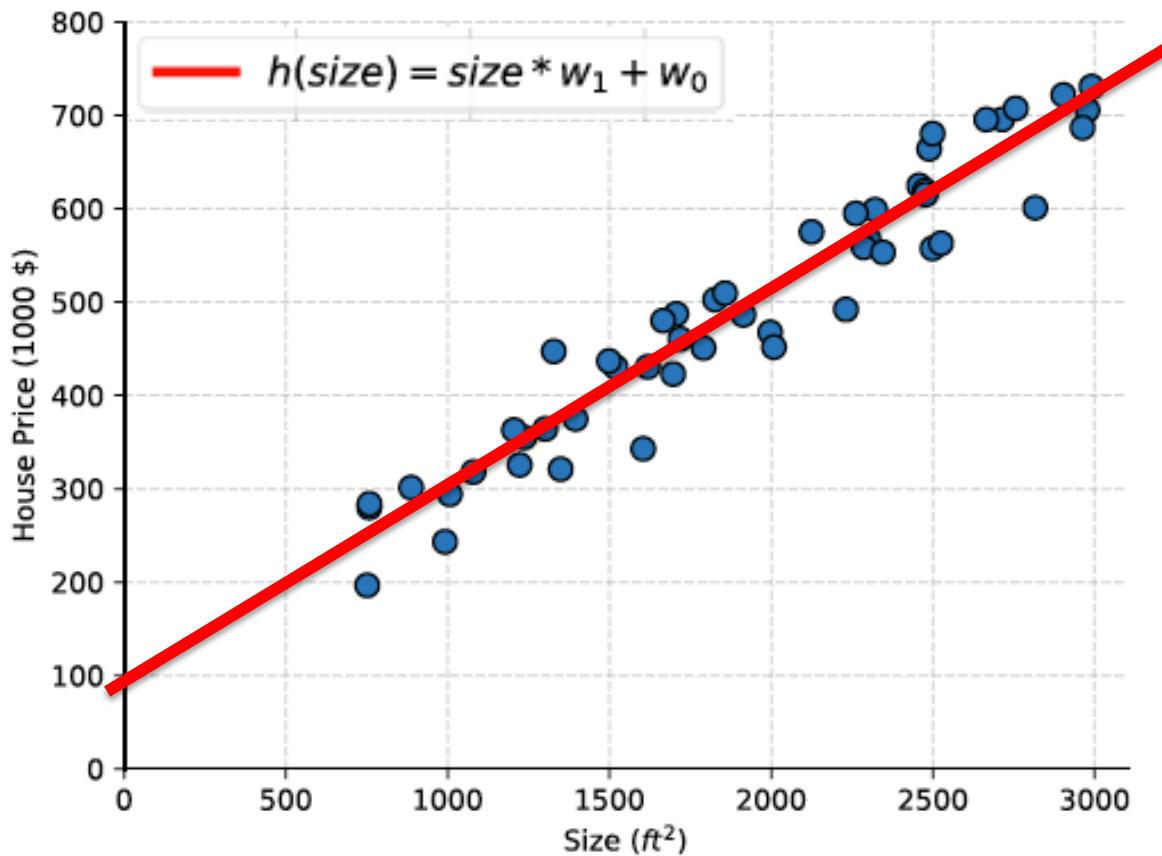
Găsim o linie care explică (se potrivește) cel mai bine cu datele. Linia explică relația dintre suprafața unei case și prețul ei.

Putem folosi modelul liniar pentru predicții pe date noi.

Panta dreptei este pozitivă, înseamnă că avem o corelație pozitivă între suprafață și preț (prețul crește pe măsura ce suprafața crește).

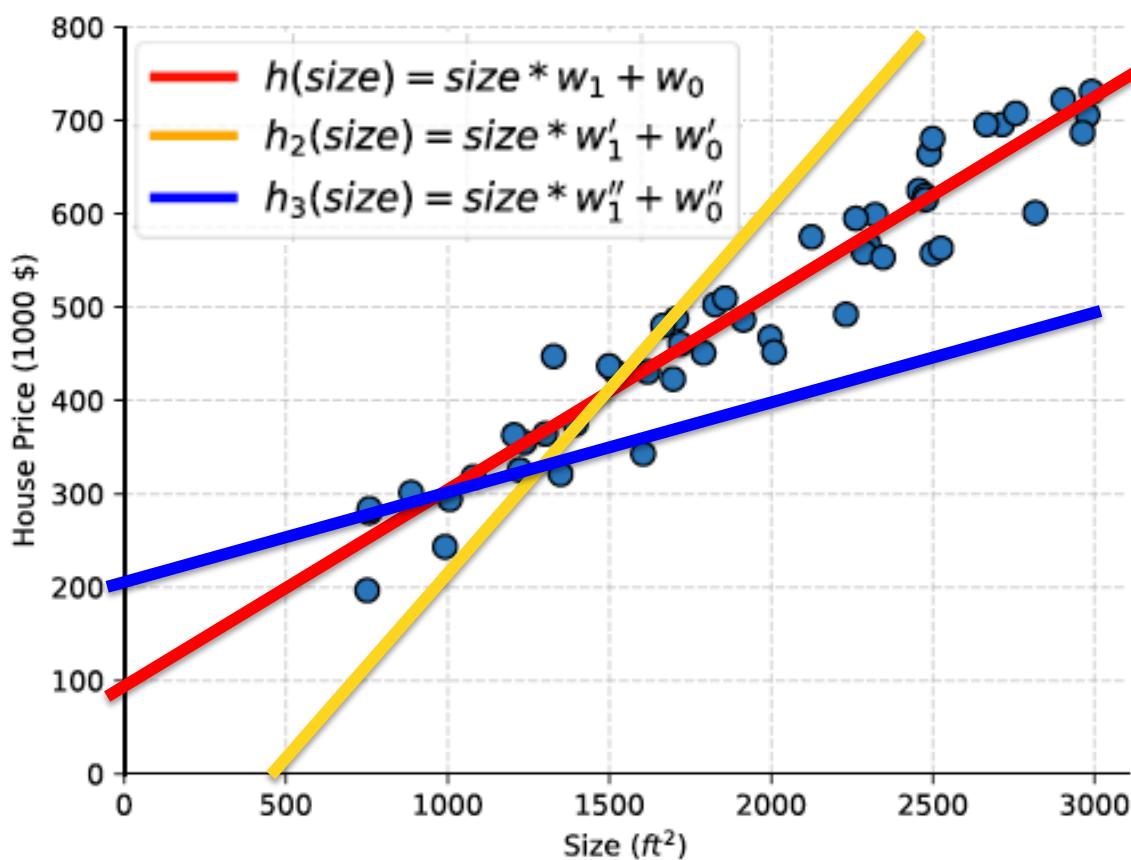
# Regresia liniară simplă

- Relația este modelată de funcție liniară



# Regresia liniară simplă

- Relația este modelată de funcție liniară



Există multe linii posibile candidat.

Avem nevoie de o metodă care găsește cea mai bună linie, adică cele mai bune valori  $w_0$  și  $w_1$ .

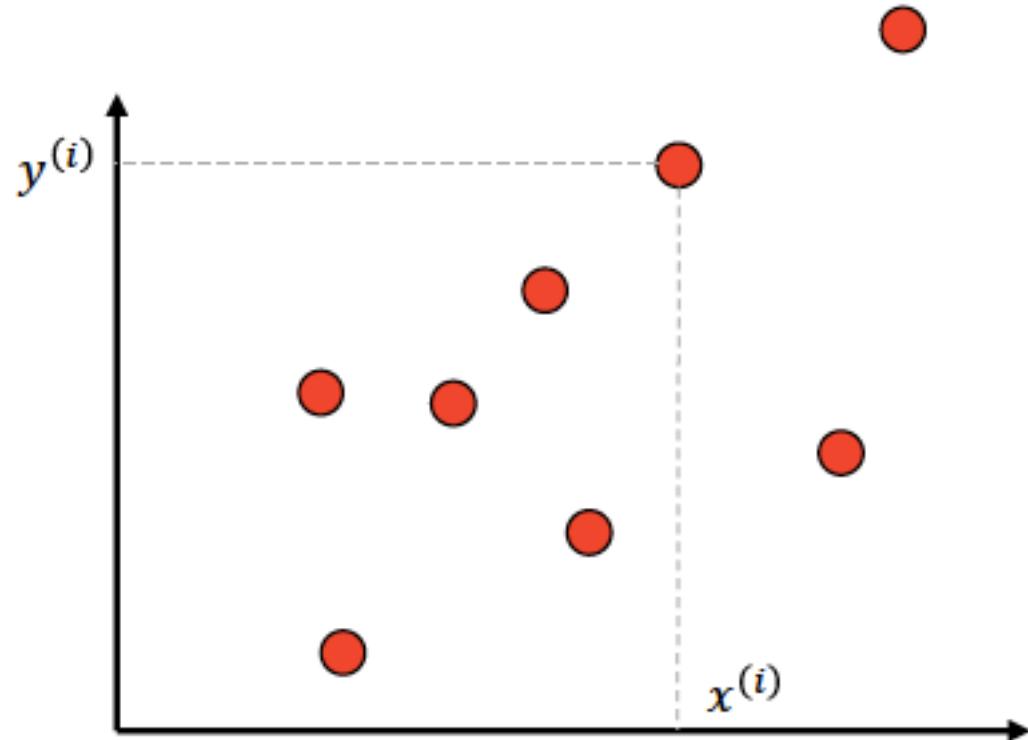
# Notății

- $x \in \mathbb{R}$  este varibila independentă (adică suprafața)
- $y \in \mathbb{R}$  este varibila dependentă (adică prețul)
- $h : \mathbb{R} \rightarrow \mathbb{R}$  este ipoteza (dreapta) pe care o căutăm, are parametri  $w_0$  și  $w_1$
- $\hat{y} = h(x)$  este valoarea prezisă pentru inputul  $x$
- Regresia liniară simplă:  $\hat{y} = h(x) = w_0 + w_1 x$
- Vrem să găsim parametri  $w_0$  și  $w_1$  astfel încât  $\hat{y}$  este cât mai aproape de  $y$

# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

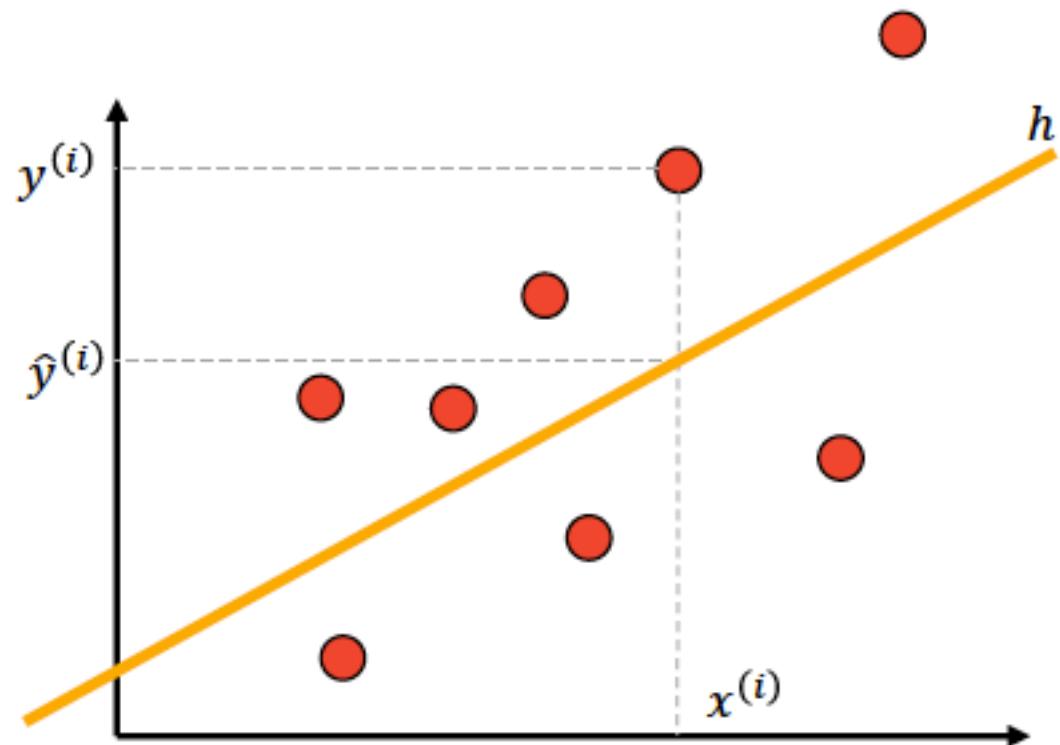
$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$



# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$



# Metoda celor mai mici pătrate

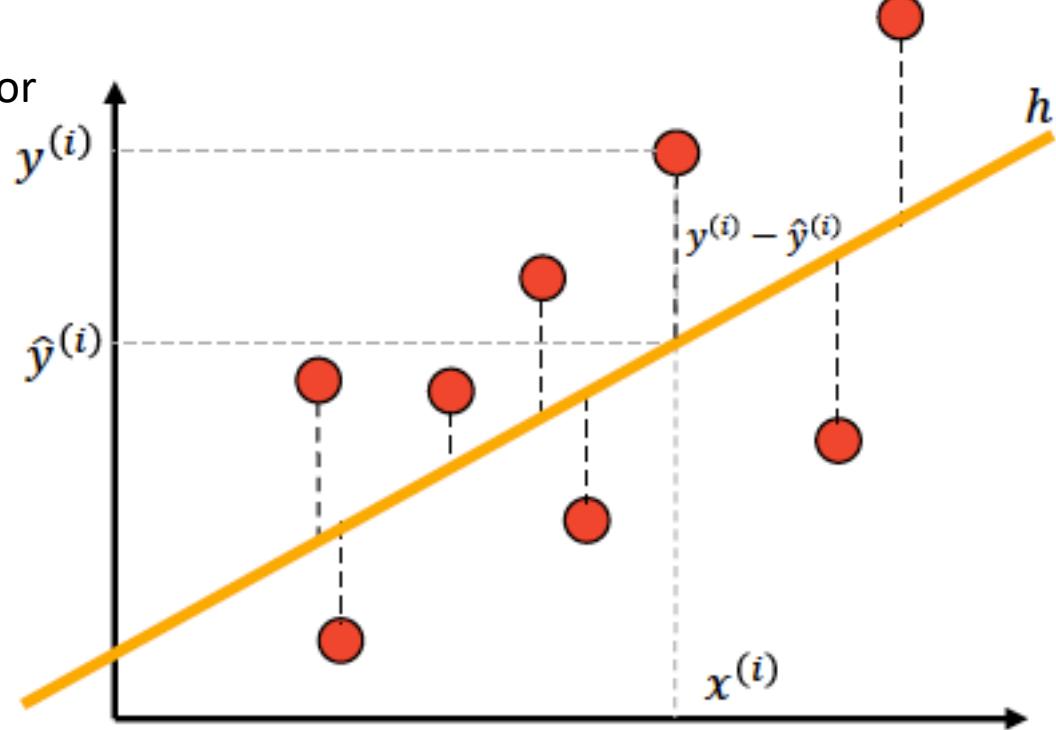
- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

Funcția cost = suma pătratelor erorilor



# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

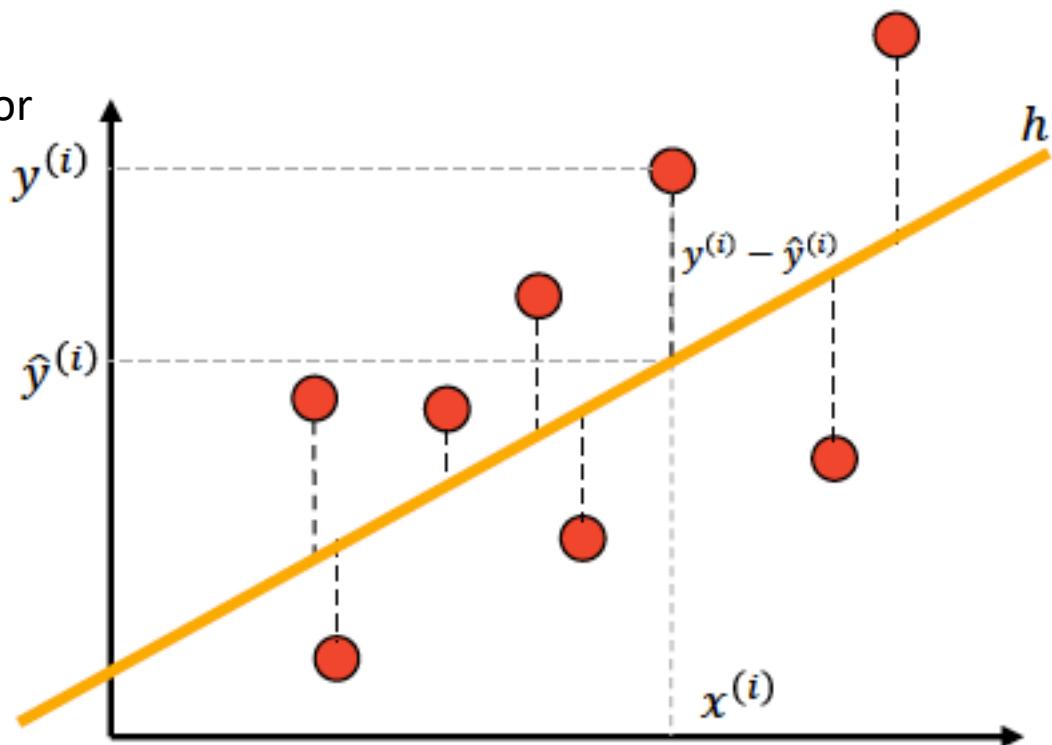
$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

Funcția cost = suma pătratelor erorilor

- Minimizăm  $\mathcal{L}_E$  în raport cu  $w_0$  și  $w_1$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$



# Calcule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

# Calcule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} =$$

# Calcule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) =$$

# Calcule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) = 2 \left( mw_0 - \sum y + w_1 \sum x \right) = 0$$

# Calcule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) = 2 \left( mw_0 - \sum y + w_1 \sum x \right) = 0$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

# Calcule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) =$$

# Calcule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

# Calcule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left( \sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

# Calcule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left( \sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2} =$$

# Calcule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left( \sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2} = \frac{m \sum xy - \sum x \sum y}{m \sum x^2 - (\sum x)^2} = \dots = \frac{\sum [(x - \bar{x})(y - \bar{y})]}{\sum (x - \bar{x})^2}$$

unde

$$\bar{x} = \frac{1}{m} \sum x^{(i)}$$

# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

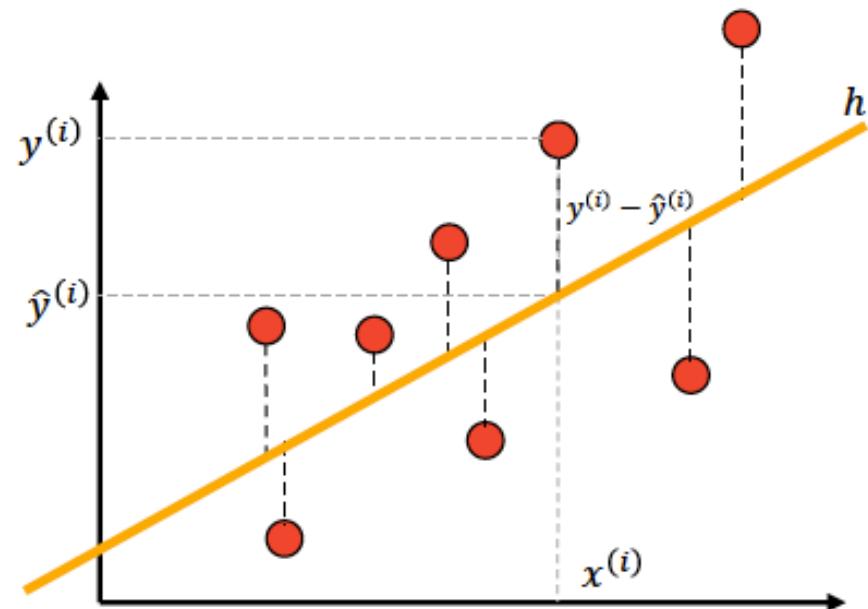
Funcția cost = suma pătratelor erorilor

- Minimizăm  $\mathcal{L}_E$  în raport cu  $w_0$  și  $w_1$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$

$$w_1 = \frac{\sum_i [(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})]}{\sum_i (x^{(i)} - \bar{x})^2}$$

$$\Rightarrow w_0 = \frac{1}{m} \left( \sum_i y^{(i)} - w_1 \sum_i x^{(i)} \right)$$



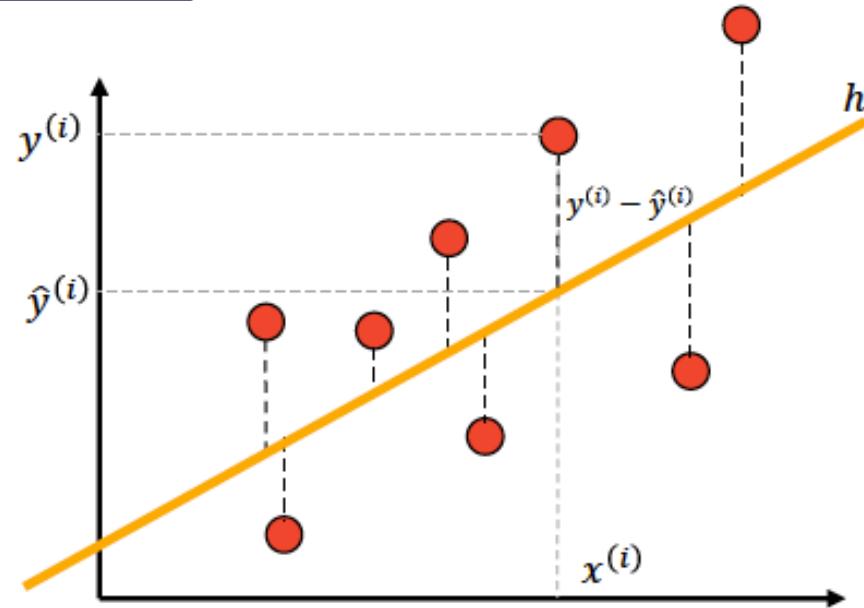
# Metoda celor mai mici pătrate

- Observație: dreapta găsită pe baza celor mai mici pătrate trece prin punctul de coordonate  $(\bar{x}, \bar{y})$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right) \quad \bar{x} = \frac{1}{m} \sum x^{(i)}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$\bar{y} = w_0 + w_1 \bar{x}$$



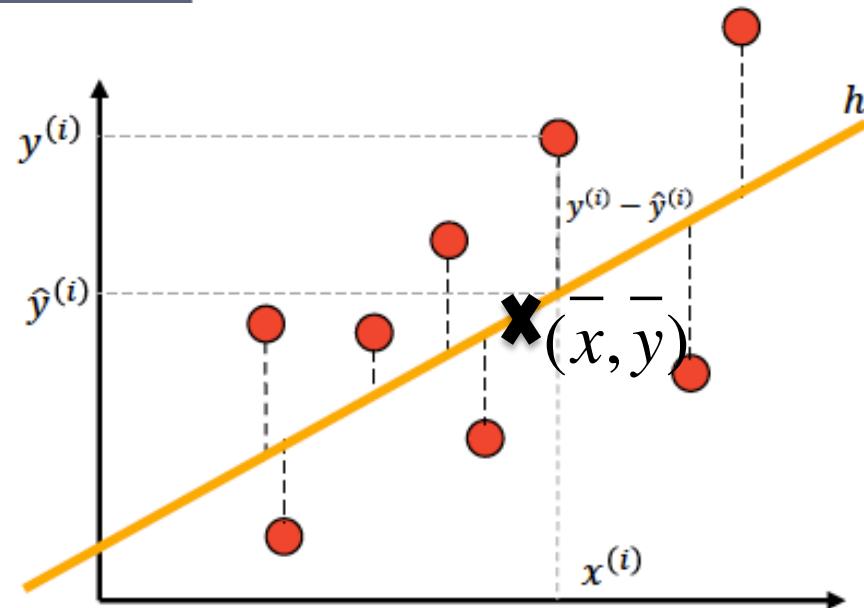
# Metoda celor mai mici pătrate

- Observație: dreapta găsită pe baza celor mai mici pătrate trece prin punctul de coordonate  $(\bar{x}, \bar{y})$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right) \quad \bar{x} = \frac{1}{m} \sum x^{(i)}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$\bar{y} = w_0 + w_1 \bar{x}$$



# Regresia liniară multiplă

- $\vec{x} \in \mathbb{R}^n = [x_1 \quad x_2 \quad \dots \quad x_n],$
- $\vec{w} \in \mathbb{R}^n = [w_1 \quad w_2 \quad \dots \quad w_n], w_0 \in \mathbb{R}$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n = w_0 + \langle \vec{w}, \vec{x} \rangle$$

# Regresia liniară multiplă

- $\vec{x} \in \mathbb{R}^n = [x_1 \ x_2 \ \dots \ x_n]$ ,
- $\vec{w} \in \mathbb{R}^n = [w_1 \ w_2 \ \dots \ w_n], w_0 \in \mathbb{R}$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n = w_0 + \langle \vec{w}, \vec{x} \rangle$$

- Înlocuim termenul liber  $w_0$  (bias-ul) făcând notațiile:

$$\vec{x} \in \mathbb{R}^{n+1} = [1 \ x_1 \ x_2 \ \dots \ x_n]$$

$$\vec{w} \in \mathbb{R}^{n+1} = [w_0 \ w_1 \ w_2 \ \dots \ w_n]$$

$\Rightarrow$

$$\hat{y} = \langle \vec{w}, \vec{x} \rangle$$

# Regresia liniară multiplă

- Avem mulțimea de date de date de antrenare:

$$E = \{(\vec{x}^{(1)}, \vec{y}^{(1)}), (\vec{x}^{(2)}, \vec{y}^{(2)}), \dots, (\vec{x}^{(m)}, \vec{y}^{(m)})\}, \vec{x}^{(i)} \in \mathbb{R}^n, \vec{y}^{(i)} \in \mathbb{R}$$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- Folosim înmulțirea matricelor pentru a calcula predicțiile:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$$

# Regresia liniară multiplă

- Avem mulțimea de date de date de antrenare:

$$E = \{(\vec{x}^{(1)}, \vec{y}^{(1)}), (\vec{x}^{(2)}, \vec{y}^{(2)}), \dots, (\vec{x}^{(m)}, \vec{y}^{(m)})\}, \vec{x}^{(i)} \in \mathbb{R}^n, \vec{y}^{(i)} \in \mathbb{R}$$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- Folosim înmulțirea matricelor pentru a calcula predicțiile:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$$

# Regresia liniară multiplă

- Avem mulțimea de date de date de antrenare:

$$E = \{(\vec{x}^{(1)}, \vec{y}^{(1)}), (\vec{x}^{(2)}, \vec{y}^{(2)}), \dots, (\vec{x}^{(m)}, \vec{y}^{(m)})\}, \vec{x}^{(i)} \in \mathbb{R}^n, \vec{y}^{(i)} \in \mathbb{R}$$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- Folosim înmulțirea matricelor pentru a calcula predicțiile:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} \Rightarrow \widehat{\mathbf{Y}} = \mathbf{X}\mathbf{w}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2 = (Y - \widehat{Y})^T (Y - \widehat{Y}) \stackrel{\text{not}}{=} (Y - \widehat{Y})^2$$

# Regresia liniară multiplă

$$\mathcal{L}_E = (\mathbf{Y} - \hat{\mathbf{Y}})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w})$$

# Regresia liniară multiplă

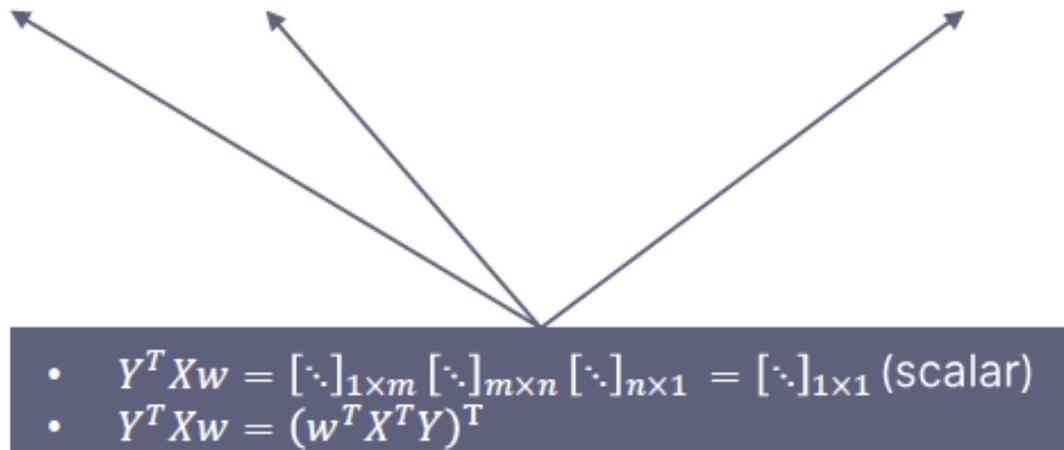
$$\begin{aligned}\mathcal{L}_E &= (\mathbf{Y} - \hat{\mathbf{Y}})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^T (\mathbf{Y} - \mathbf{X}\mathbf{w}) \\ &= (\mathbf{Y}^T - \mathbf{w}^T \mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\mathbf{w})\end{aligned}$$

# Regresia liniară multiplă

$$\begin{aligned}\mathcal{L}_E &= (\mathbf{Y} - \hat{\mathbf{Y}})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\ &= (\mathbf{Y}^T - \mathbf{w}^T \mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X}\mathbf{w} - \mathbf{w}^T \mathbf{X}^T \mathbf{Y} + \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w} = \mathbf{Y}^T \mathbf{Y} - 2\mathbf{w}^T \mathbf{X}^T \mathbf{Y} + \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w}\end{aligned}$$

# Regresia liniară multiplă

$$\begin{aligned}\mathcal{L}_E &= (\mathbf{Y} - \hat{\mathbf{Y}})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^2 = (\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\ &= (\mathbf{Y}^T - \mathbf{w}^T \mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X}\mathbf{w} - \mathbf{w}^T \mathbf{X}^T \mathbf{Y} + \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w} = \mathbf{Y}^T \mathbf{Y} - 2\mathbf{w}^T \mathbf{X}^T \mathbf{Y} + \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w}\end{aligned}$$



Un scalar (un număr) este o matrice  $1 \times 1$

Transpusa unui scalar este același scalar

# Regresia liniară multiplă

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimizăm  $\mathcal{L}_E$  în raport cu  $w = (w_0, w_1, \dots, w_n)$ , toate derivatele partiale sunt 0:

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0 \Rightarrow -2X^T Y + 2X^T X w = 0 \Rightarrow X^T X w = X^T Y \Rightarrow$$

$$w = (X^T X)^{-1} X^T Y$$

- Matricea  $X^T X$  trebuie să fie inversabilă

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 8

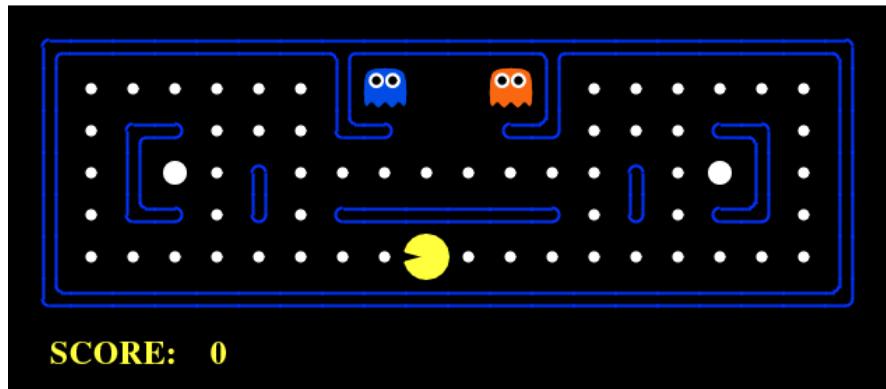
# Recapitulare – cursul trecut

1. Rețele feedforward multistrat de perceptri
2. Regresia liniară simplă și multiplă

# Cuprinsul cursului de azi

1. Agenți inteligenți și mediile în care aceștia funcționează
2. Rezolvarea problemelor prin căutare
  - Graful stărilor
  - Arborele de căutare

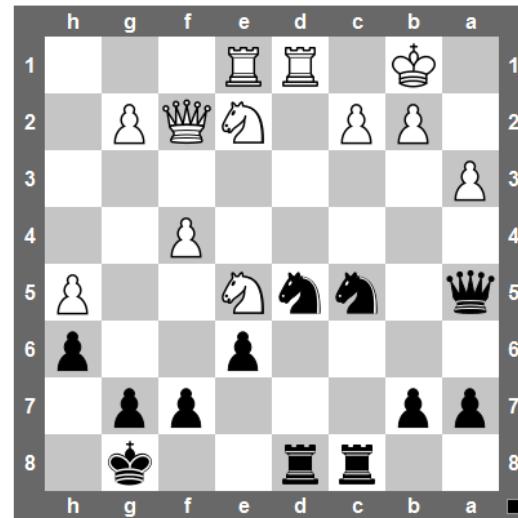
# Agenti inteligenți și mediile în care aceștia funcționează



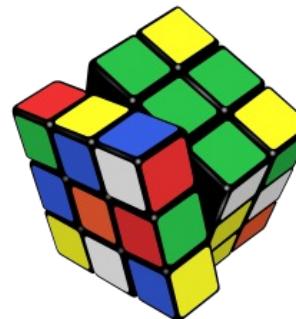
1	2	3
8		4
7	6	5

# Căutarea

- căutare = tehnică de rezolvare a problemelor ce explorează în mod sistematic un spațiu de stări ale problemei
- exemplu de stare a unei probleme: configurația tablei de sah, a unui cub Rubik, a unui careu Sudoku, a unui puzzle



				8	3		5	6
3		7	5		9			
8		3	1					
		9	2					
2	5							9
6	7	1						
9		2	6					
	5							

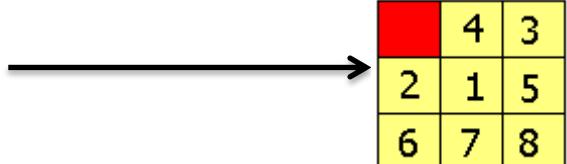


1	2	3
4		5
6	7	8

# Căutarea

- spațiu de căutare = graf (arbore) în care un nod desemnează o soluție parțială iar o muchie reprezintă un pas în construcția unei soluții
- scopul căutării: a găsi un drum în graf de la un *nod-inițial* (configurație inițială) la un *nod-scop* (configurație finală)
- un program reprezintă un agent intelligent; agenții cu care vom lucra vor adopta un *scop* și vor urmări *satisfacerea* lui.

1	2	3
4		5
6	7	8

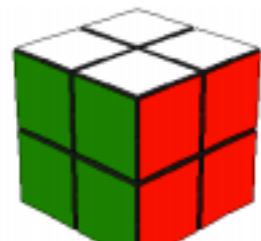


Nod inițial

Nod scop



Nod inițial



Nod scop

# A doua parte a semestrului: curs

1. Rezolvarea problemelor prin căutare

Stare

Scop

Funcție succesor

Acțiuni

Spațiul stărilor

Arbore de căutare

Căutare neinformată

Căutare informată

2. Strategii de căutare neinformată

Căutare în lățime

Căutare în adâncime

Căutare cu cost uniform

Căutare cu adâncime limitată

Căutare cu adâncime incrementală

3. Strategii de căutare informată

Euristică

Greedy

A\*

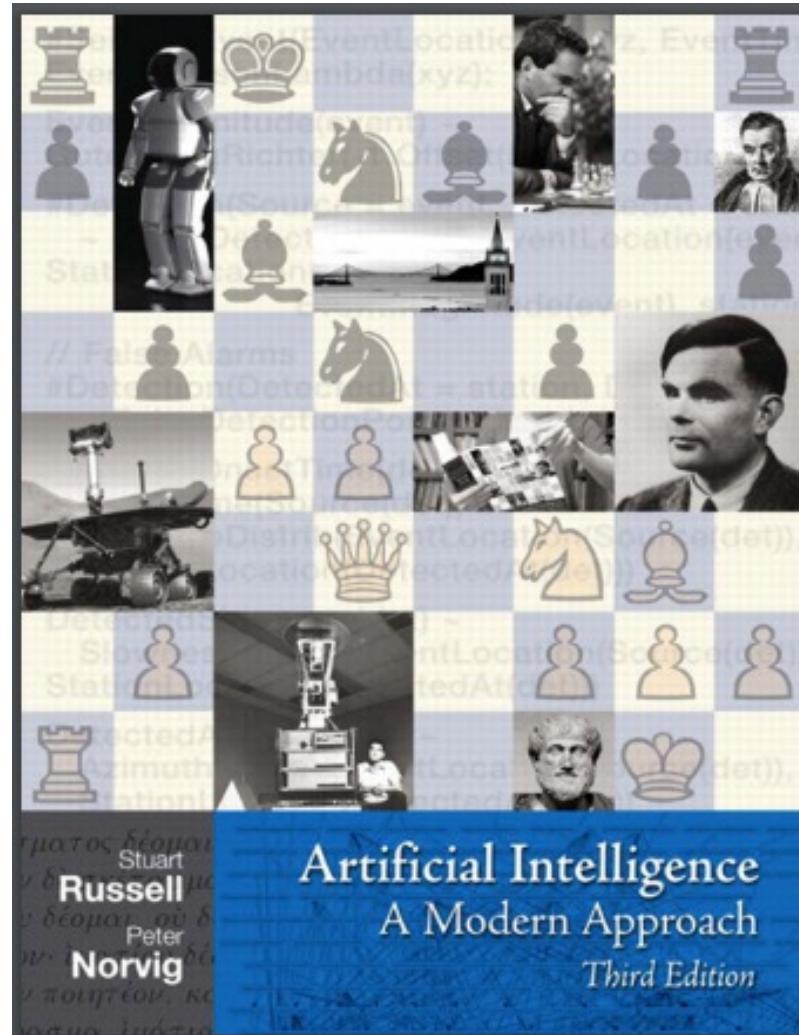
Euristică admisibilă

4. Strategii de căutare adversariale

Algoritmul minimax

Alpha-Beta retezare

# Bibliografie

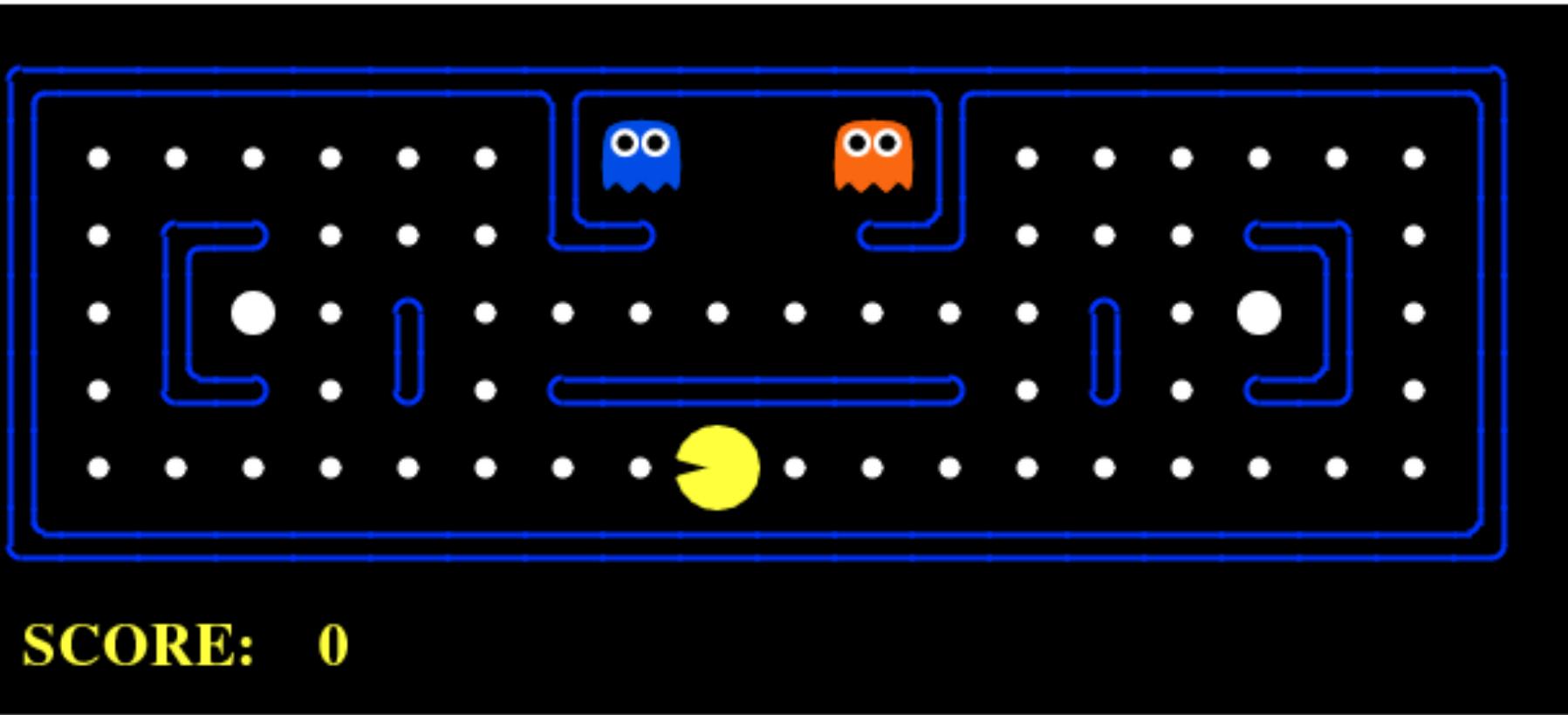


Prima și a doua parte

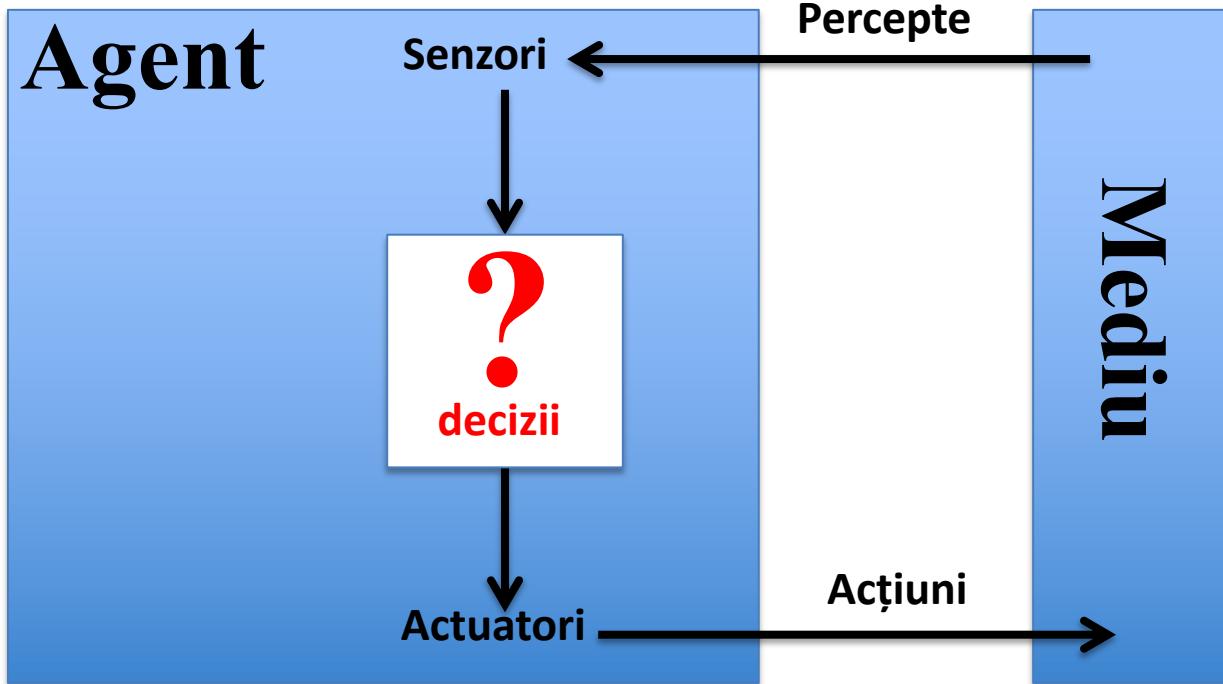
# Agenti inteligenți și mediile în care aceștia funcționează



Agenti inteligenți și mediile în care aceștia funcționează



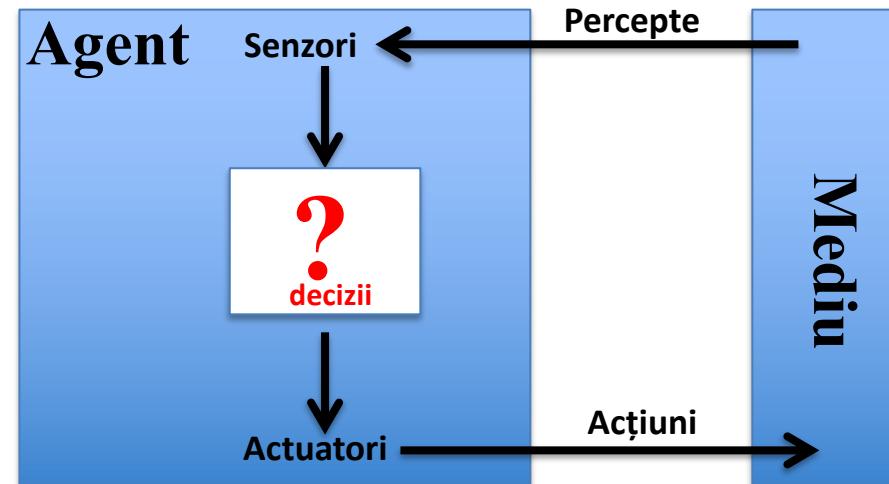
# Agenti inteligenți și mediile în care aceștia funcționează



- Un agent este o entitate care **percepe** și **acționează**.
- **Percepția** – input-urile perceptuale la fiecare moment de timp.
- **Secvența de percepții** – istoria completă a tot ceea ce a fost perceput de către agent

# Agenti inteligenți și mediile în care aceștia funcționează

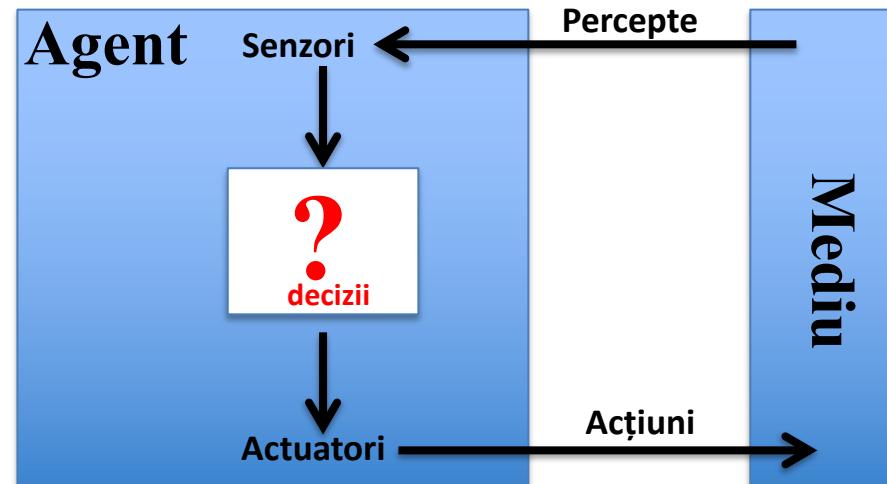
- Scopul celei de-a doua părți a cursului: studiul unui set de principii pentru construirea unor agenti **raționali**.
- Exemplu de agenți: om, robot, mașină autonomă, program software
- Acțiunile unui agent depind în totalitate numai de ce percepse (input-ul) primește agentul din mediul în care funcționează prin intermediul senzorilor. Comportamentul unui agent este modelat de o funcție  $f$  care transformă cunoașterea agentului (orice secvență de percepțe) în acțiuni



$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

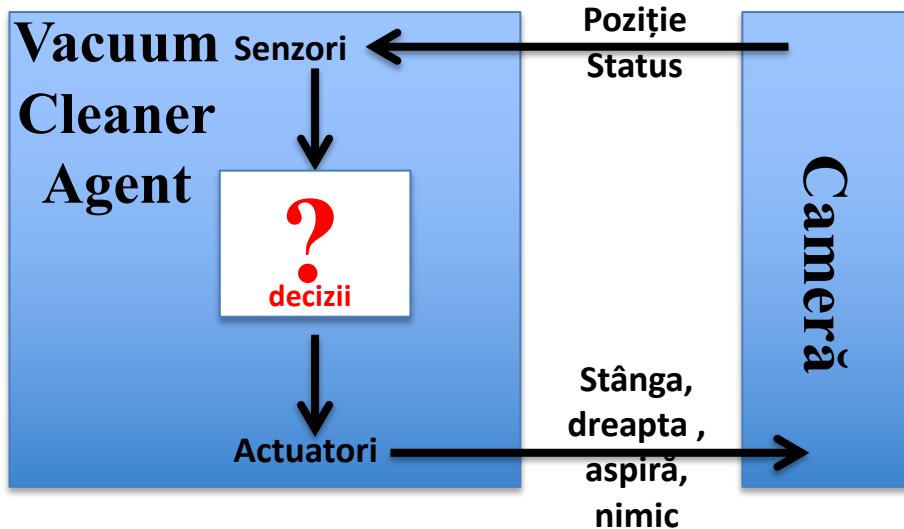
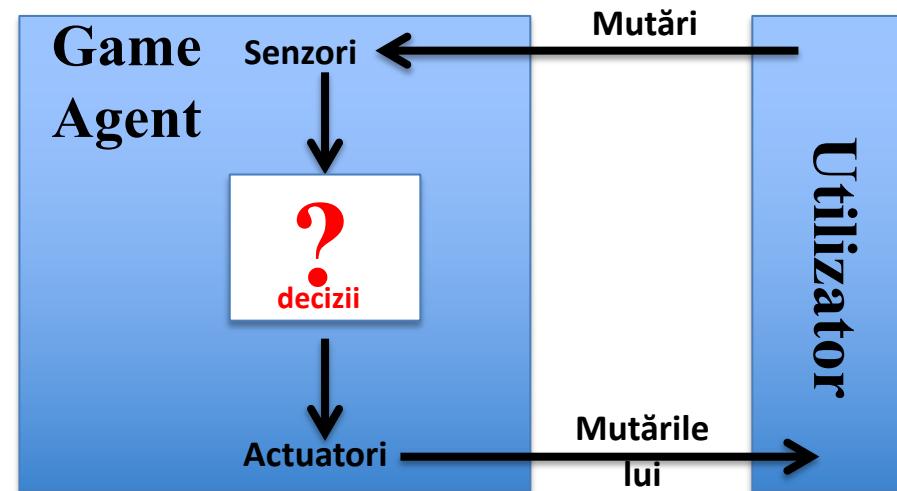
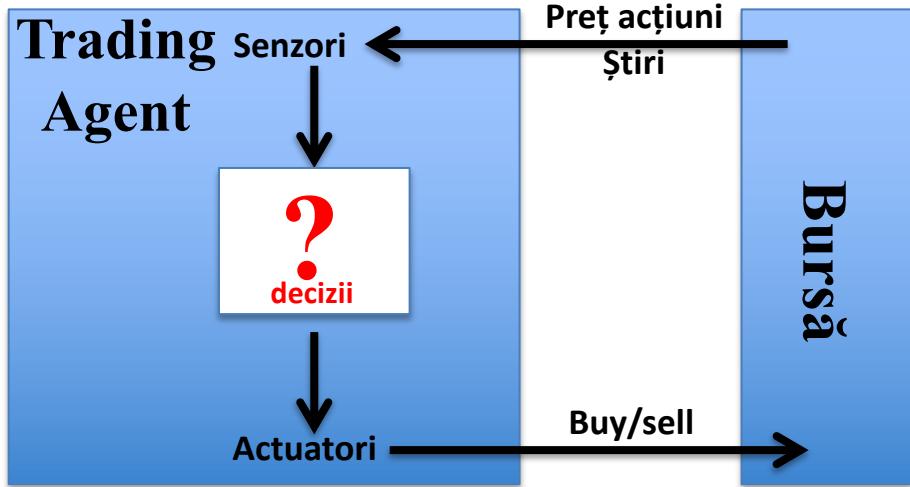
# Agenti inteligenți și mediile în care aceștia funcționează

- Scopul celei de-a doua părți a cursului: studiul unui set de principii pentru construirea unor agenti **rationali**.
- Exemplu de agenți: om, robot, mașină autonomă, program software
- Vrem să găsim funcția **f** prin care agentul ia decizii
- Mediu → Senzori → Decizii → Actuatori

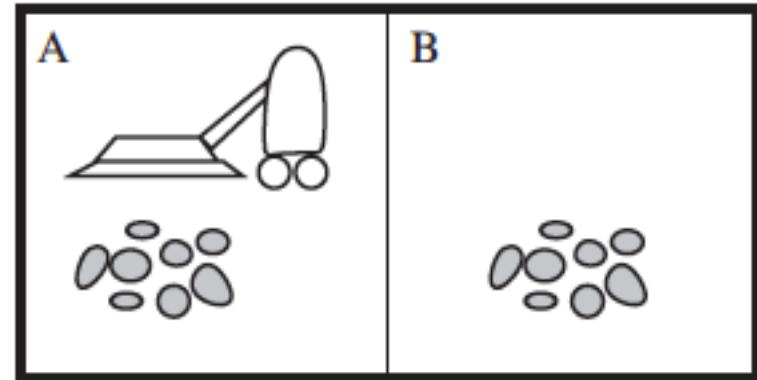
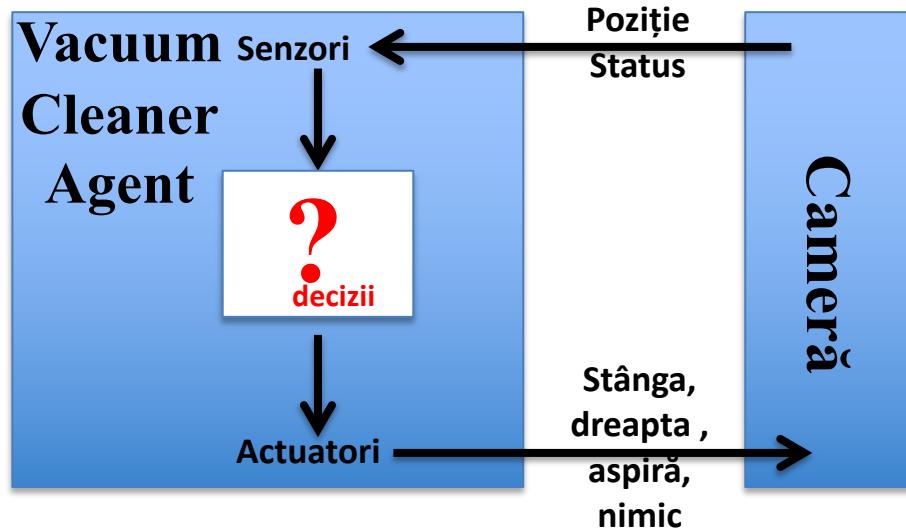


*Ciclul percepțe – acțiuni*

# Agenti inteligenți și mediile în care aceștia funcționează



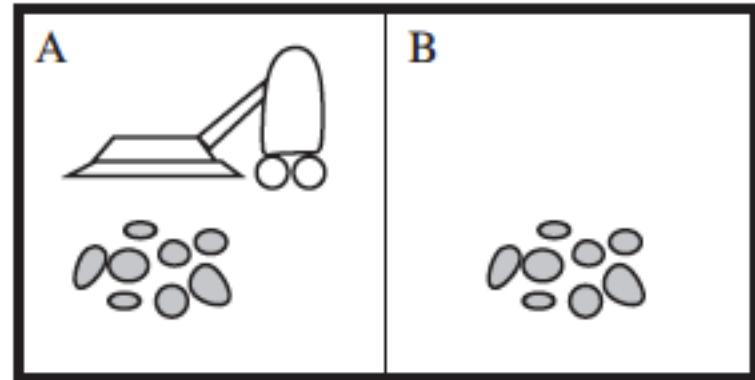
# Un agent aspirator



- Aspiratorul percepce poziția (A sau B) și starea (curat sau murdar)
- Acțiuni: mergi în stânga, mergi în dreapta, aspiră, nimic.
- Consideră următoarea funcție f:  $P^* \rightarrow A$ , dacă poziția curentă este murdară atunci aspiră, altfel mută-te la cealaltă poziție
- Putem tabela funcția f

# Un agent aspirator

Secvența de percepție	Acțiuni
[A, curat]	Dreapta
[A, murdar]	Aspiră
[B, curat]	Stanga
[B, murdar]	Aspiră
[A, curat], [B, curat]	Stanga
[A, curat], [B, murdar]	Aspiră
...	...



- Aspiratorul percepție poziția (A sau B) și starea (curat sau murdar)
- Acțiuni: mergi în stânga, mergi în dreapta, aspiră, nimic.
- Consideră următoarea funcție f:  $P^* \rightarrow A$ , dacă poziția curentă este murdară atunci aspiră, altfel mută-te la cealaltă poziție
- Putem tabela funcția f

# Proprietăți ale mediilor în care funcționează agenții

- **Complet observabil vs parțial observabil:** la orice moment în timp mediul este complet observabil dacă senzorii agentului îi dau acces complet la starea mediului la fiecare moment și îi oferă toate aspectele relevante în luarea deciziei optime.
  - săh vs poker
- **Determinist vs stochastic:** dacă următoarea stare a mediului este în întregime determinată de starea curentă și de acțiunile agentului, mediul este determinist.
  - săh vs poker

# Proprietăți ale mediilor în care funcționează agenții

- **Discret vs continuu:** dacă numărul de percepții și de acțiuni ale agentului sunt finite atunci mediul este discret.
  - șah vs condusul unei mașini
- **Episodic vs sevențial:** într-un mediu episodic, experiența agentului este împărțită în episoade independente. Un episod constă în percepție urmată de o singură acțiune. Episoadele următoare nu depind de acțiunile episoadelor anterioare.
  - corectarea unui test grilă vs șah

# Proprietăți ale mediilor în care funcționează agenții

- **Static vs dinamic:** dacă mediul se schimbă cât timp agentul decide ce acțiuni să facă mediul este dinamic.
  - rezolvarea unui rebus vs condusul unei mașini
- **Agent vs multiagent:** dacă sunt mai mulți agenți mediul este multiagent (sisteme multiagent competitive vs sisteme multiagent cooperative).
  - corectarea unui test grilă vs șah

# Jocul de dame



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs sevențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Determinist

Discret

Sevențial

Static

Multiagent competitiv

# Jocul de poker



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs sevențial

Static vs dinamic

Agent vs multiagent

Parțial observabil

Stochastic

Discret

Sevențial

Static

Multiagent competitiv

# 8 - puzzle

1	2	3
8		4
7	6	5

Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs sevențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Determinist

Discret

Sevențial

Static

Agent

# Mașină autonomă



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Parțial observabil

Stochastic

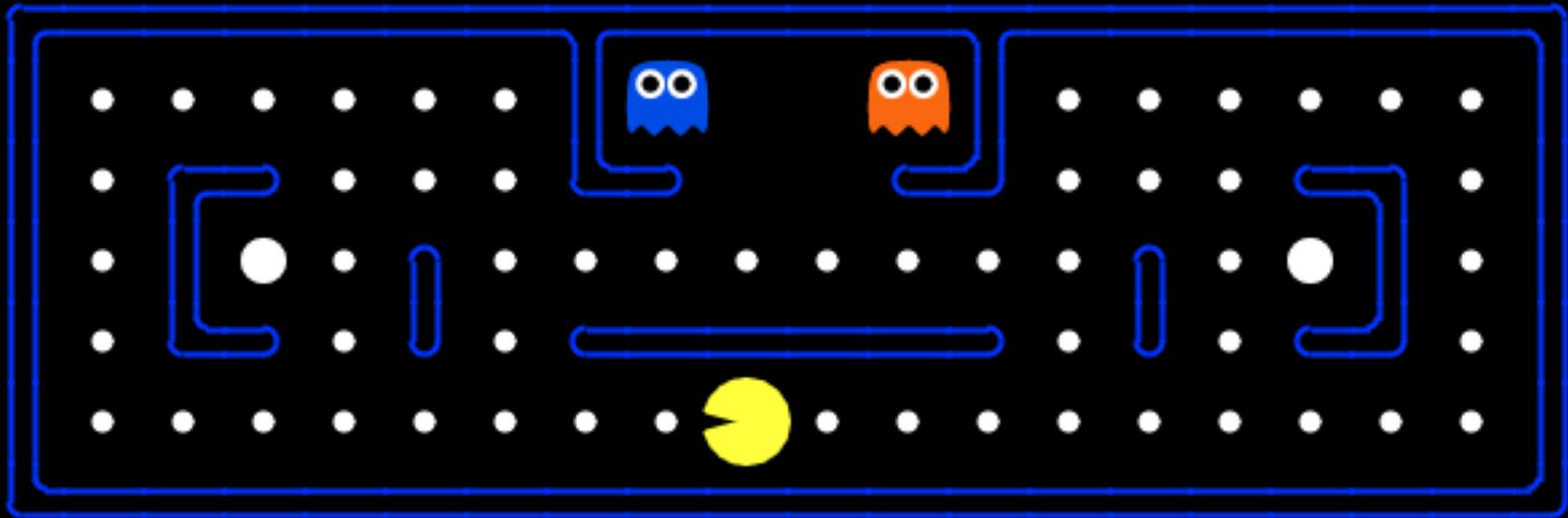
Continuu

Secvențial

Dinamic

Multiagent cooperativ

# PAC MAN

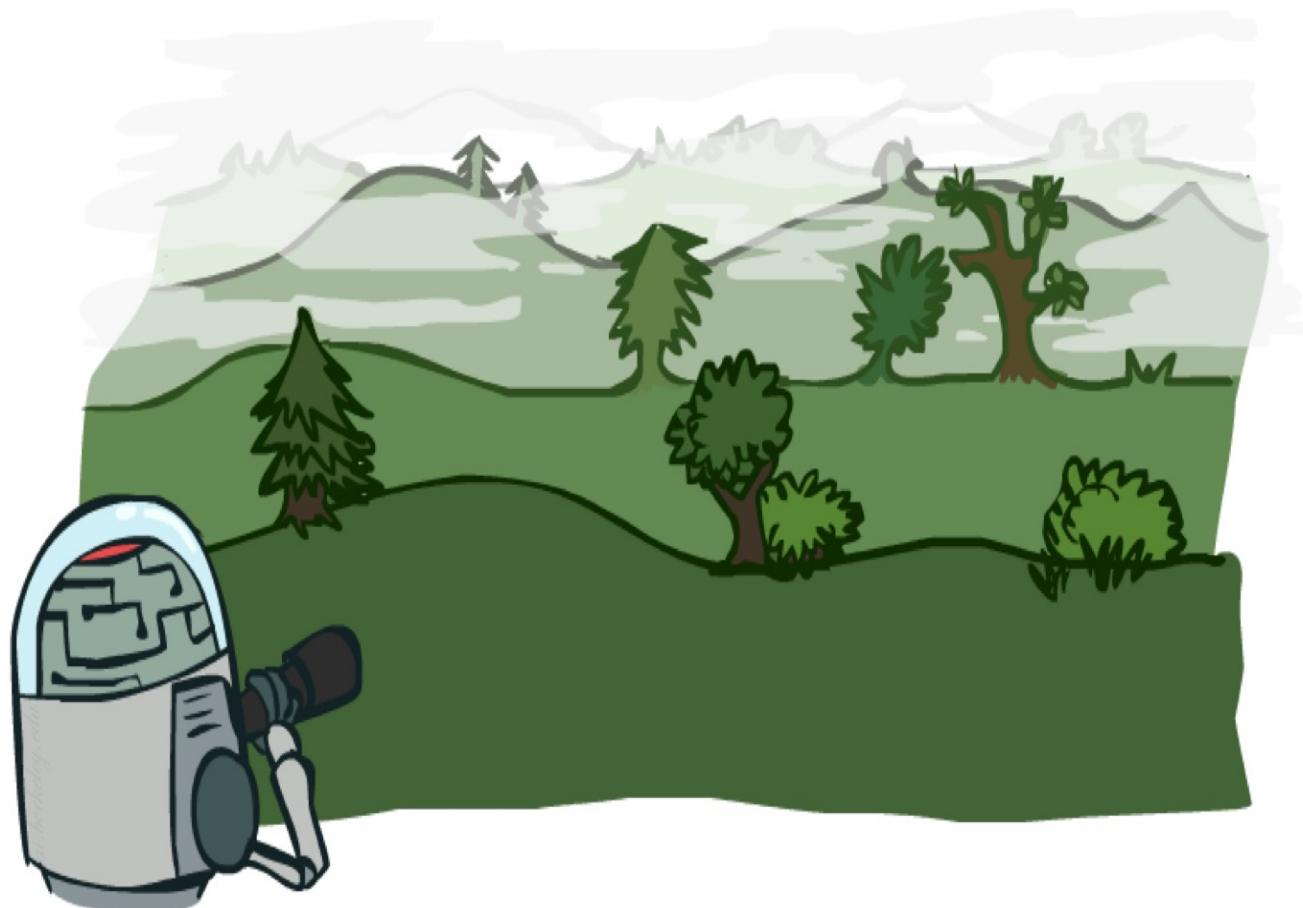


**SCORE:** 0

Complet observabil vs parțial observabil  
Determinist vs stochastic  
Discret vs continuu  
Episodic vs sevențial  
Static vs dinamic  
Agent vs multiagent

Complet observabil  
Stochastic  
Discret  
Sevențial  
Dinamic  
Multiagent competitiv

# Probleme de căutare



# Rezolvarea problemelor prin căutare

- primul pas: formularea scopului
  - un scop este o multime de stări ale universului în care vrem să ajungem
- al doilea pas: formularea problemei
  - ce acțiuni (operatori) poate face agentul dintr-o anumită stare
  - stare = configurație a lumii înconjurătoare (a universului problemei)
  - acțiunile generează tranziția între stări ale universului
  - o acțiune se poate defini printr-o funcție successor:  $S(x) = \text{multimea stărilor în care se poate ajunge din } x \text{ printr-o unică acțiune}$
  - ce stări trebuie luate în considerare
- algoritm de căutare
  - primește ca input o problemă
  - întoarce ca output o soluție sub forma unei succesiuni de acțiuni

# Formularea unei probleme

- Spațiul stărilor
  - stare = configurație a lumii înconjurătoare
  - mulțimea stărilor în care se poate ajunge plecând din starea inițială, prin intermediul oricărei secvențe de acțiuni
  - ce stări trebuie luate în considerare
  - stare inițială
  - stare scop
- Acțiuni
  - ce acțiuni (operatori) poate face agentul dintr-o anumită stare
  - acțiunile generează tranziția între stări ale universului
  - uneori acțiunile au un cost asociat
- Funcția successor
  - $SUCCESSOR(x, a) = \text{mulțimea stărilor în care se poate ajunge din starea } x \text{ prin acțiunea } a$

# Problema 8-puzzle

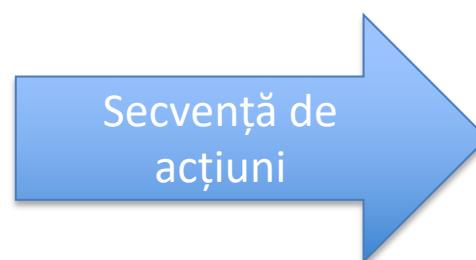
## Descrierea problemei

Pe o tablă  $3 \times 3$  se găsesc 8 piese, numerotate de la 1 la 8. La fiecare moment o singură piesă se poate mișca cu o poziție, pe orizontală sau verticală, în limitele cadrului tablei, în locul rămas liber (poziția roșie). Se dă o configurație inițială și una finală a tablei. Trebuie să se găsească secvența de mutări care să aducă piesele din configurația inițială în cea finală.

2		3
1	8	4
7	6	5

Stare inițială

[2, X, 3, 1, 8, 4, 7, 6, 5]



1	2	3
8		4
7	6	5

Stare scop

[1, 2, 3, 8, X, 4, 7, 6, 5]

# Formularea problemei 8-puzzle

- Spatiul stărilor

- stare = configurație a pătratului  $3 \times 3$
- $9!/2$  stări posibile
- stare inițială
- stare scop

2		3
1	8	4
7	6	5

Stare inițială

1	2	3
8		4
7	6	5

Stare scop

[2, X, 3, 1, 8, 4, 7, 6, 5]

[1, 2, 3, 8, X, 4, 7, 6, 5]

- Acțiuni

- STÂNGA, JOS, DREAPTA, SUS
- fiecare acțiune (mutare) are cost = 1

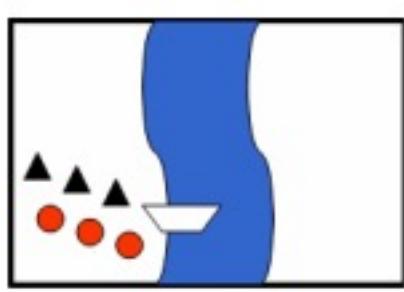
- Funcția succesor

- SUCCESOR([X, a, b , c, d, e, f ,g , h], STANGA) = NU EXISTA
- SUCCESOR([X, a, b , c, d, e, f ,g , h], SUS) = NU EXISTA
- SUCCESOR([X, a, b , c, d, e, f ,g , h], DREAPTA) = [a, X, b , c, d, e, f ,g , h]
- SUCCESOR([X, a, b , c, d, e, f ,g , h], JOS) = [c, a, b , X, d, e, f ,g , h]
- ...

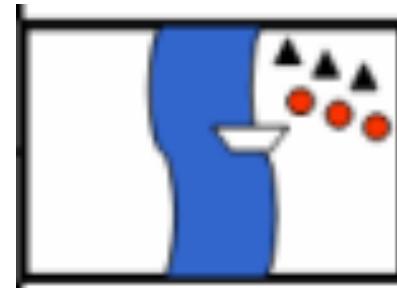
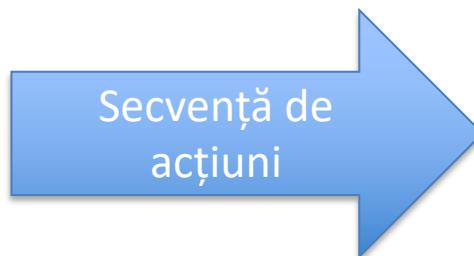
# Problema misionarilor și a canibalilor

## Descrierea problemei

Trei misionari și **trei canibali** se află la marginea unui râu, cu scopul de a trece pe celălalt mal. Ei au la dispoziție o barcă de două persoane. Dacă la un moment dat, pe un mal sau pe celălalt, numărul canibalilor este mai mare decât cel al misionarilor, canibalii îi vor mâncă pe misionari. Problema constă în a afla cum pot trece cele 6 persoane în deplină siguranță de pe un mal pe celălalt ☺. Barca nu merge singură, este nevoie de cel puțin o persoană în barcă.



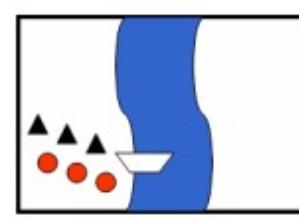
Stare inițială



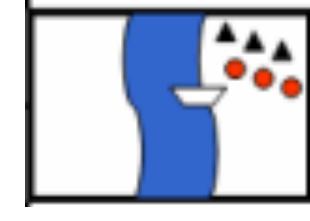
Stare scop

# Formularea problemei misionarilor și a canibalilor

- Spațiul stărilor
  - triplet  $(m, c, b)$ :  $m$  = numărul de misionari pe malul stâng,  $c$  = numărul de canibali pe malul stâng,  $b$  = prezența bărcii (0 sau 1) pe malul stâng
  - stare inițială:  $(3,3,1)$
  - stare scop:  $(0,0,0)$
- Acțiuni
  - pot duce de pe un mal către celălalt mal 1 sau 2 misionari, 1 sau 2 canibali, 1 misionar + 1 canibal ( $1m, 2m, 1c, 2c, 1m+1c$ )
- Funcția succesor
  - $\text{Succesor}((m, c, b), 1c) = (m, c + 1 - 2b, 1 - b), \dots$
  - $\text{Succesor}((m, c, b), 1m) = (m + 1 - 2b, c, 1 - b), \dots$
  - unele stări sunt invalide  $(2,3,0)$  sau ilegale  $(0,0,1)$



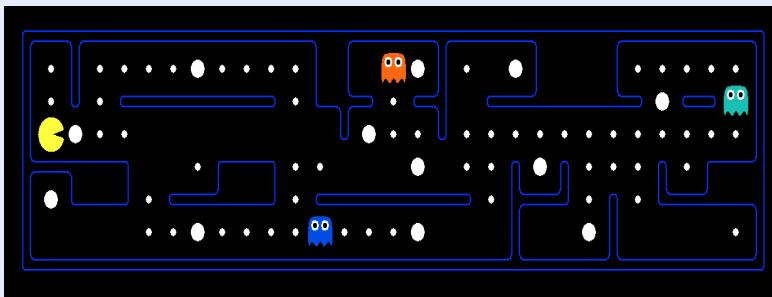
Stare inițială



Stare scop

# Ce conține spațiul stărilor?

O **stare a lumii** include fiecare detaliu al mediului

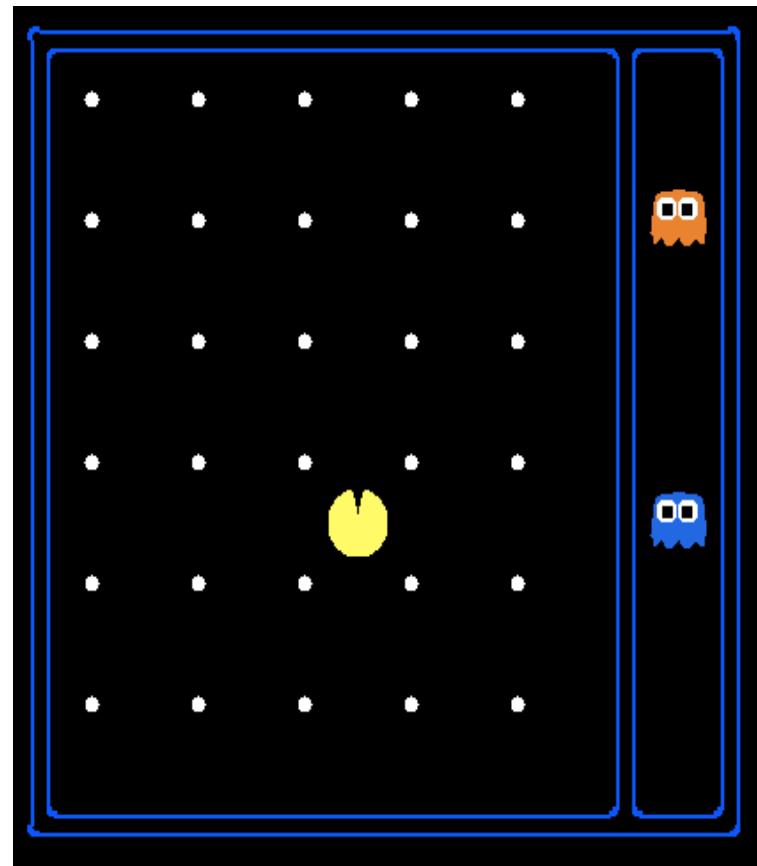


O **stare pentru căutare** păstrează numai detaliile relevante planificării (abstractizare)

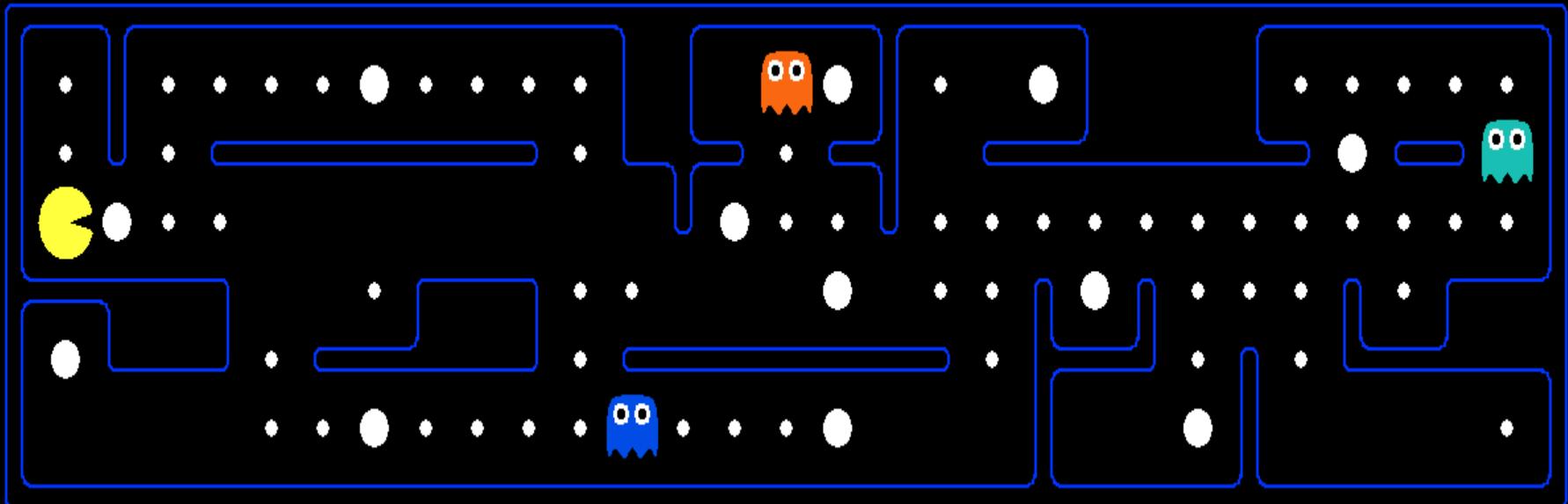
- Problema: vreau să ajung într-un anumit punct (Pathing)
  - stări: poziția (x,y)
  - acțiuni: Nord Sud Est Vest
  - funcție succesor: actualizează numai poziția
  - test scop:  $(x,y)=\text{END}$  ?
- **Problemă: Mănâncă toate punctele albe (Eat-all-dots)**
  - stări:  $\{(x,y), \text{var. booleană pentru puncte albe}\}$
  - acțiuni: Nord Sud Est Vest
  - succesor: actualizează poziția + posibil variabila booleană
  - test scop: toate variabilele booleene sunt false

# Dimensiunea spațiului stărilor

- Stările lumii:
  - pozițiile agentului: 120
  - numărul de puncte albe: 30
  - poziții fantome: 12
  - orientarea agentului: NSEW
- Câte:
  - stări ale lumii?  
 $120 \times (2^{30}) \times (12^2) \times 4$
  - stări pentru problema Pathing?  
120
  - stări pentru problema Eat-all-dots?  
 $120 \times (2^{30})$



# Safe Passage Problem

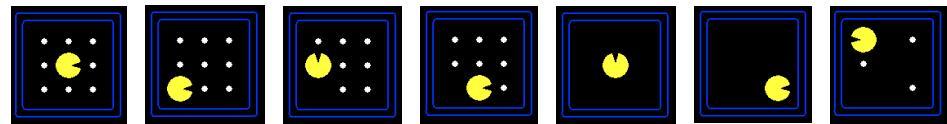


- Problemă: mănâncă toate punctele iar în tot acest timp fantomele trebuie să fie speriate
- **Ce trebuie să specificăm pentru o stare?**
  - (poziția agentului, var. booleană pentru puncte albe, var. booleană pentru putere puncte albe, timp rămas ca fantomele să fie speriate)

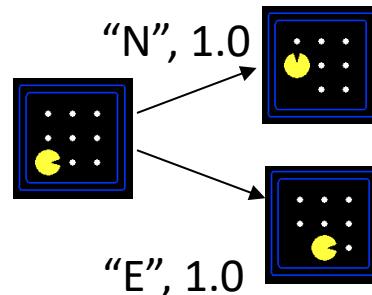
# Probleme de căutare - summar

- O problemă de căutare cuprinde:

- un spațiu al stărilor



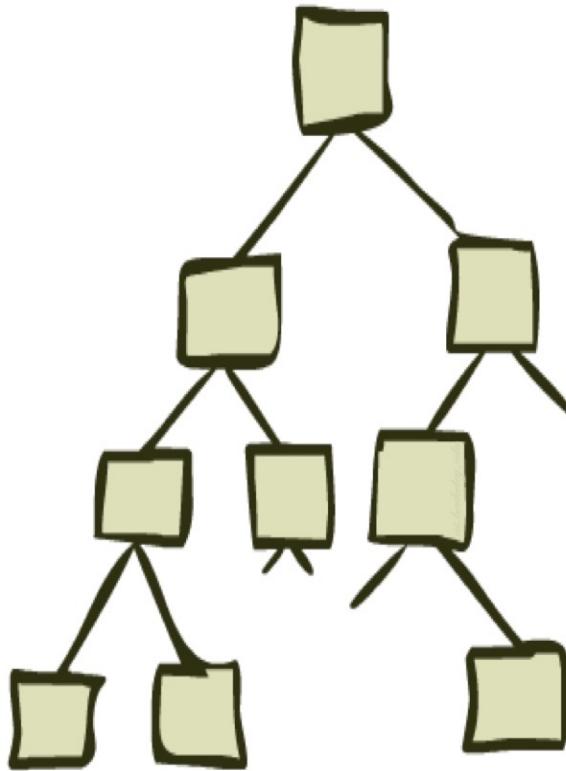
- o funcție successor  
(cu acțiuni, costuri)



- o stare inițială și un scop care poate fi verificat

- o soluție este o secvență de acțiuni (un plan) care transformă starea inițială într-o stare scop

# Graful de stări și arborele de căutare

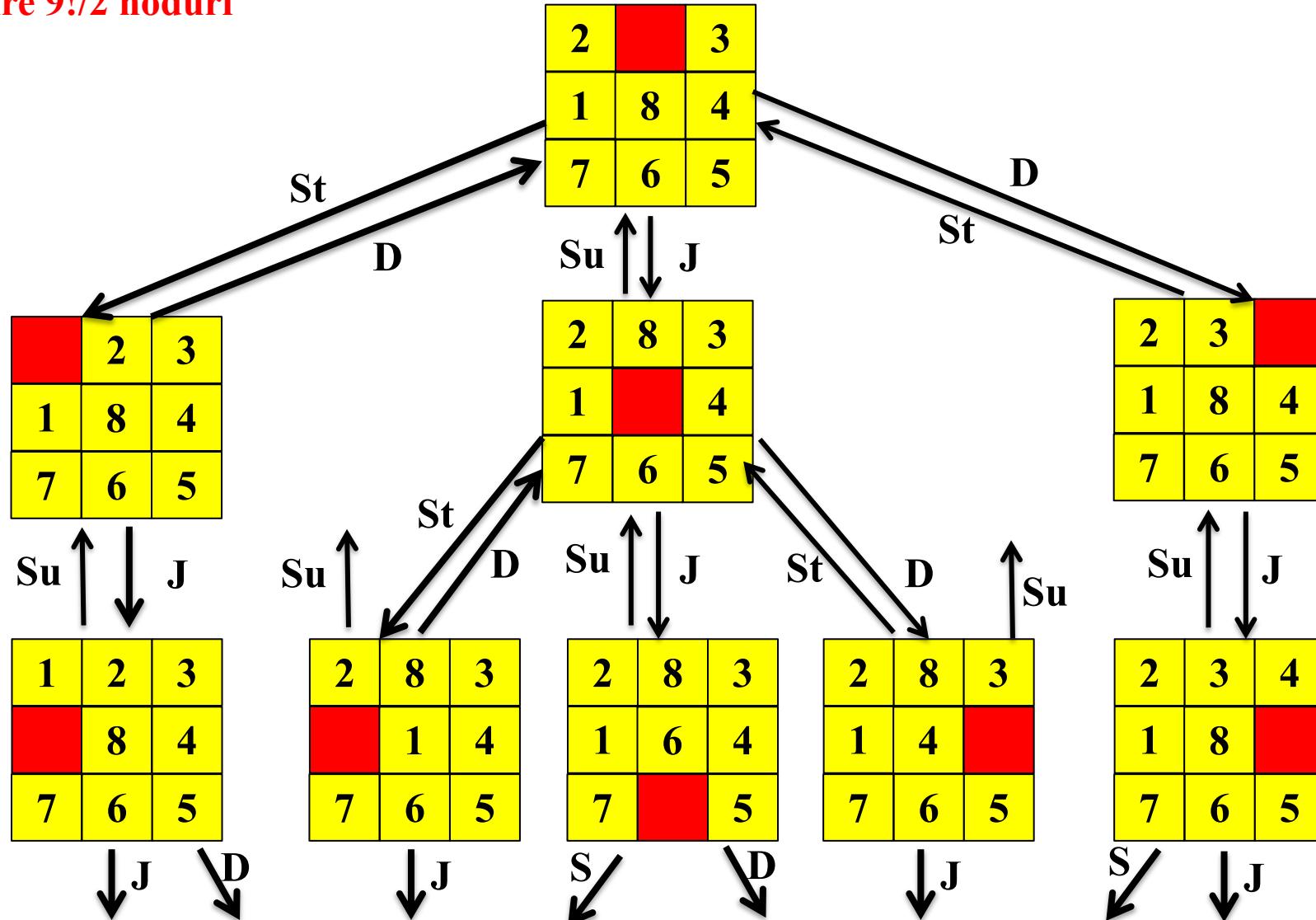


# Graful de stări al unei probleme

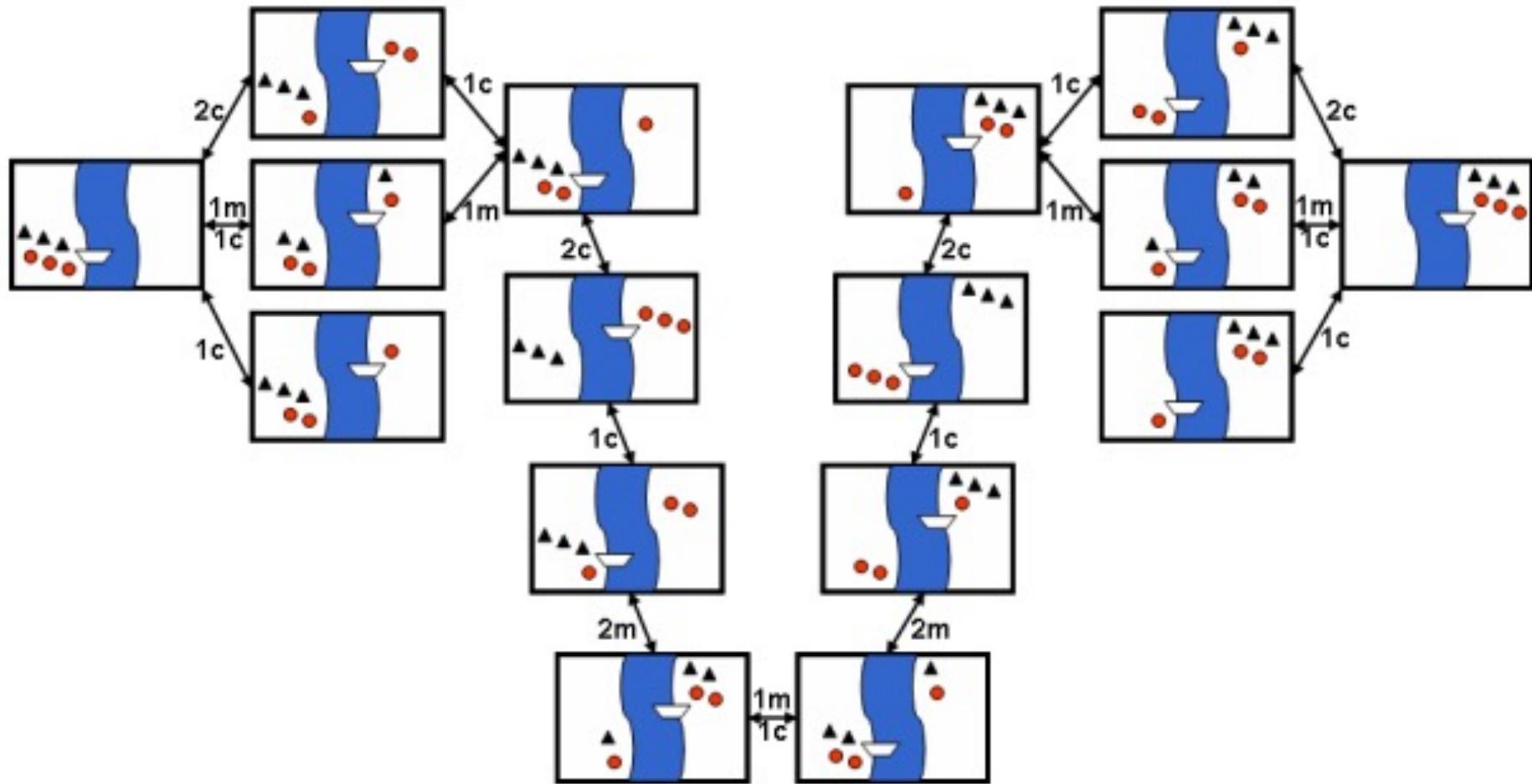
- Graf orientat care conține toate stările lumii (o stare = un nod)
- Fiecare stare apare o singură dată
- Arcele reprezintă acțiuni
  - modelează funcția successor
  - în ce stări putem ajunge din starea curentă realizând o anumită acțiune
  - deseori avem arc de la starea A la starea B și invers (acțiuni simetrice)

# Graful de stări al problemei 8 - puzzle

Are  $9!/2$  noduri



# Graful de stări al problemei misionarilor și a canibalilor



# Arborele de căutare al unei probleme

- Tot un graf orientat, dar organizat pe nivele:
  - nivelul 0 = rădăcina (starea inițială)
  - nivelul i = nod obținut prin traversarea a i arce de la rădăcină din graful de stări
- Un nod desemnează o soluție parțială
- Un arc reprezintă un pas în construirea unei soluții
- Nodurile se pot repeta (spre deosebire de graful stărilor)
  - pot ajunge într-o stare urmând drumuri diferite
- Vrem să găsim un drum (= soluție) în graful de stări de la starea inițială la o stare scop (stare finală).

# Arborele de căutare al problemei 8-puzzle

Stare inițială		
2		3
1	8	4
7	6	5

Stânga  
Jos  
Dreapta  
Sus

# Arborele de căutare al problemei 8-puzzle

Stare inițială

2		3
1	8	4
7	6	5

Stânga  
Jos  
Dreapta  
Sus

St

D

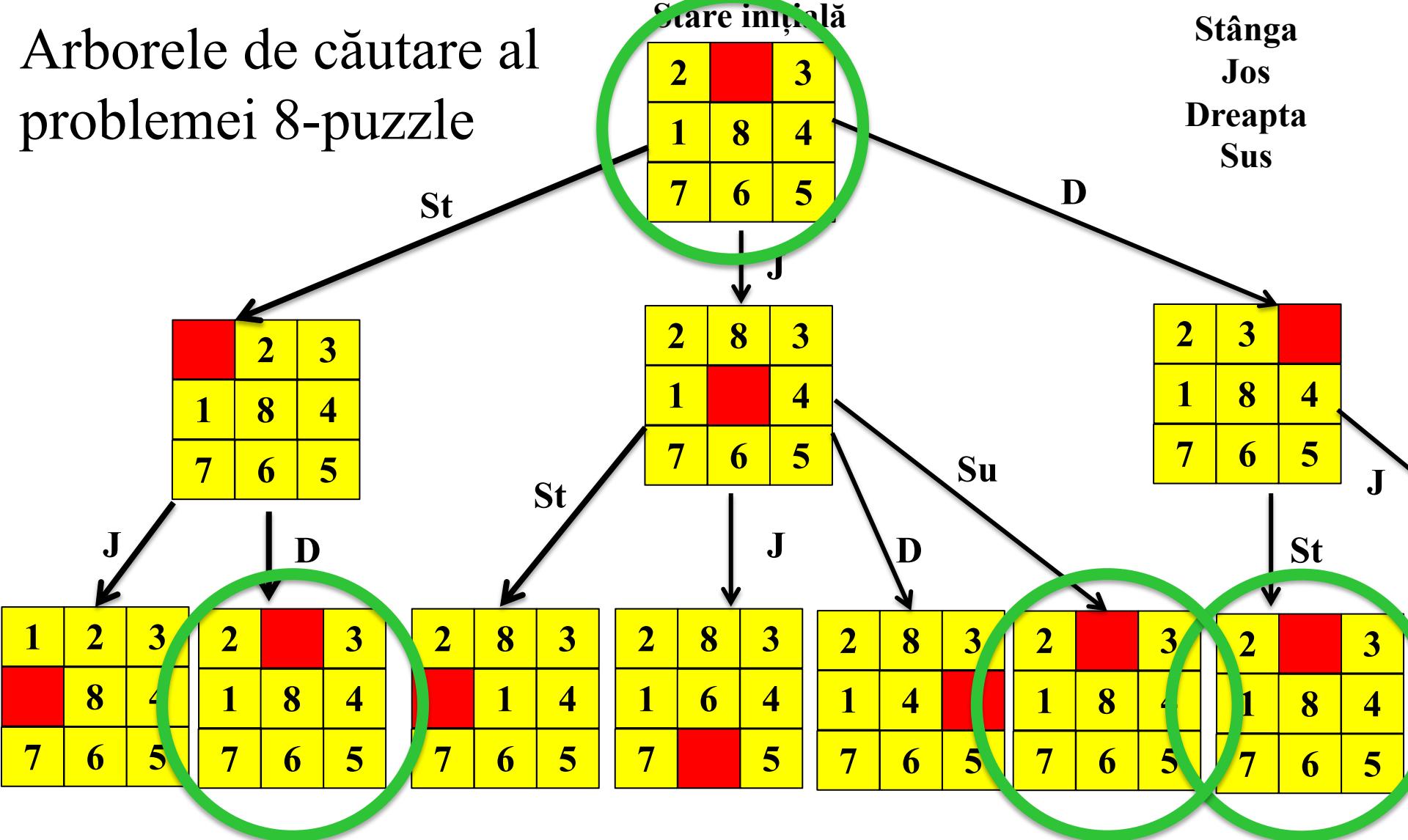
J

	2	3
1	8	4
7	6	5

2	8	3
1		4
7	6	5

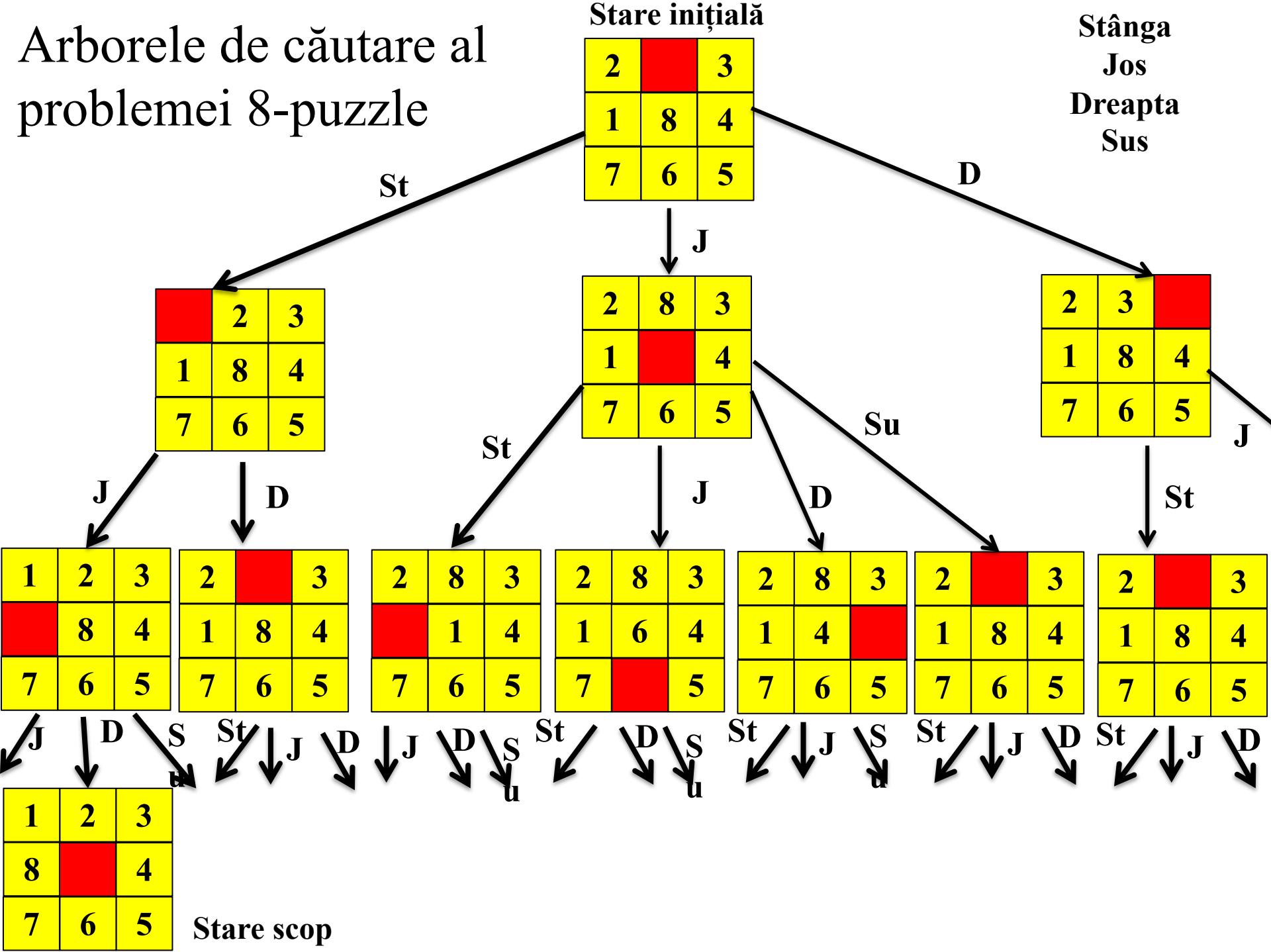
2	3	
1	8	4
7	6	5

# Arborele de căutare al problemei 8-puzzle



# Nod care se repetă

# Arborele de căutare al problemei 8-puzzle



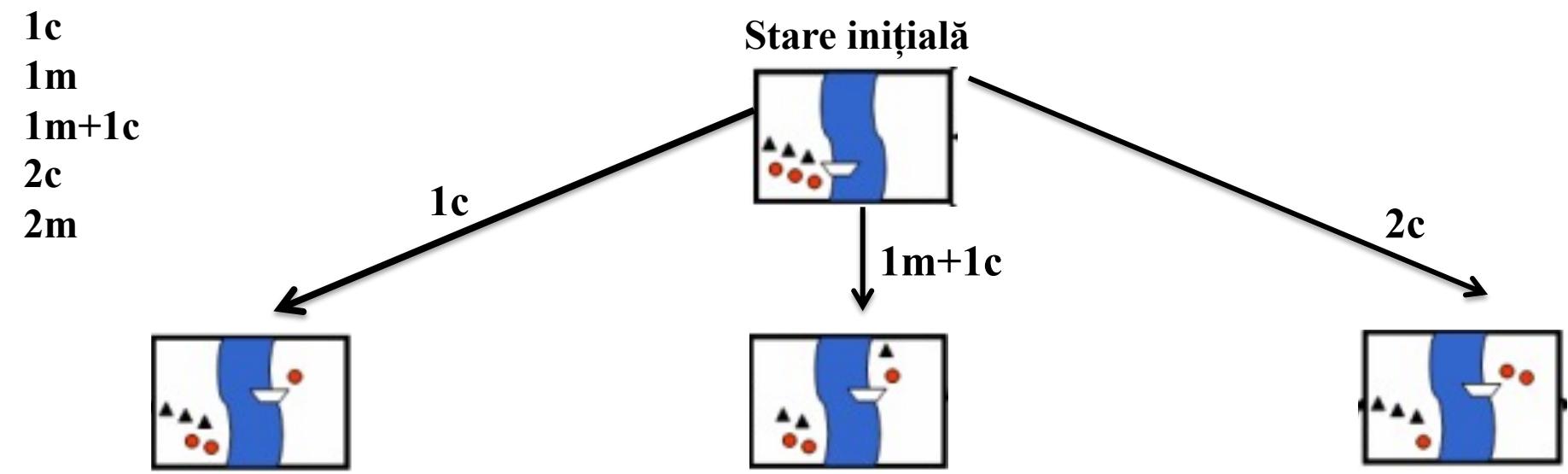
# Arborele de căutare al problemei misionarilor și canibalilor

1c  
1m  
1m+1c  
2c  
2m

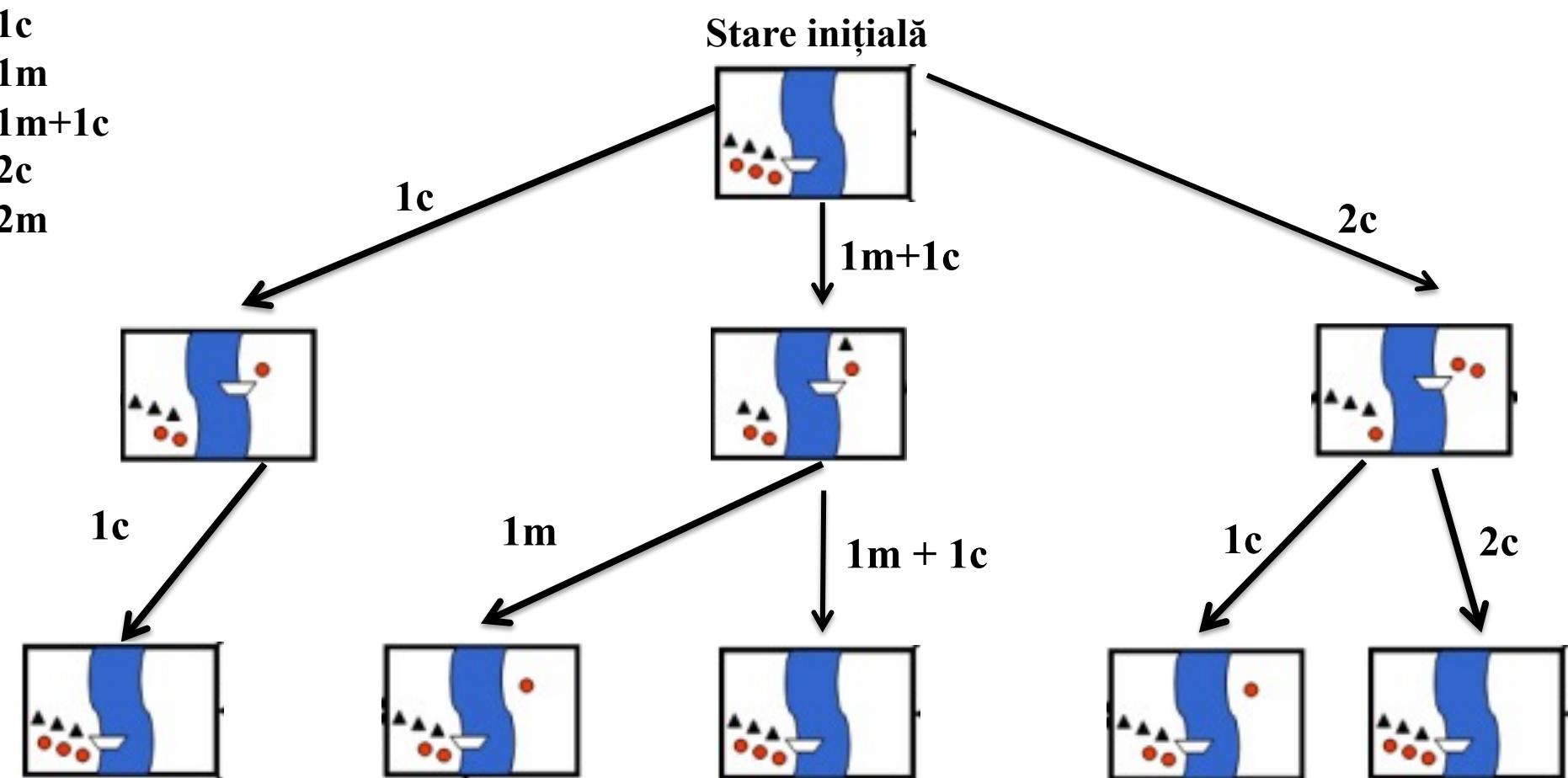
Stare inițială



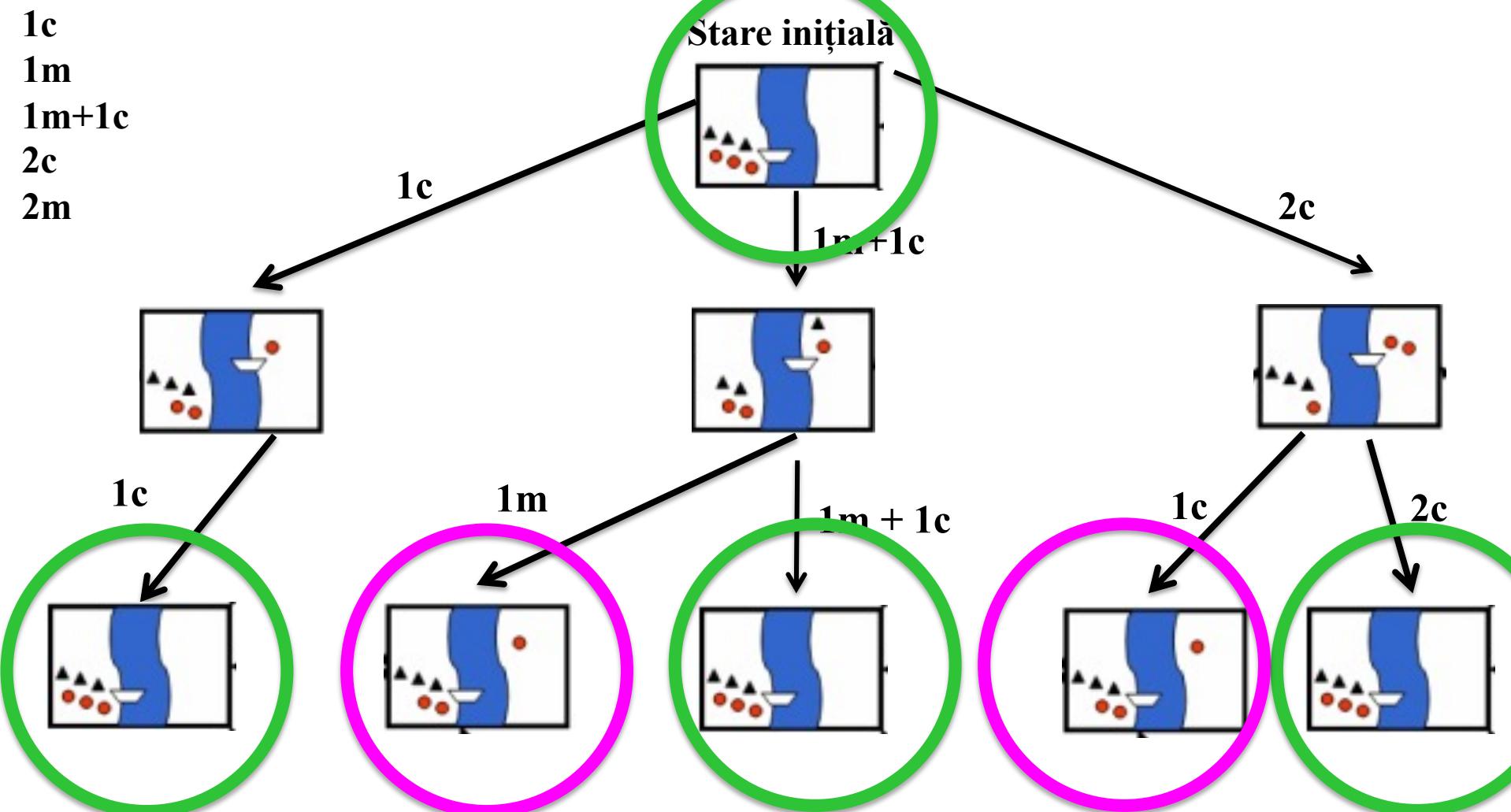
# Arborele de căutare al problemei misionarilor și canibalilor



# Arborele de căutare al problemei misionarilor și canibalilor



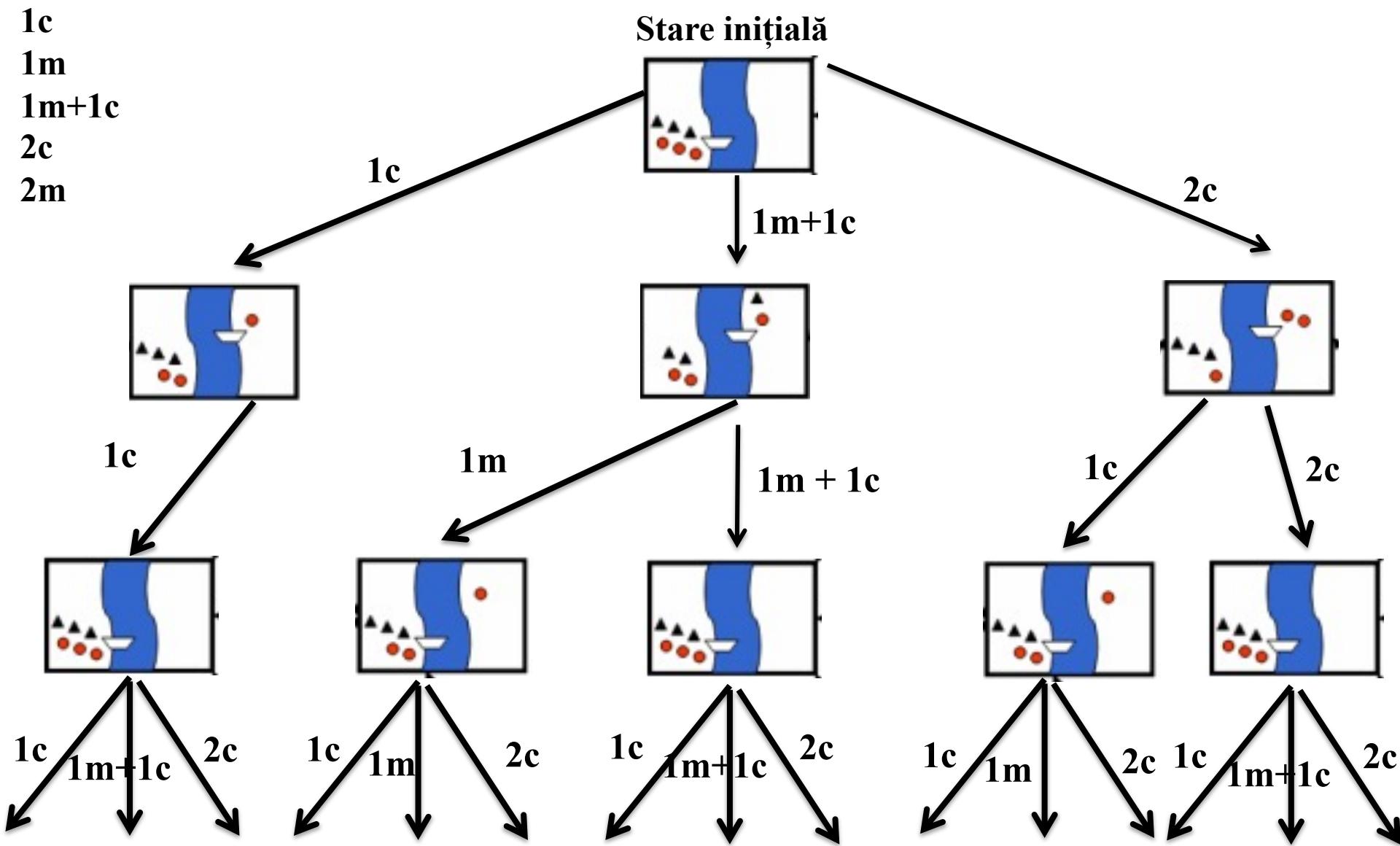
# Arborele de căutare al problemei misionarilor și canibalilor



Nod care se repetă

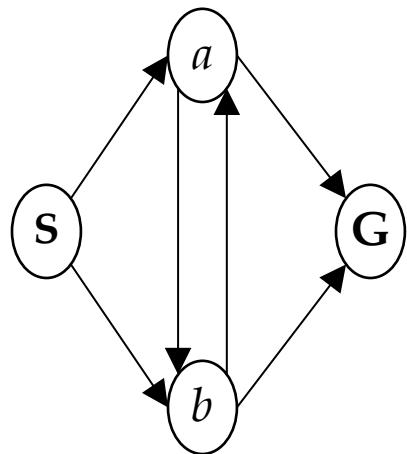
Nod care se repetă

# Arborele de căutare al problemei misionarilor și canibalilor



# Graful de stări vs. Arborele de căutare

Considerăm graful cu 4 stări de mai jos:

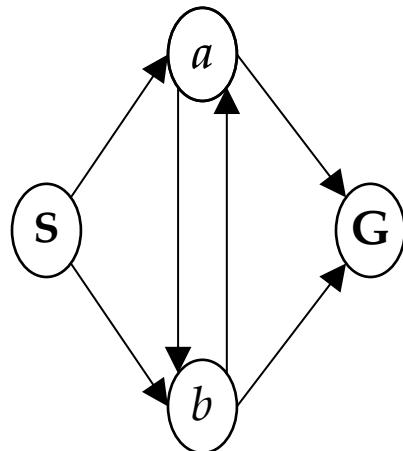


Cât de mare este arborele de căutare (pornim din S)

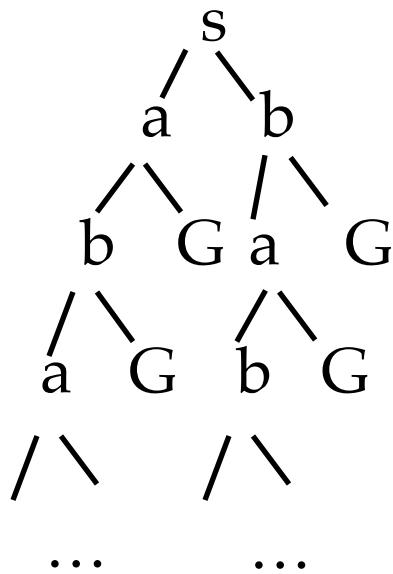


# Graful de stări vs. Arborele de căutare

Considerăm graful cu 4 stări de mai jos:



Cât de mare este arborele de căutare (pornim din S)



Important: foarte multe structuri repetitive în arborele de căutare

# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 9

# Note Proiect Kaggle - 361

Nr. crt.	Nume prenume student	Nota Proiect
1	ALBEI C. C. LIVIU-ANDREI	0,8
2	BALAWI I. USAMA	0
3	BORA T. I. DRAGOŞ-IONUȚ	0,85
4	CIOFLAN D. C. CEZAR-OCTAVIAN	0
5	CODREANU C. RADU-ȘTEFAN	0,9
6	COJOCARU L. ANDREI-LAURENȚIU	1,34
7	CREȚU V. LAURENȚIU-VASILE	1,42
8	CROITORU I. EDUARD-ADRIAN	0
9	DAVID D. V. VICTOR	1,16
10	FLORIAN D. LUCA-PAUL	1,04
11	IANCU G. FLORENTINA-MIHAELA	0,8
12	MARCU V. IOAN	0,65
13	MINCOV A. VIRGINICA-CRISTINA	0
14	NAE I. E. MIRCEA-ȘTEFAN	0,5
15	NĂSTASE D. MARIUS-NICOLAE	0,75
16	NIȚĂ B. AL. RAUL-ALEXANDRU	0
17	POPESCU C. ALINA-ELENA	1,03
18	SCORBUREANU V. SERGIU-NICHITA	0,5
19	SORETE G. ROBERT-ALEXANDRU	1,38
20	SPĂTARU D. CĂTĂLIN-GABRIEL	1,43
21	STANA M. D. MARIUS-VLAD	0,8
22	STOIAN C. O. VLAD	0,7
23	ȚUGUI I. IUSTIN-ION	0,5

# Note Proiect Kaggle - 362

Nr. crt.	Nume prenume student	Nota Proiect
1	BILICI C. MIHAI-RAZVAN	0,65
2	BLOGU AL. FL. ADRIAN-TOMA	0,73
3	BORCAN M. CRISTIAN-BOGDAN	1,16
4	BRÎNCEANU E. ANDI-MĂDĂLIN	1,21
5	CIBOTARI GH. AUGUSTIN-ION	0,97
6	DRAGHIOTI I.-C. ANDREEA-MARIA	0,85
7	GHERGU M. NICOLAE-MARIUS	1,02
8	HOLMANU M. ANTONIO-MARIUS	0
9	LICU N. MIHAI-GEORGE	1,4
10	LINCĂ I. M. BIANCA-MARIA	0
11	MATACHE M. ALEXANDRU	0,9
12	MATEI N. B. TUDOR-CRISTIAN	1,43
13	MORARU M. RADU-ANDREI	1,41
14	MUŞAT E. R. SILVIU-GEORGE	0,7
15	STANCIU M. ALEXANDRA-ANDREEA	1,42
16	STĂTESCU I. RELU	1
17	TELEA C. C. MARIA-LAURA	1,31
18	TOMA M. FL. ALEXANDRU	0,85
19	TRĂNCĂNĂU M. O. CEZAR-ALEXANDRU	1,16
20	VRACIU V. ANDREEA	0,9

# Note Proiect Kaggle - 363

Nr. crt.	Nume prenume student	Nota Proiect
1	ANTONESCU N. CRISTINA-ANDREEA	0,93
2	APOSTU A. V. MIHAI-ADRIAN	0,8
3	BANCEA C. DAN-ANDREI	0,75
4	BĂNESARU I. DENISA-GEORGIANA	1,24
5	BURLACU V. MIRCEA-FLORIAN	0,8
6	CROITORU M. VLADIMIR-ANDREI	0
7	CUCU AL. ȘTEFAN-CĂTĂLIN	1,44
8	DURA D. FL. ALEXANDRU-BOGDAN	1,01
9	ENCIU L. C. ELENA-CRISTINA	0,85
10	FĂRCĂȘANU E.C. TUDOR- ANDREI	1,5
11	FILIP L. RAZVAN ADRIAN	0,85
12	GAVRILĂ M.-S. VLAD-THEODOR	0,62
13	ISAC V. O. ȘTEFAN	0
14	MILITARU A.A ANDREI- ALEXANDRU	1,19
15	NIȚĂ M. ANDREEA-DIANA	1,13
16	PAVEL I. FL. ALEXANDRU	0,86
17	POPESCU G. FLORIN- DANIEL	0
18	SON V. ANDREEA-MARINA	1,24
19	STANA D. M. ANDREEA-THEODORA	0,85
20	TRĂȘCĂLIE L. RADU-NICOLAE	0,92
21	TUDOR M. NICU-CORNEL	0
22	UDREA C. IULIA-MARIA	1,27
23	ZOTIC V. D. MELANIA-ANEMONA	1,23

# Note Proiect Kaggle - 364

Nr. crt.	Nume prenume student	Nota Proiect
1	ALEXANDRESCU D. AL. TUDOR-ALEX	1,35
2	ALEXANDRESCU D. MARIAN-GABRIEL	0,8
3	BRÎNCEANU D. RALUCA-ALEXANDRA	0,54
4	BUTURUGĂ C. E. GEORGE-ALEXANDRU	0,7
5	CRIȚĂ B. M. ANDREI-IONUȚ	0
6	DINOIU V. C. NICOLETA-ANASTASIA	1,35
7	DUMITRACHE V. V. RĂZVAN-CRISTIAN	0,5
8	DUMITRU M. L. RADU-ANDREI	0,95
9	GHEORGHIȚĂ E. ELENA-RALUCA-LO	0,75
10	GHIȚĂ G. D. DARIUS-FILIP	0,5
11	GUGIU E. ALEXANDRU	0
12	IONESCU C. LORENA-ELENA	1,07
13	IONIȚĂ M. R. ROXANA-DIANA	0,93
14	ISTRATE V. DORIAN-MIHAI	0,85
15	LASCU M. V. GEORGE-ALBERT	0,87
16	MANOLE A. GEORGE-ADRIAN	1,24
17	OANCEA E. E. VLAD	0
18	POPA V. D. BOGDAN-GABRIEL	1
19	POPESCU C. MIHNEA	1,28
20	ȘAMATA GH. ROBERT	0,65
21	URLĂTEANU C. A. ALEXANDRU-IOAI	0,8

# Note Proiect Kaggle – anul 4

Anul 4		
Nr. crt.	Nume prenume student	Nota Proiect
1	DINU ALEXANDRU	1,07
2	HONCIU BOGDAN	0,5
3	VARASCIUC ANDREI	0,52
4	BUCURIE BOGDAN	0,9
5	CRISTEA MIHNEA STEFAN	0,7
6	JIGAU ANDREI	0,96

# Recapitulare – cursul trecut

1. Agenți inteligenți și mediile în care aceștia funcționează
2. Rezolvarea problemelor prin căutare
  - Graful stărilor
  - Arborele de căutare

# Cuprinsul cursului de azi

## 1. Rezolvarea problemelor prin căutare

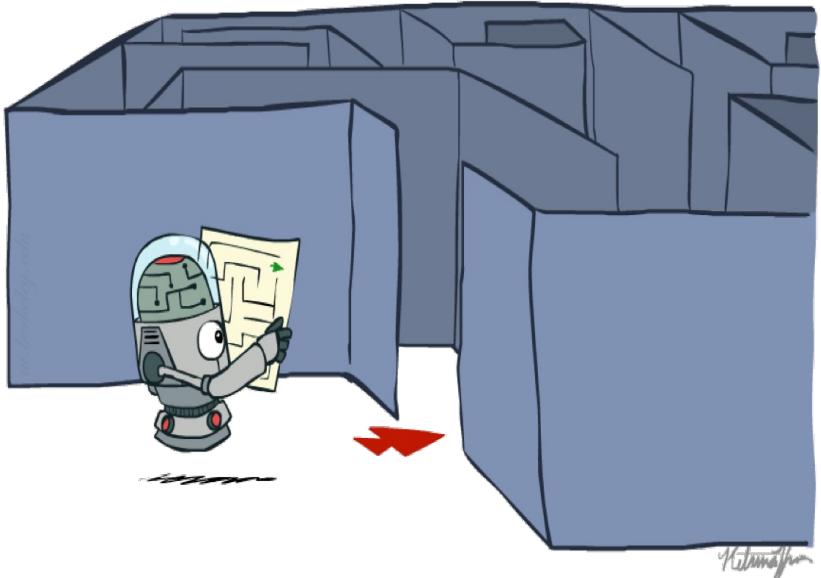
- Graful stărilor
- Arborele de căutare

## 2. Căutare neinformată

- Căutare în lățime (Bread-Fist Search)
- Căutare în adâncime (Depth-Fist Search)
- Căutare în adâncime limitată (depth-limited search)
- Căutare în adâncime incrementală (iterative deepening search)
- Căutare uniformă după cost (uniform-cost search)

# Căutare

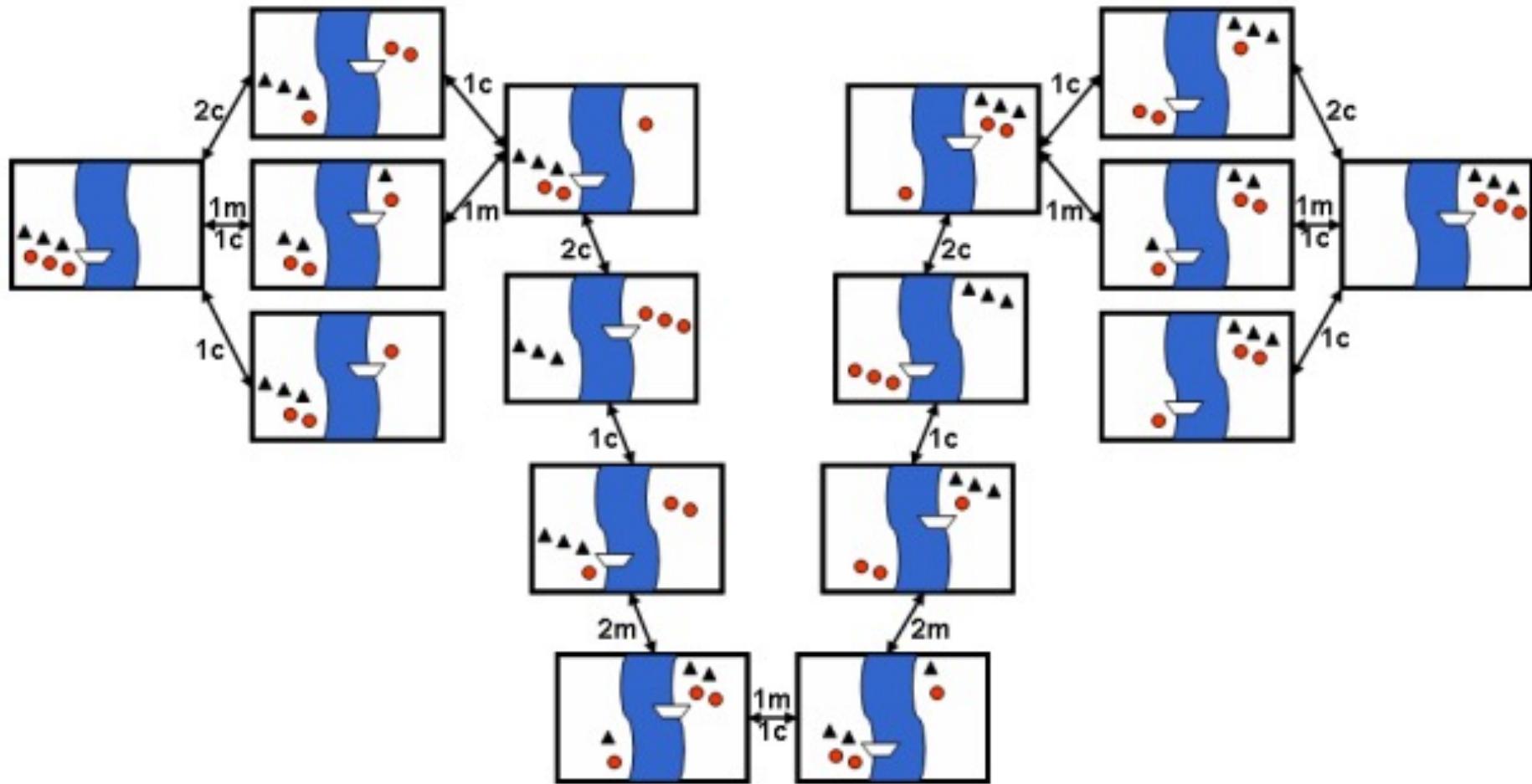
- Problemă de căutare
  - stări (configurații ale lumii)
  - acțiuni și costuri
  - funcția succesor
  - stare inițială și stare scop
- Arbore de căutare
  - noduri: soluții parțiale
  - arce: acțiuni
  - o soluție parțială are asociat un cost (suma costurilor acțiunilor = arcele)



# Graful de stări al unei probleme

- Graf orientat care conține toate stările lumii (o stare = un nod)
- Fiecare stare apare o singură dată
- Arcele reprezintă acțiuni
  - modelează funcția successor
  - în ce stări putem ajunge din starea curentă realizând o anumită acțiune
  - deseori avem arc de la starea A la starea B și invers (acțiuni simetrice), crează circuite în arborele de căutare

# Graful de stări al problemei misionarilor și a canibalilor



# Arborele de căutare al unei probleme

- Tot un graf orientat, dar organizat pe nivele:
  - nivelul 0 = rădăcina (starea inițială)
  - nivelul i = nod obținut prin traversarea a i arce de la rădăcină din graful de stări
- Un nod desemnează o soluție parțială
- Un arc reprezintă un pas în construirea unei soluții
- Nodurile pot conține stări care se repetă
  - pot ajunge într-o stare urmând drumuri diferite
- Vrem să găsim un drum (= soluție) în graful de stări de la starea inițială la o stare scop (stare finală).

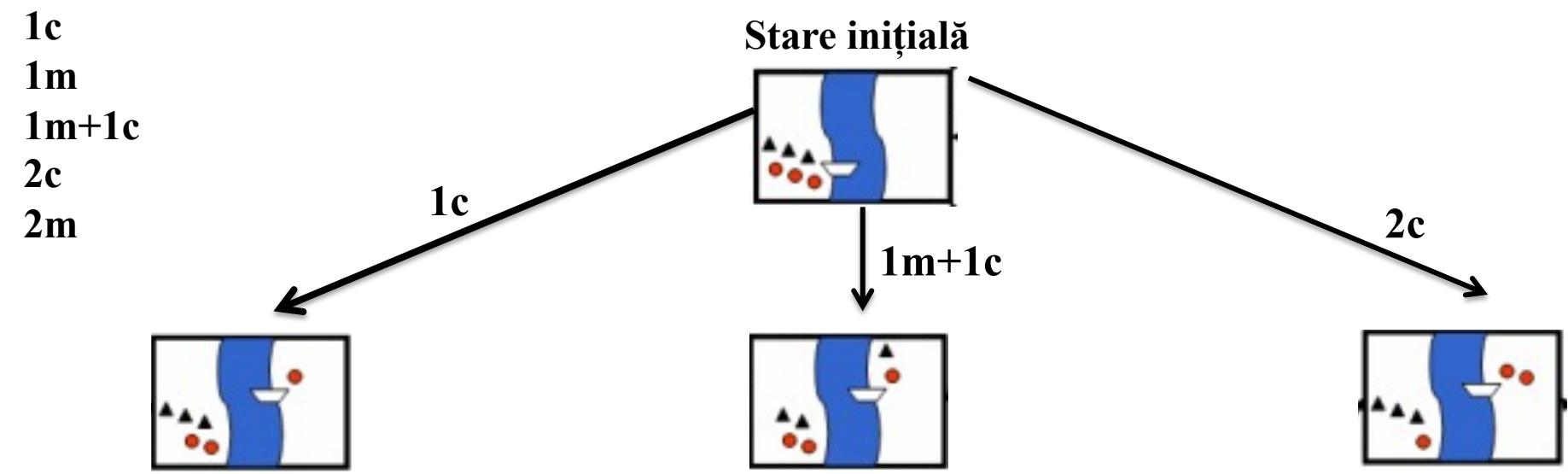
# Arborele de căutare al problemei misionarilor și canibalilor

1c  
1m  
1m+1c  
2c  
2m

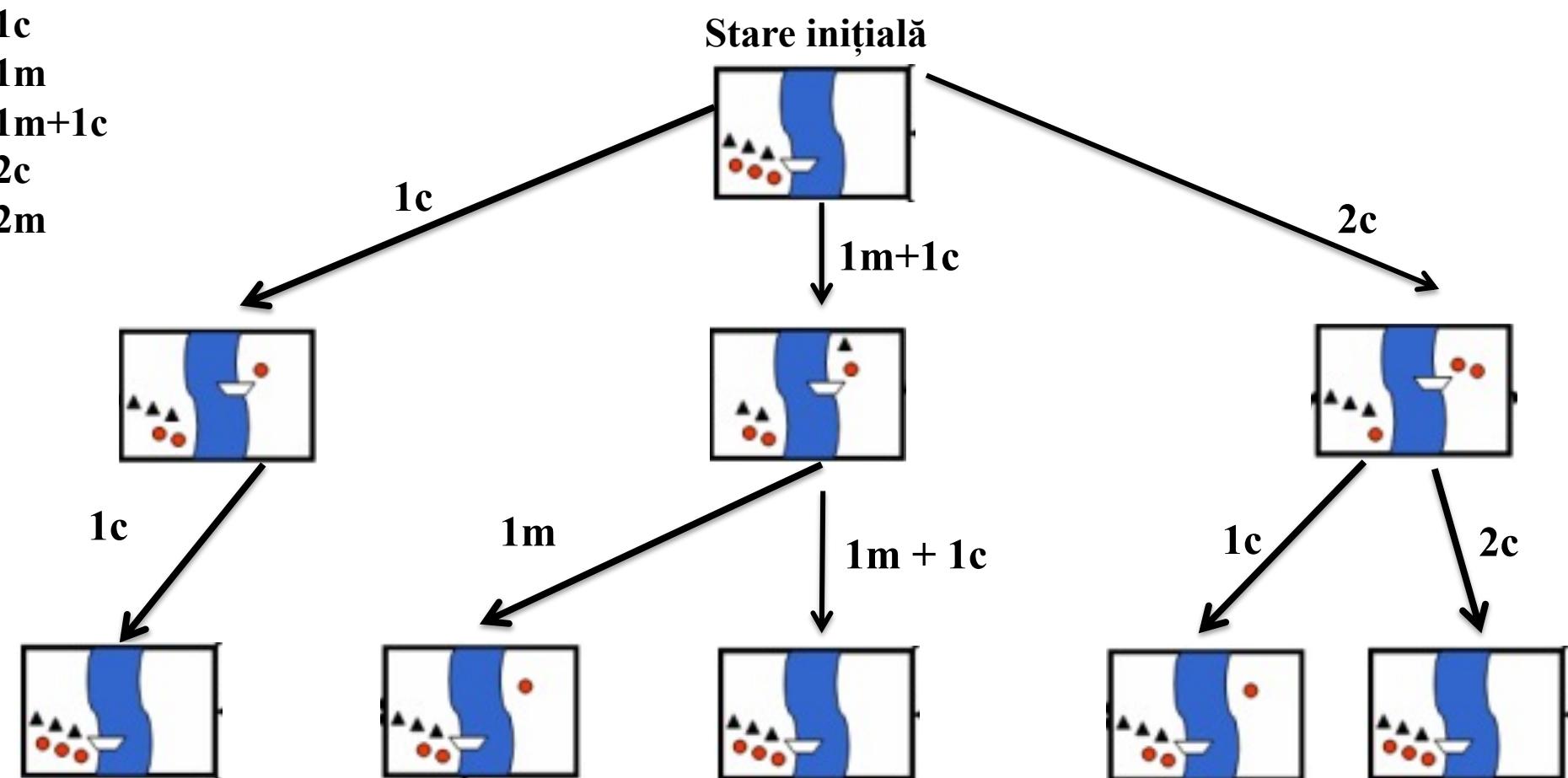
Stare inițială



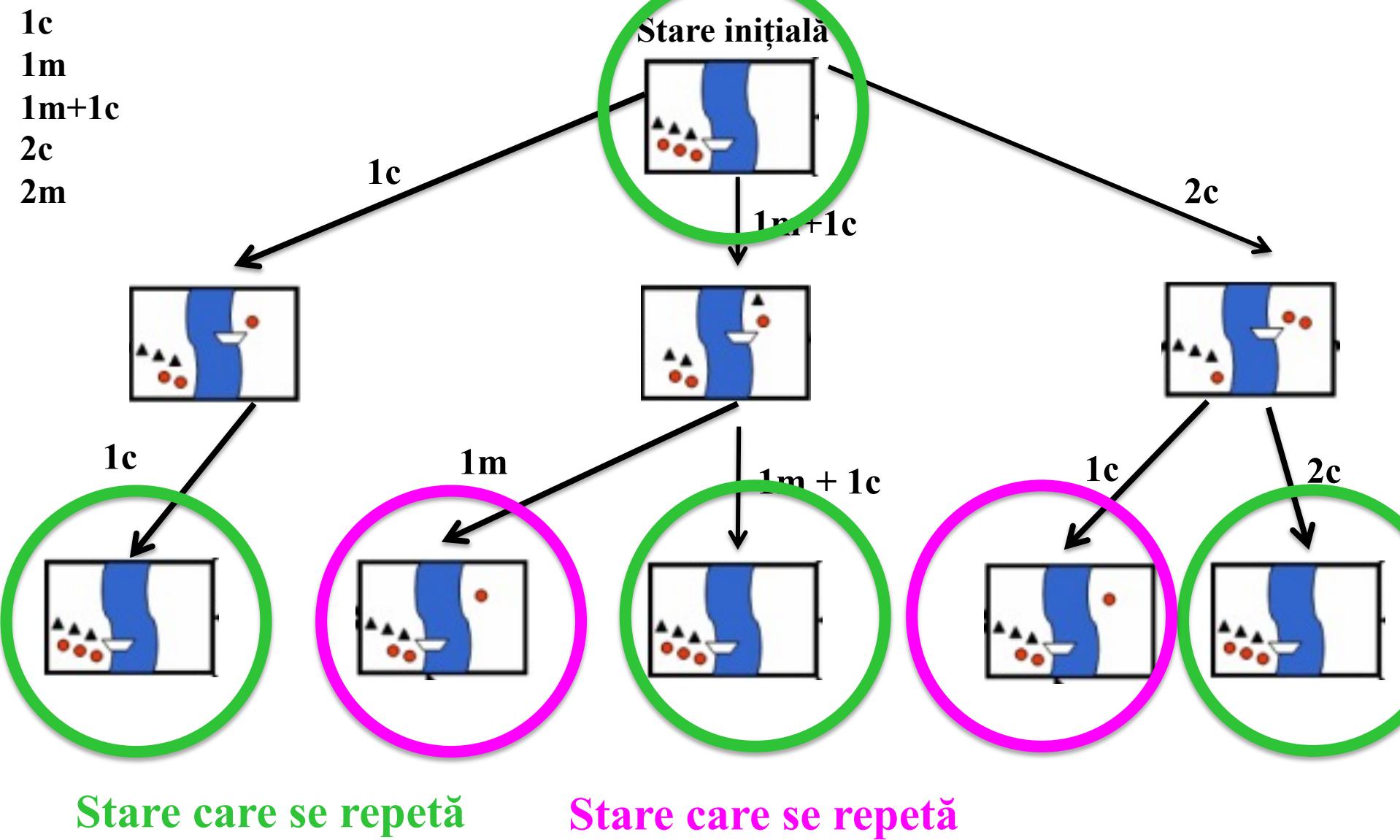
# Arborele de căutare al problemei misionarilor și canibalilor



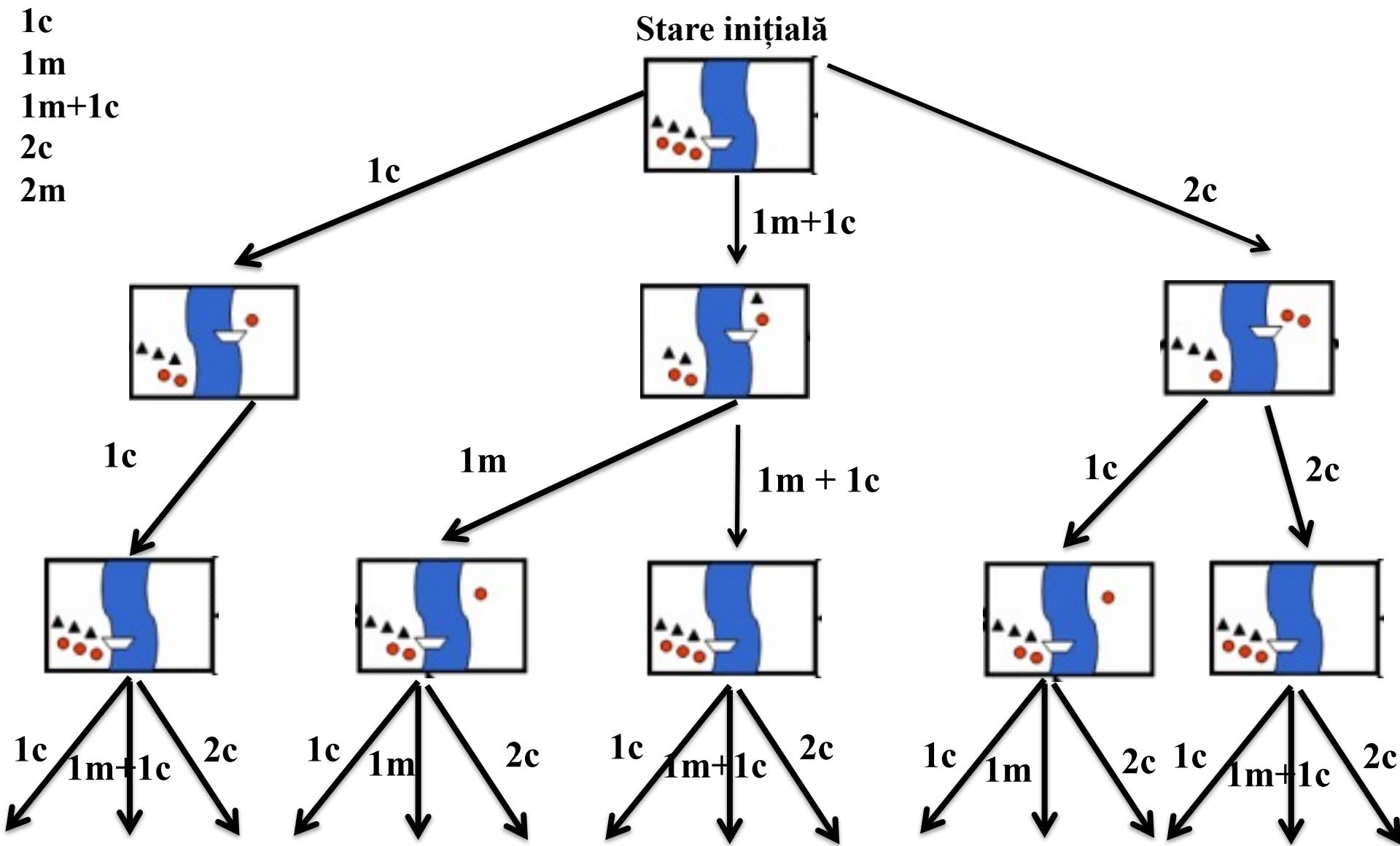
# Arborele de căutare al problemei misionarilor și canibalilor



# Arborele de căutare al problemei misionarilor și canibalilor

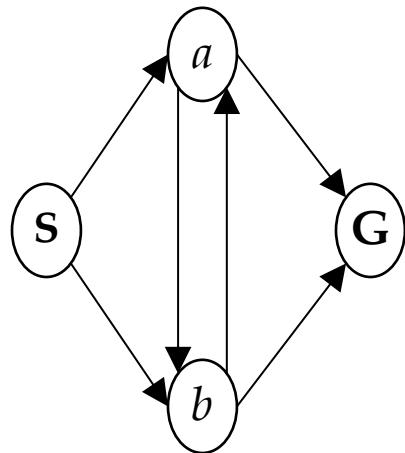


# Arborele de căutare al problemei misionarilor și canibalilor



# Graful de stări vs. Arborele de căutare

Considerăm graful cu 4 stări de mai jos:

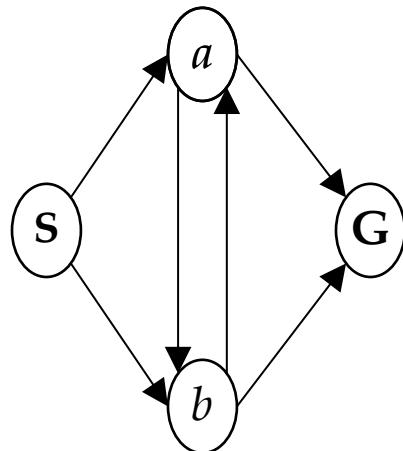


Cât de mare este arborele de căutare (pornim din S)

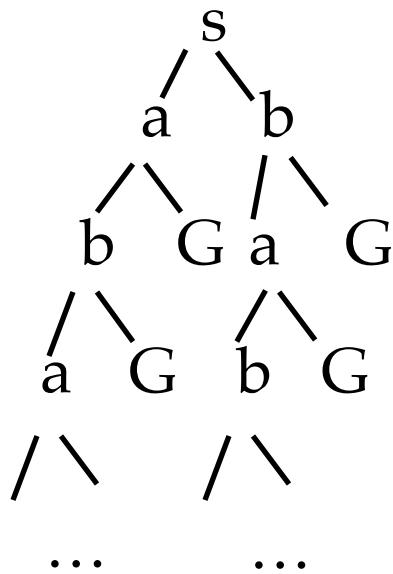


# Graful de stări vs. Arborele de căutare

Considerăm graful cu 4 stări de mai jos:

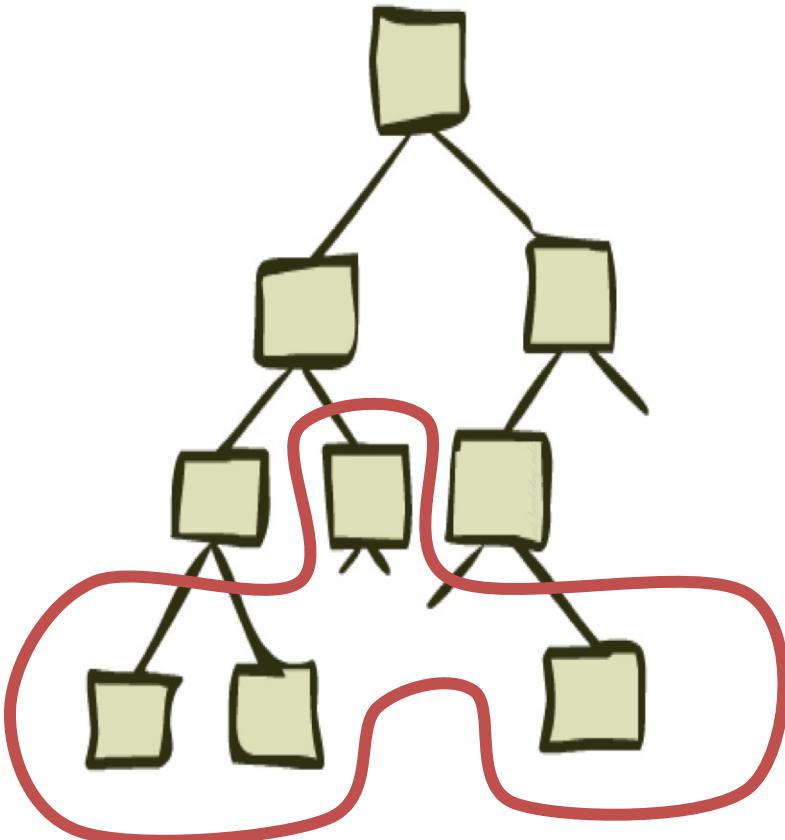


Cât de mare este arborele de căutare (pornim din S)

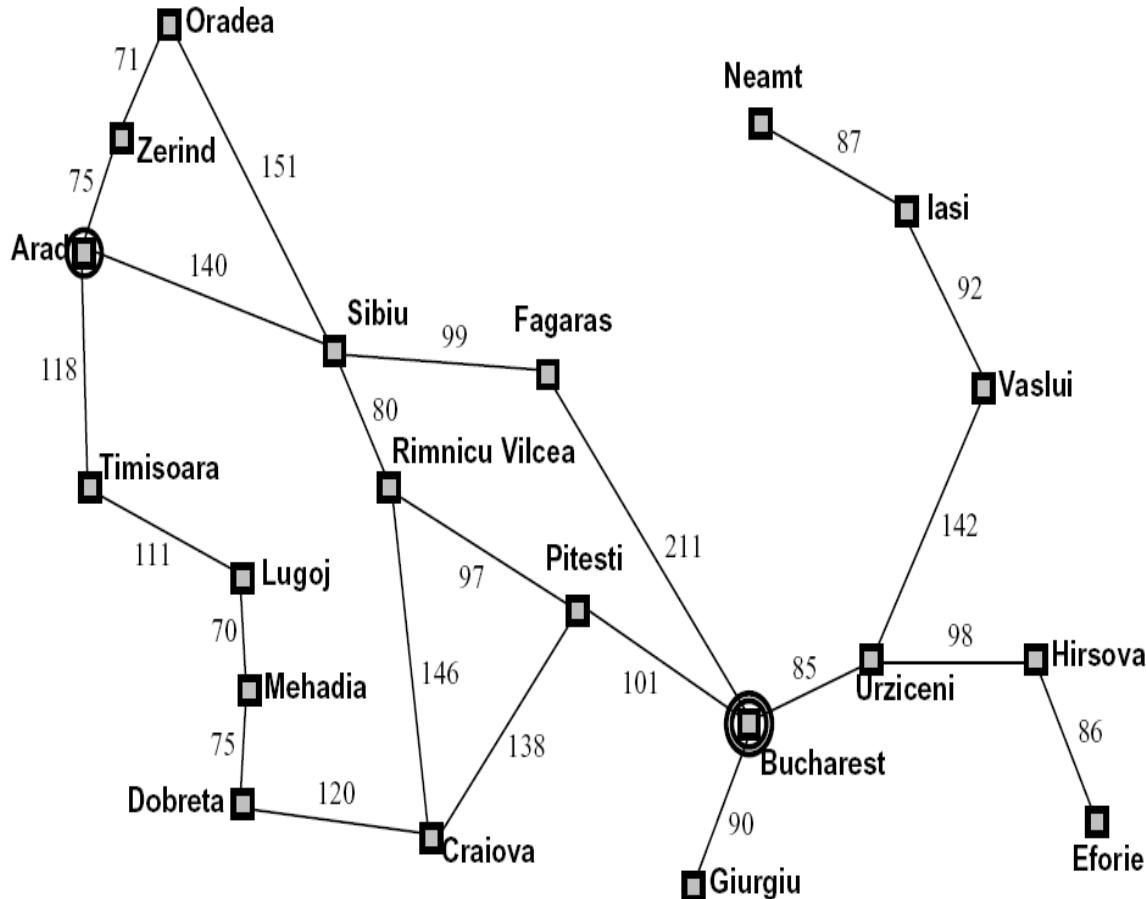


Important: foarte multe structuri repetitive în arborele de căutare

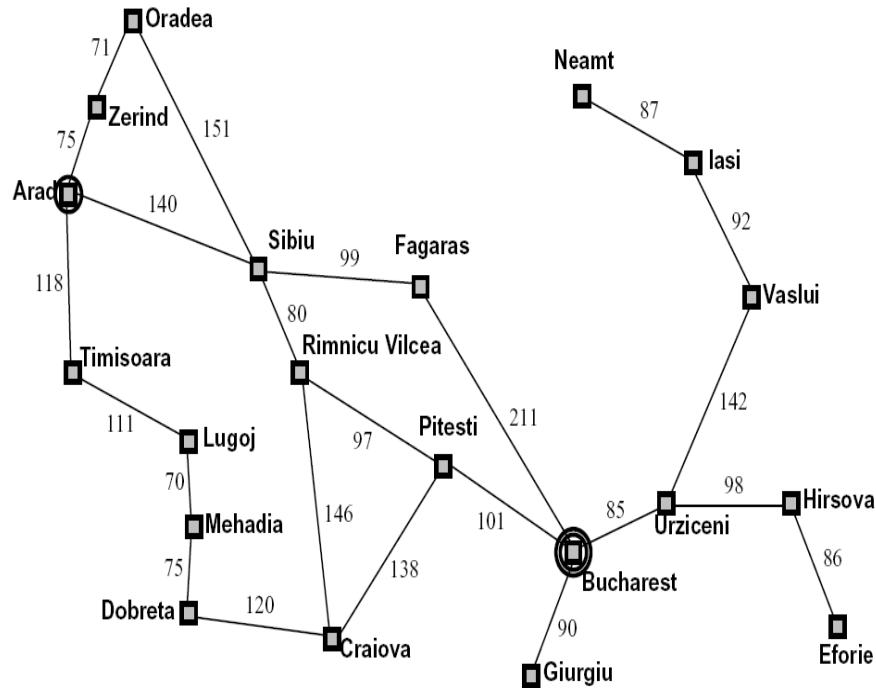
# Căutare cu un arbore de căutare



# Exemplu de căutare: Arad-Bucureşti

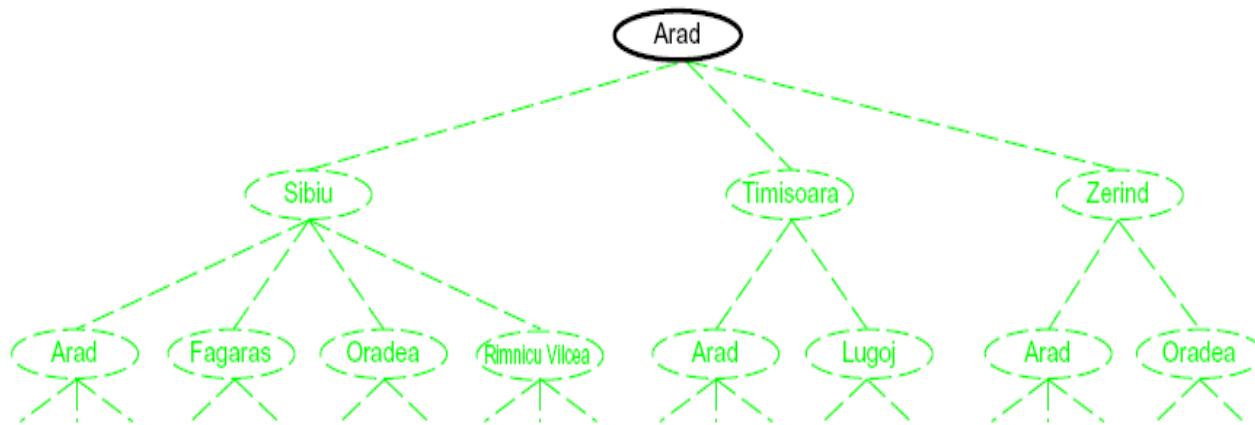


# Exemplu de căutare: Arad-Bucureşti



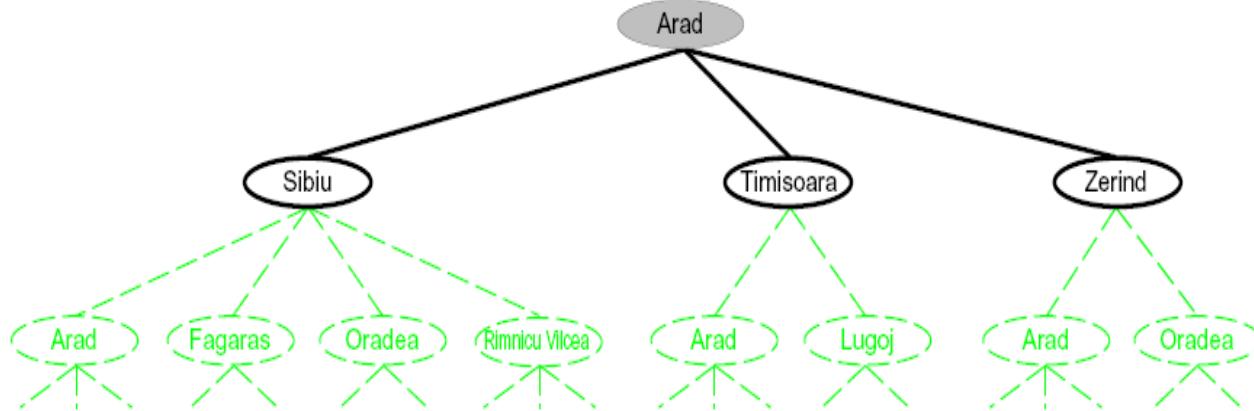
- Spațiul stărilor:
  - Orașe
- Funcție succesor:
  - drumuri: mergi la orașul adjacente cu costul = distanța
- Stare inițială:
  - Arad
- Scop care poate fi verificat:
  - stare == Bucureşti?
- Soluție?

# Căutare cu un arbore de căutare



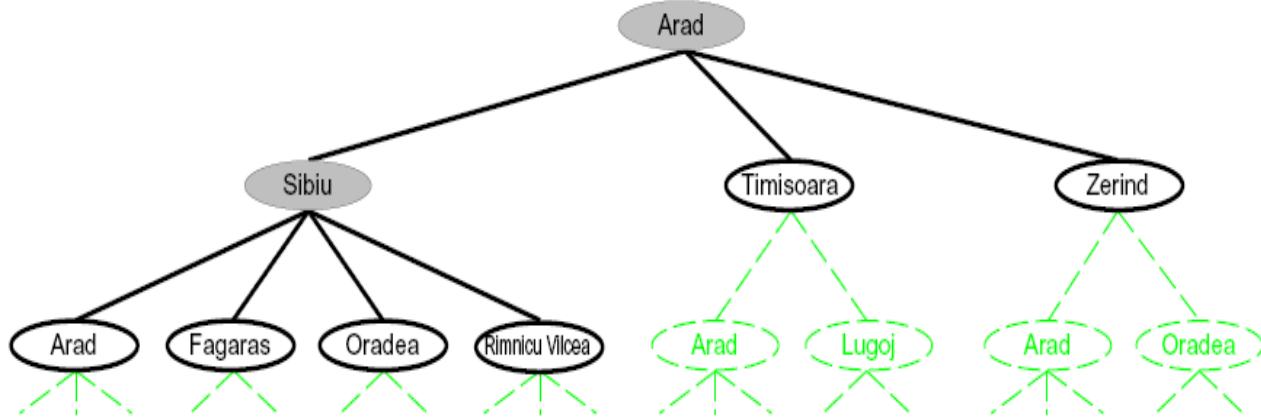
- Căutare:
  - expandarea planurile posibile (nodurile din arbore)
  - menținerea unei liste de noduri de **frontieră**, ce separă graful spațiului de căutare în regiunea explorată și regiunea neexplorată
  - se încearcă expandarea unui număr cât mai mic de noduri

# Căutare cu un arbore de căutare



- Căutare:
  - expandarea planurile posibile (nodurile din arbore)
  - menținerea unei liste de noduri de **frontieră**, ce separă graful spațiului de căutare în regiunea explorată și regiunea neexplorată
  - se încearcă expandarea unui număr cât mai mic de noduri

# Căutare cu un arbore de căutare



- Căutare:
  - expandarea planurile posibile (nodurile din arbore)
  - menținerea unei liste de noduri de **frontieră**, ce separă graful spațiului de căutare în regiunea explorată și regiunea neexplorată
  - se încearcă expandarea unui număr cât mai mic de noduri

# Arbore de căutare general

funcția **ARBORE-CAUTARE** (problemă, strategie) returnează o soluție sau eșec

initializează **frontiera** folosind **starea inițială a problemei**

ciclează

dacă **frontiera** este vidă **atunci returnează** eșec

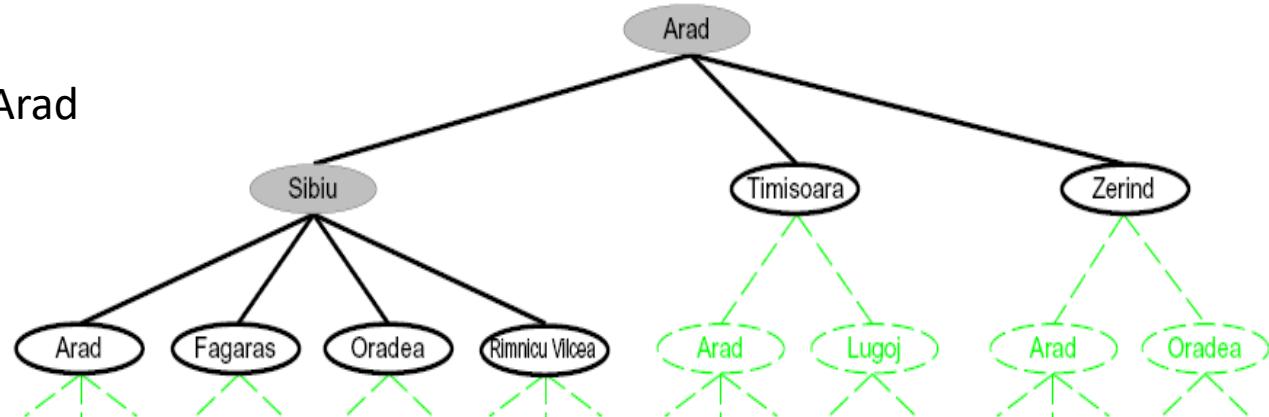
alege un **nod frunză** conform **strategiei** și elimină-l din **frontieră**

dacă **nodul** conține o **stare scop**

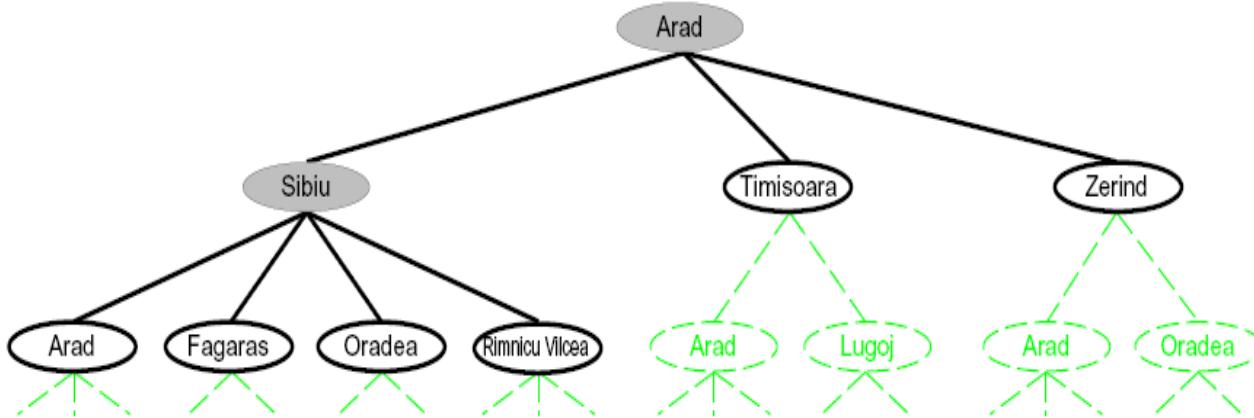
**atunci returnează** soluția corespunzătoare

expandează nodul ales, adăugând nodurile succesor generate în **frontieră**

Drum redundant: Arad-Sibiu-Arad



# Arbore de căutare general



Frontiera conține nodurile frunză din arborele de căutare (în desenul de mai sus frontiera e dată de nodurile albe: Arad, Făgăraș, Oradea, Râmnicu Vâlcea, Timișoara, Zerind). Fiecare nod frunză reprezentă o soluție parțială construită = o secvență de acțiuni ce pornește din nodul rădăcină.

La fiecare pas:

- alege pe baza unei strategii un anumit nod frunză din frontieră;
- testează dacă nodul curent este nod-scop: dacă este nod-scop returnează soluția;
- expandează nodul curent: generează toți succesorii nodului curent și îi adaugă în frontieră

# Arbore de căutare general

funcția **ARBORE-CAUTARE** (problemă, strategie) returnează o soluție sau eșec

initializează **frontiera** folosind **starea inițială a problemei**  
initializează multimea de noduri explorate cu multimea vidă  
**ciclează**

dacă **frontiera** este vidă **atunci returnează** eșec

alege un **nod frunză** conform **strategiei** și elimină-l din **frontieră**

dacă **nodul** conține o **stare scop**

**atunci returnează** soluția corespunzătoare

adaugă nodul curent multimii de noduri explorate (vizitate)

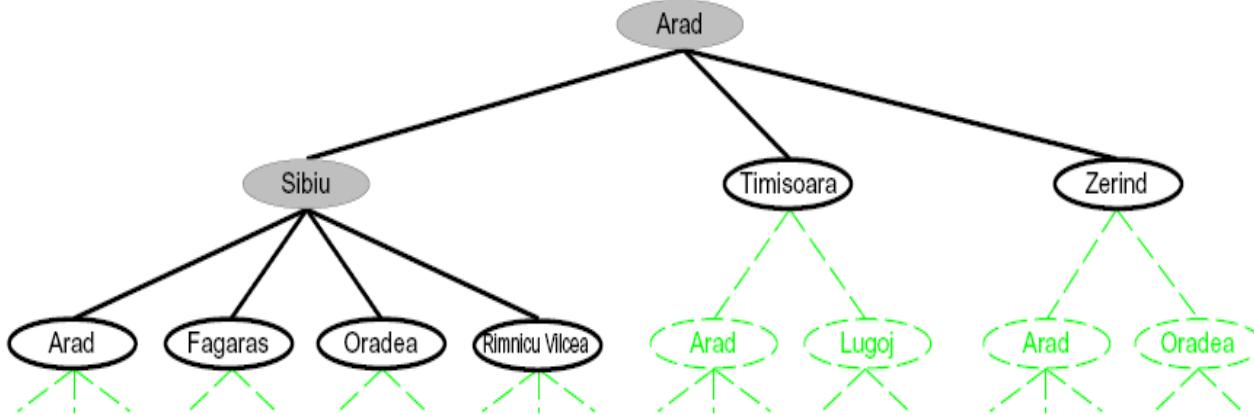
expandează nodul ales, adăugând nodurile succesor generate în **frontieră**

dacă nodurile nu sunt în **frontieră** sau în multimea de noduri explorate

**Frontiera** = structură de date ce păstrează toate nodurile care mai trebuie expandate

- coadă (FIFO) – scoate cel mai vechi element din listă
- stivă (LIFO) – scoate cel mai nou element din listă
- listă de priorități – scoate un element cu cea mai mare prioritate conform unei funcții de ordonare

# Arbore de căutare general

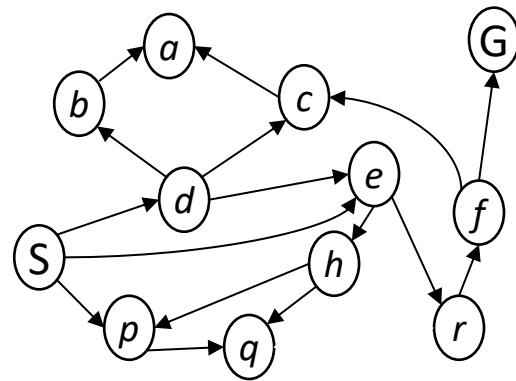


Frontiera conține nodurile frunză din arborele de căutare (în desenul de mai sus frontiera e dată de nodurile albe: Arad, Făgăraș, Oradea, Râmnicu Vâlcea, Timișoara, Zerind). Fiecare nod frunză reprezentă o soluție parțială construită = o secvență de acțiuni ce pornește din nodul rădăcină.

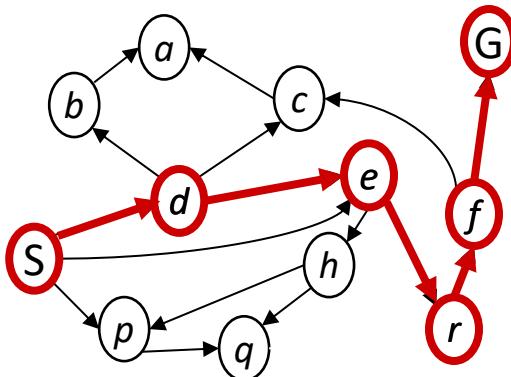
La fiecare pas:

- alege pe baza unei strategii un anumit nod frunză din frontieră;
- testează dacă nodul curent este nod-scop: dacă este nod-scop returnează soluția;
- expandează nodul curent: generează toți succesorii nodului curent și îi adaugă în frontieră **dacă aceștia nu au fost explorați (vizitați) sau nu sunt deja în frontieră (evită crearea de circuite)**

# Arbore de căutare - exemple

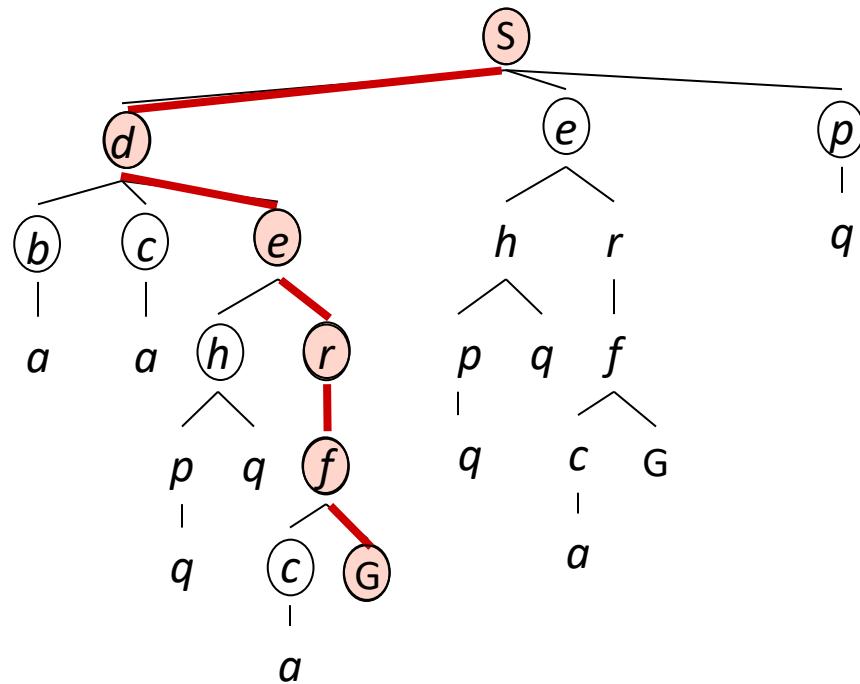


# Arbore de căutare - exemplu



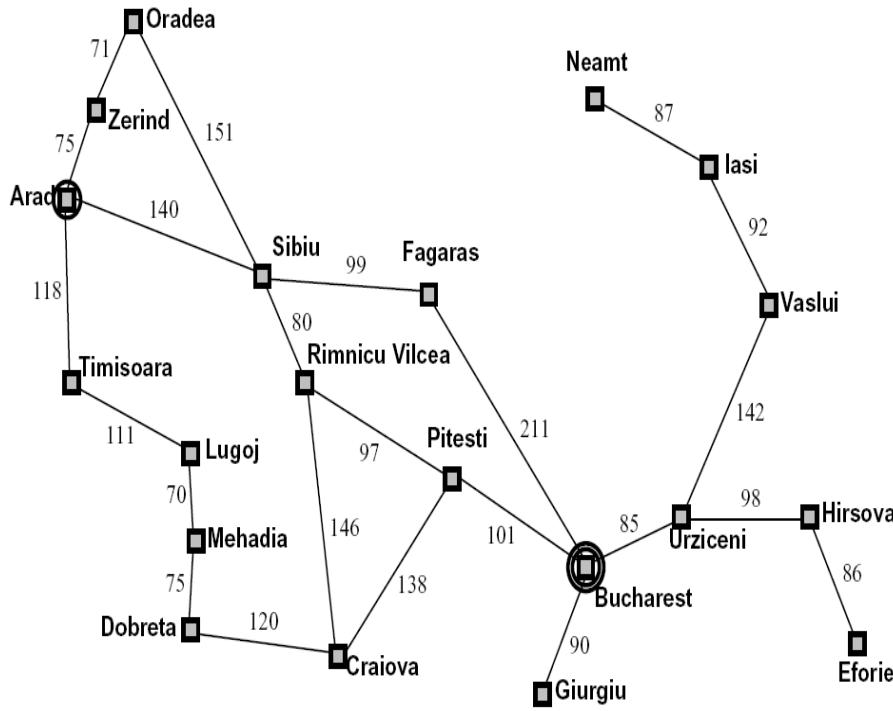
Arborele de căutare

Frontiera



~~s~~  
~~s → d~~  
 s → e  
 s → p  
 s → d → b  
 s → d → c  
~~s → d → e~~  
 s → d → e → h  
~~s → d → e → r~~  
~~s → d → e → r → f~~  
 s → d → e → r → f → c  
~~s → d → e → r → f → G~~

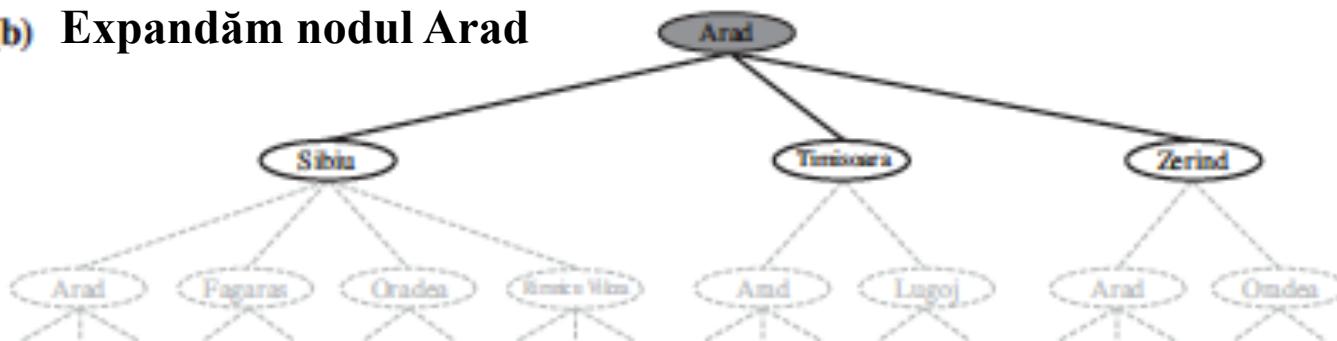
# Arbore de căutare - exemple



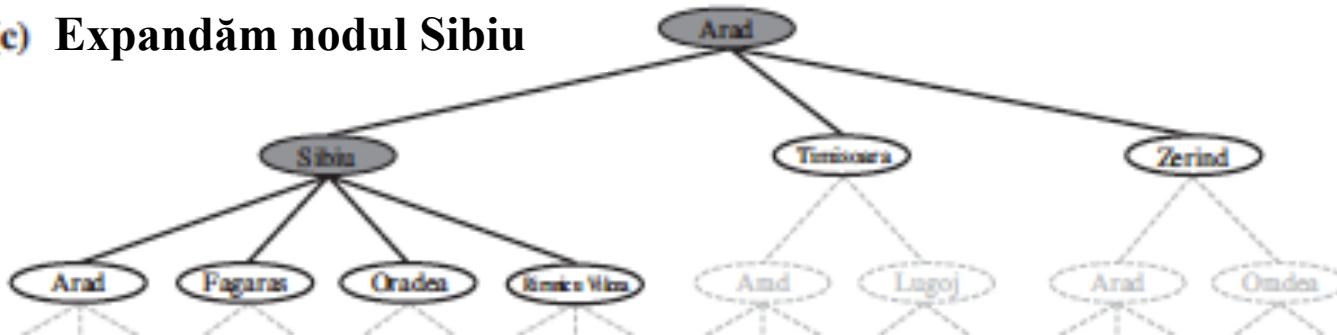
**(a) Stare inițială**



**(b) Expandăm nodul Arad**



**(c) Expandăm nodul Sibiu**

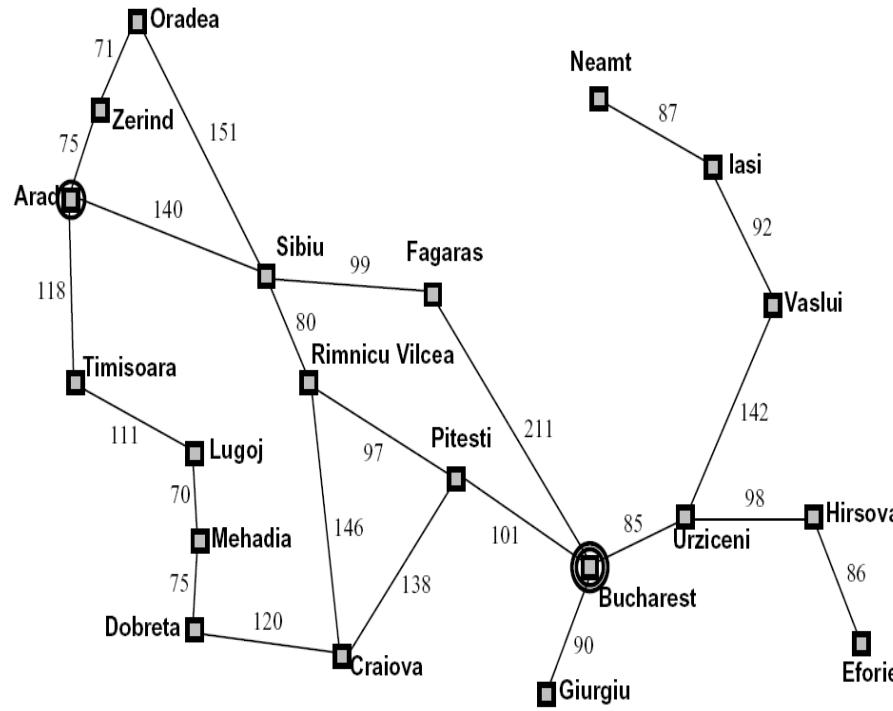


Trei tipuri de noduri:

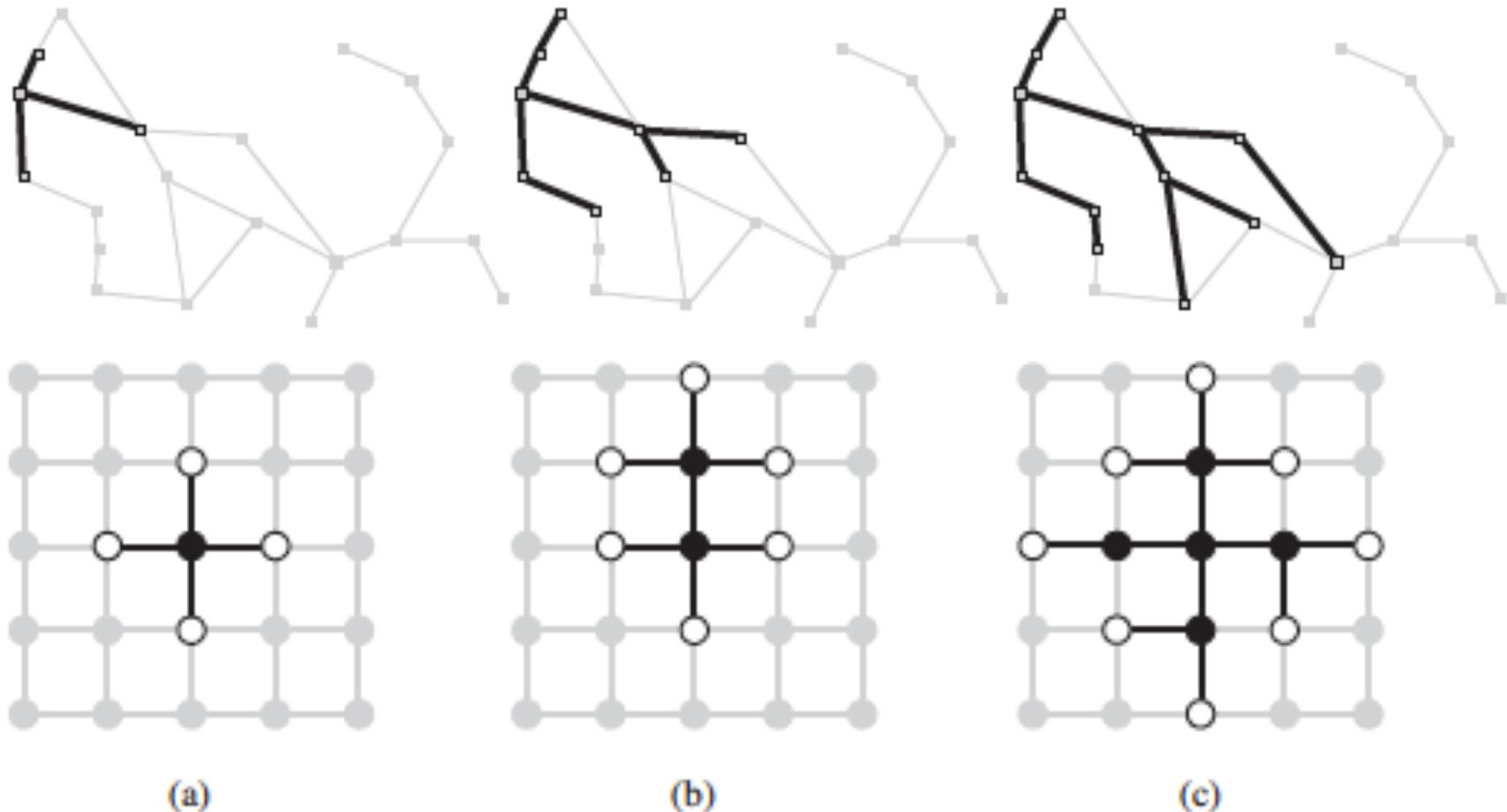
- **noduri vizitate**
- **noduri din frontieră**
- **noduri negenerate**



# Arbore de căutare - exemple



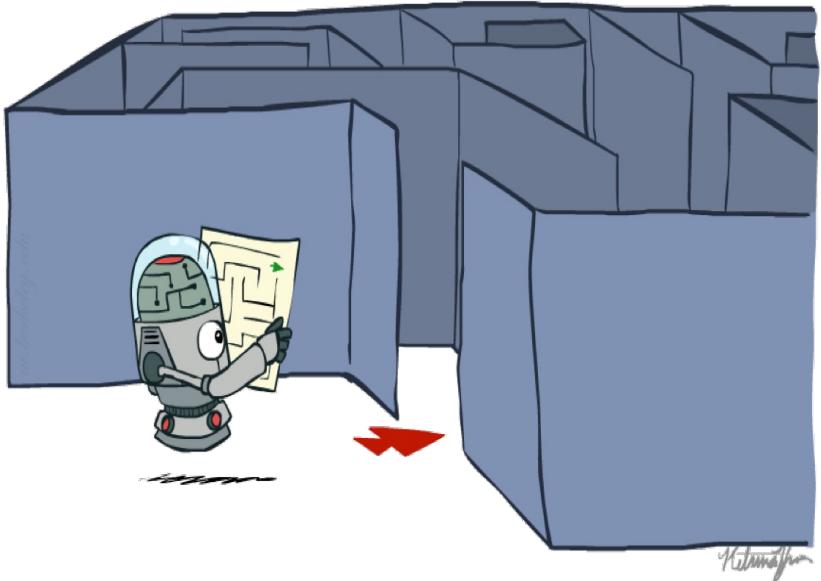
# Arbore de căutare - exemple



Frontiera (nodurile albe) separă regiunea explorată a stărilor (nodurile negre) de regiunea neexplorată (nodurile gri). În (a), numai nodul rădăcină a fost expandat. În (b), un nod frunză din cei patru succesorii ai nodului rădăcină a fost expandat. În (c), toți succesorii nodului rădăcină au fost expandați în ordinea acelor de ceasornic.

# Căutare

- Problemă de căutare
  - stări (configurații ale lumii)
  - acțiuni și costuri
  - funcția succesor
  - stare inițială și stare scop
- Arboare de căutare
  - noduri: soluții parțiale
  - arce: acțiuni
  - o soluție parțială are asociat un cost (suma costurilor acțiunilor = arcele)
- Strategie de căutare
  - construiește un arbore de căutare
  - alege ordinea în care explorează/expandează nodurile din frontieră
  - optimale dacă găsesc soluțiile de cost minim



# Strategii de căutare

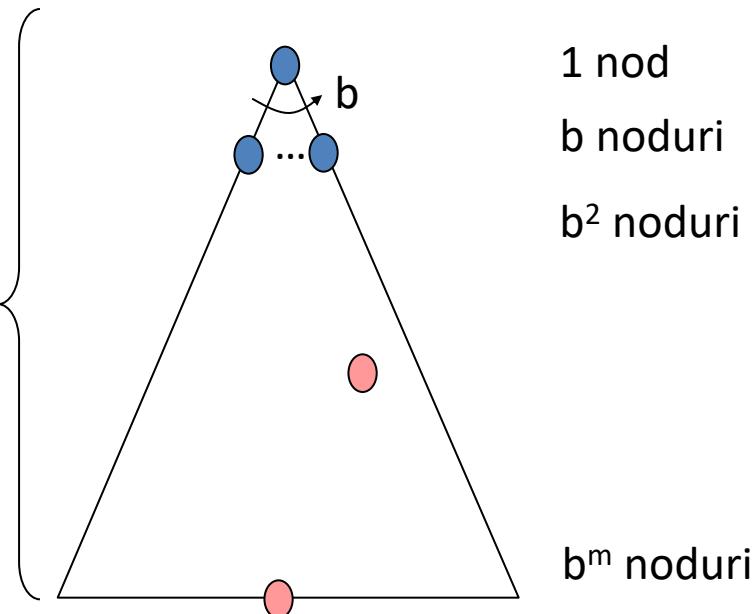
**Strategie** = alegerea unei ordini de explorare a nodurilor

Strategile de căutare se evaluatează conform următoarelor patru criterii:

1. **Compleitudine**: dacă, atunci când o soluție există, strategia dată garantează găsirea acesteia
2. **Complexitate a timpului**: durata în timp pentru găsirea unei soluții
3. **Complexitate a spațiului**: necesitățile de memorie pentru efectuarea căutării
4. **Optimalitate**: atunci când există mai multe soluții, strategia dată o găsește pe cea mai bună dintre ele (pe baza unei funcții – spre exemplu găsește funcția cu costul cel mai mic)

# Strategii de căutare

- Completitudine: garantarea unei soluții dacă ea există?
- Optimalitate: garantarea unei soluții de cost minim?
- Complexitate timp?
- Complexitate spațiu?
- Ilustrare arbore de căutare:
  - $b = \text{factor ramificare (branching factor)}$
  - $m = \text{adâncimea (nivelul) maximă}$
  - soluții la adâncimi diferite
- Numărul de noduri în arbore?
  - $1 + b + b^2 + \dots + b^m = O(b^m)$

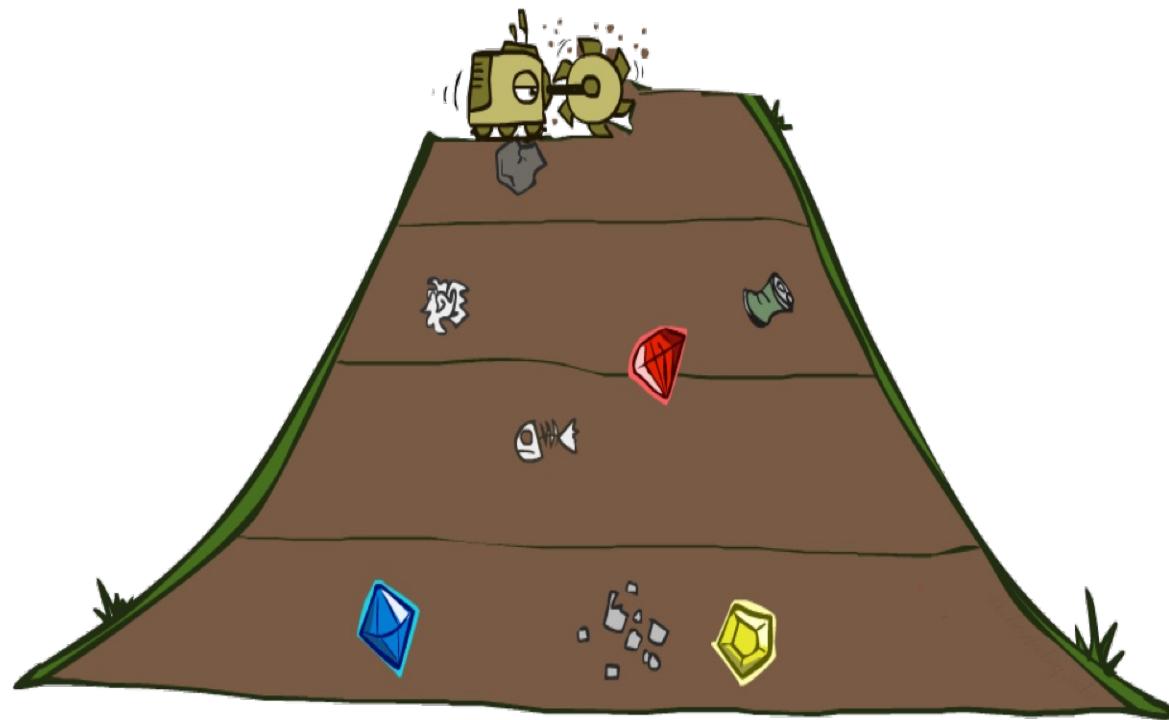


# Strategii de căutare neinformată

Strategii neinformate – folosesc numai informație disponibilă din definirea problemei. Pot genera succesiuni și distinge dacă o stare este scop sau. Din acest motiv se mai numesc și strategii de căutare ***oarbe***.

- căutare în lățime (bread-first search)
- căutare în adâncime (depth-first search)
- căutare cu cost uniform (uniform-cost search)
- căutare cu adâncime limitată (depth-limited search)
- căutare cu adâncime incrementală (iterative deepening search)

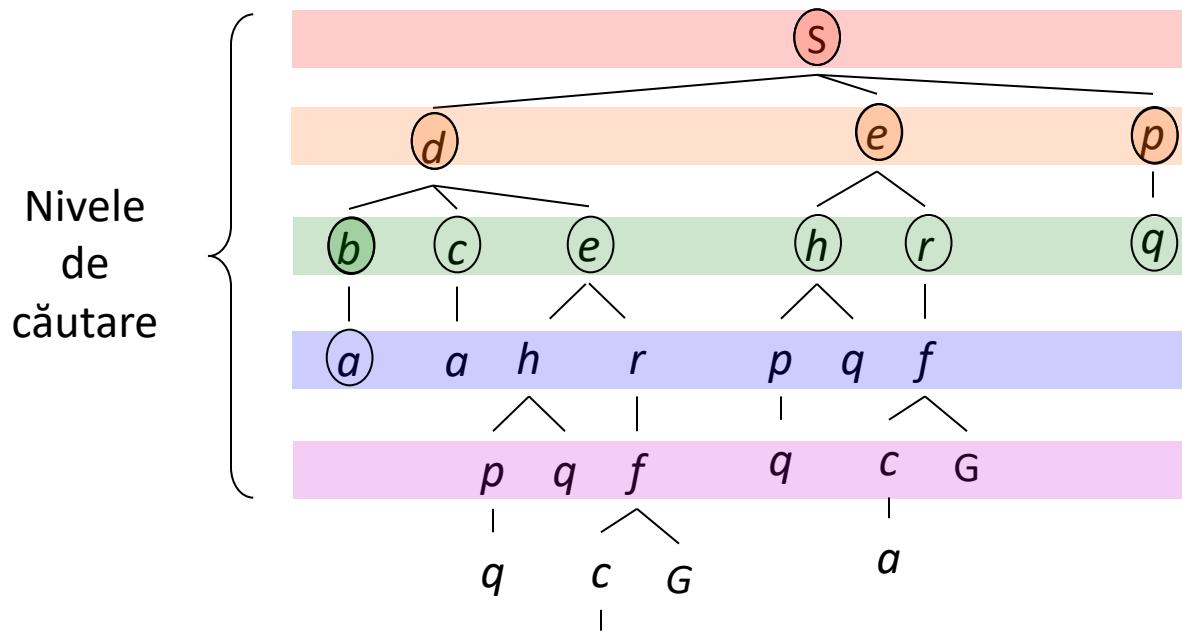
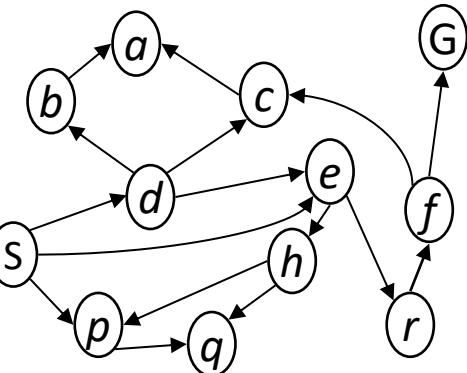
# Căutare în lățime (bread-first = BF)



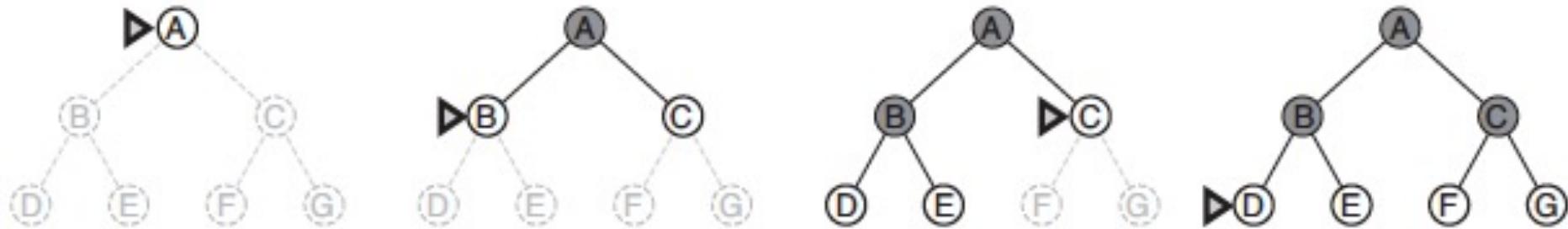
# Căutare în lățime (bread-first = BF)

Strategie: expandează succesiv toate nodurile în ordinea adâncimii începând cu nodul rădăcină

Frontiera este implementată cu o coadă (FIFO)



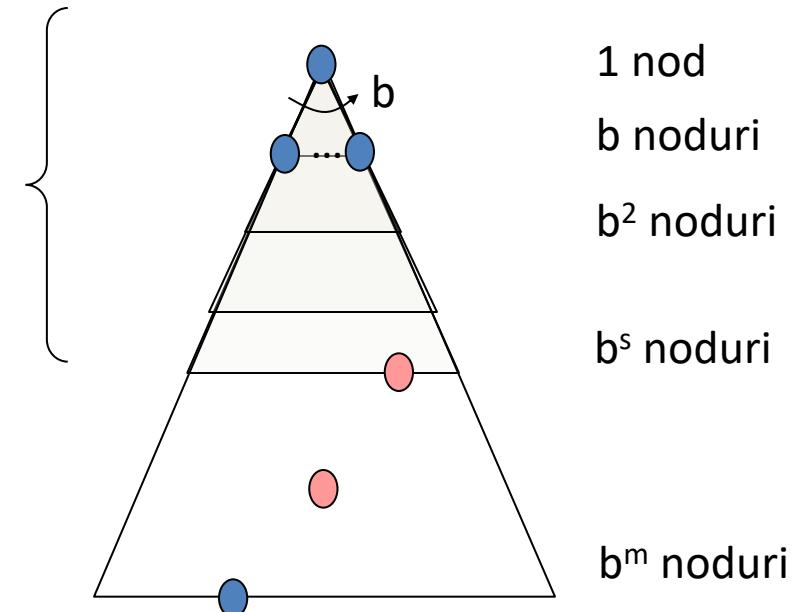
# Căutare în lățime pe un arbore binar



**Căutare în lățime pe un arbore binar:** la fiecare pas, săgeata indică nodul care va fi expandat.

# Proprietățile căutării în lățime

- Ce fel de noduri expandează BF?
  - procesează toate nodurile deasupra soluției de adâncime minimă
  - fie  $s =$  adâncimea minimă a unui nod scop
  - complexitate timp  $O(b^s)$
- Cât de mult spațiu necesită BF?
  - este spațiul pentru memorarea frontierei  $O(b^s)$   
+ memorarea nodurilor explorate  $O(b^{s-1})$
  - limită superioară dată de ultimul nivel
  - complexitate  $O(b^s)$
- Completitudine?
  - dacă o soluție există atunci s este finit, deci DA!
- Optimalitate?
  - dacă costurile fiecărei muchii sunt egale ( $= 1$ ) atunci DA!



# Reducerea complexității pentru căutare în lățime

funcția ARBORE-CAUTARE (problema, strategie) returnează o soluție sau eșec

initializează frontiera folosind starea inițială a problemei

initializează multimea de noduri explorate cu multimea vidă  
ciclează

dacă frontiera este vidă atunci returnează eșec

alege un **nod frunză** conform **strategiei** și elimină-l din **frontieră**  
**dacă nodul** conține o **stare scop**

**atunci returnează** soluția corespunzătoare

adaugă nodul curent multimii de noduri explorate (vizitate)

expandează nodul ales, adăugând nodurile succesor generate în frontieră

dacă nodurile nu sunt în frontieră sau în multimea de noduri explorate

- pentru reducerea complexității timp pentru căutarea în lățime se verifică dacă un nod succesor generat este nod scop.
- dacă generăm noduri soluție la nivelul s, nu putem avea un nod de pe nivelul s+1 soluție
- reduce complexitatea timp de la  $O(b^{s+1})$  la  $O(b^s)$

# Reducerea complexității pentru căutare în lățime

funcția ARBORE-CAUTARE (problema, strategie) returnează o soluție sau eșec

initializează frontiera folosind starea inițială a problemei

**dacă starea inițială este o stare scop**

**atunci returnează** soluția corespunzătoare

initializează multimea de noduri explorate cu multimea vidă

ciclează

    dacă frontiera este vidă atunci returnează eșec

    alege un **nod frunză** conform **strategiei** și elimină-l din **frontieră**

    adaugă nodul curent multimii de noduri explorate (vizitate)

    expandează nodul ales, adăugând nodurile succesor generate în frontieră

        dacă nodurile nu sunt în frontieră sau în multimea de noduri explorate

**dacă nodul succesor generat** conține o **stare scop**

**atunci returnează** soluția corespunzătoare

- pentru reducerea complexității timp pentru căutarea în lățime se verifică dacă un nod succesor generat este nod scop.
- dacă generăm noduri soluție la nivelul s, nu putem avea un nod de pe nivelul s+1 soluție
- reduce complexitatea timp de la  $O(b^{s+1})$  la  $O(b^s)$

# Timpul și memoria necesară pentru căutarea în lățime - BF.

Considerăm:

- factor de ramificare (branching factor)  $b = 10$ ;
- timp de procesare = 1 milion de noduri/secundă;
- necesară de memorie pentru stocarea unui nod = 1000 bytes

Adâncime	#Noduri	Timp	Memorie
2	110	.11 ms	107 kB
4	11110	11 ms	10.6 MB
6	$10^6$	1.1 sec	1 GB
8	$10^8$	2 min	103 GB
10	$10^{10}$	3 ore	10 TB
12	$10^{12}$	13 zile	1 PB
14	$10^{14}$	3.5 ani	99 PB
16	$10^{16}$	350 ani	10 EB

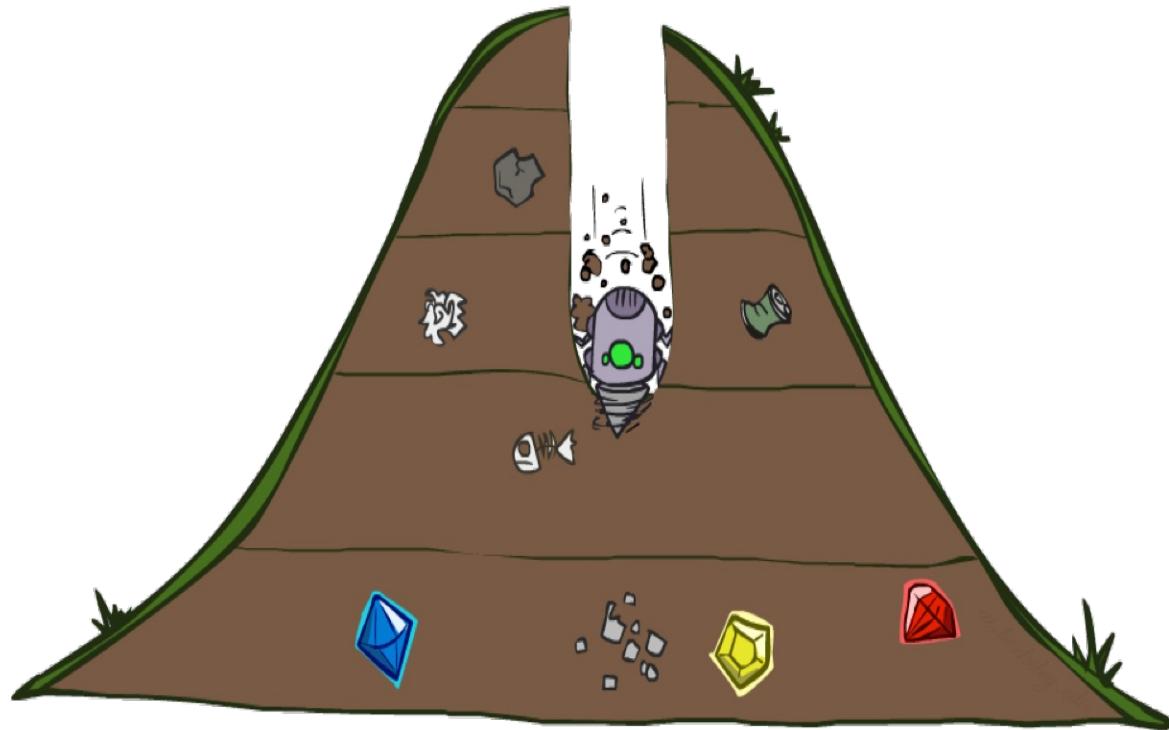
# Timpul și memoria necesară pentru căutarea în lățime - BF.

Adâncime	#Noduri	Timp	Memorie
2	110	.11 ms	107 kB
4	11110	11 ms	10.6 MB
6	$10^6$	1.1 sec	1 GB
8	$10^8$	2 min	103 GB
10	$10^{10}$	3 ore	10 TB
12	$10^{12}$	13 zile	1 PB
14	$10^{14}$	3.5 ani	99 PB
16	$10^{16}$	350 ani	10 EB

Remarci:

- complexitate exponentială timp și spațiu
- timp de execuție aproape infinit pentru adâncimi relativ moderate

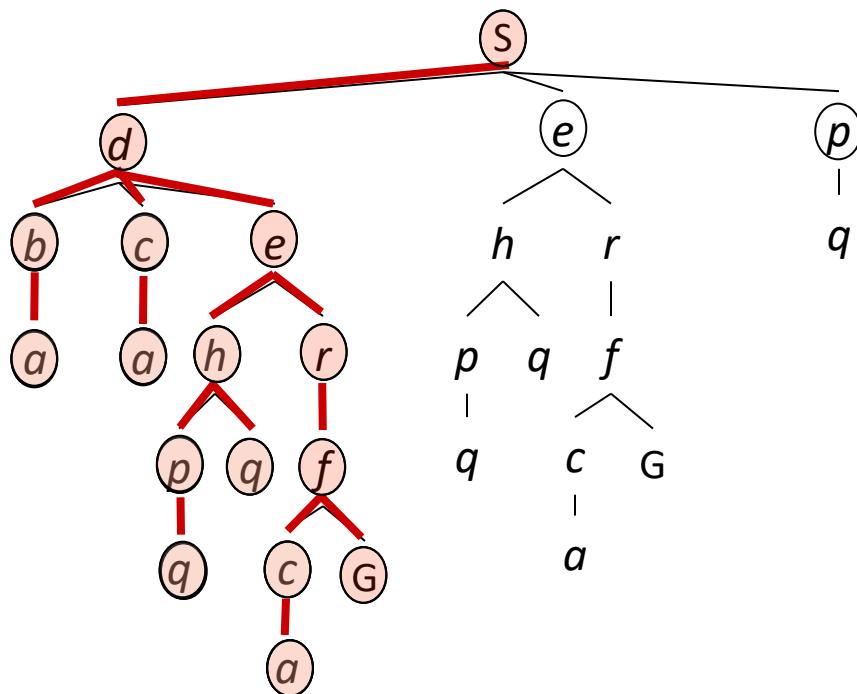
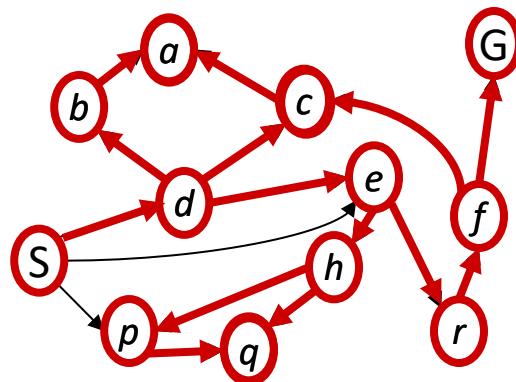
# Căutarea în adâncime (depth-first = DF)



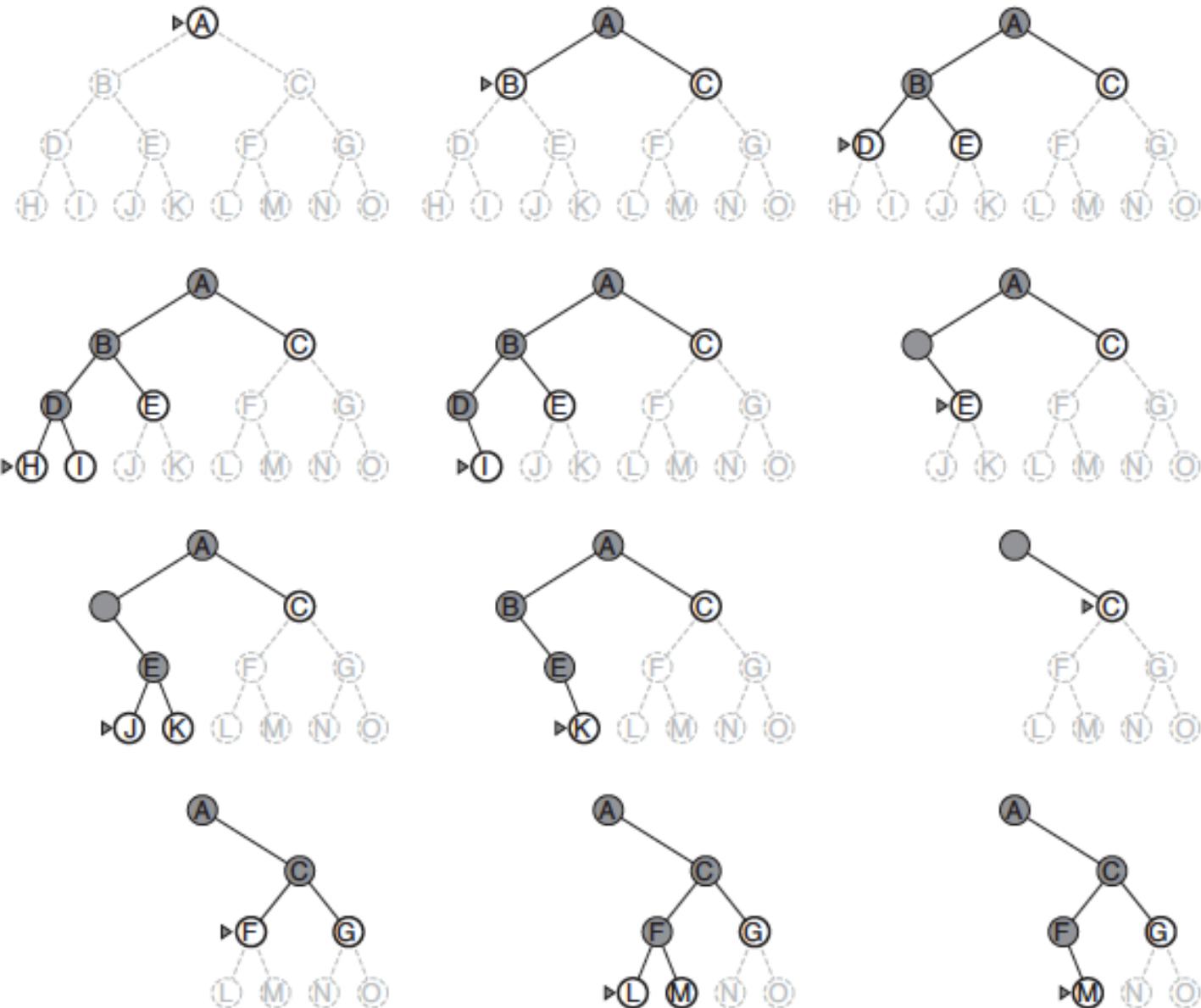
# Căutarea în adâncime (depth-first = DF)

Strategie: expandează nodul curent de adâncime maximă

Frontiera este implementată cu o stivă (LIFO)

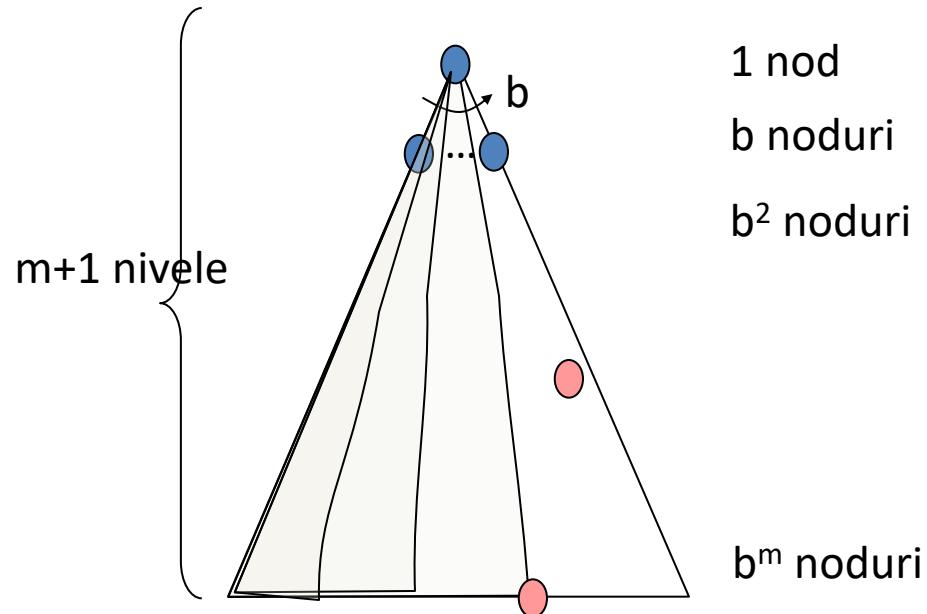


# Căutare în adâncime pe un arbore binar

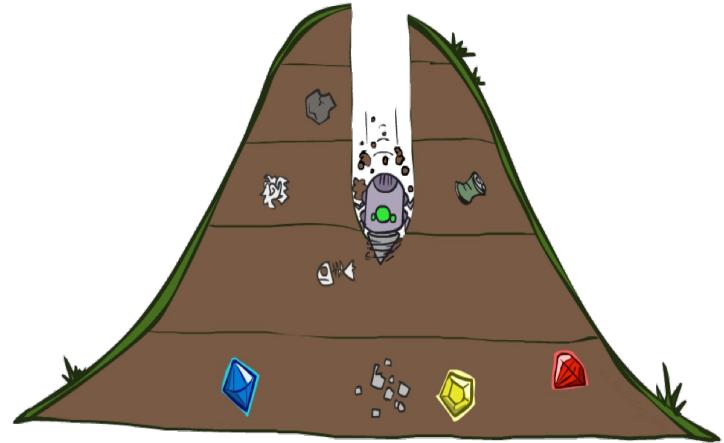
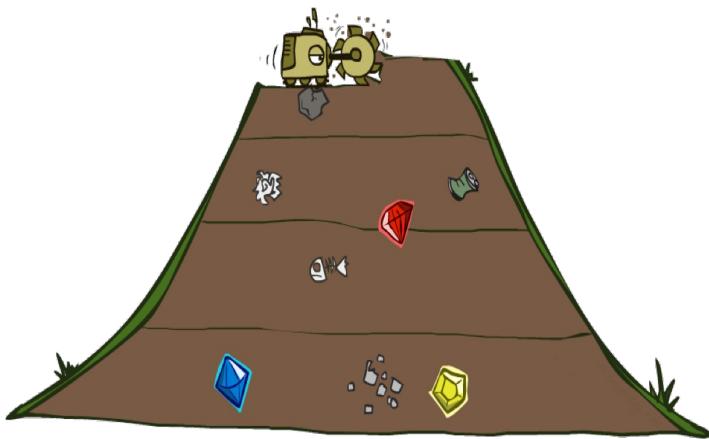


# Proprietățile căutării în adâncime

- Ce fel de noduri expandează DF?
  - partea stângă prefix din arbore
  - poate procesa întreg arborele!
  - complexitate timp  $O(b^m)$
- Cât de mult spațiu necesită DF?
  - numai fiii de la rădăcină spre frunză neexpandați, deci  $O(bm)$
- Completitudine?
  - $m$  ar putea fi infinit, în acest caz NU!
  - $m$  finit + dacă se țin minte nodurile vizitate DA!
- Optimalitate?
  - NU, găsește “cea mai din stânga” soluție, indiferent de adâncime sau cost

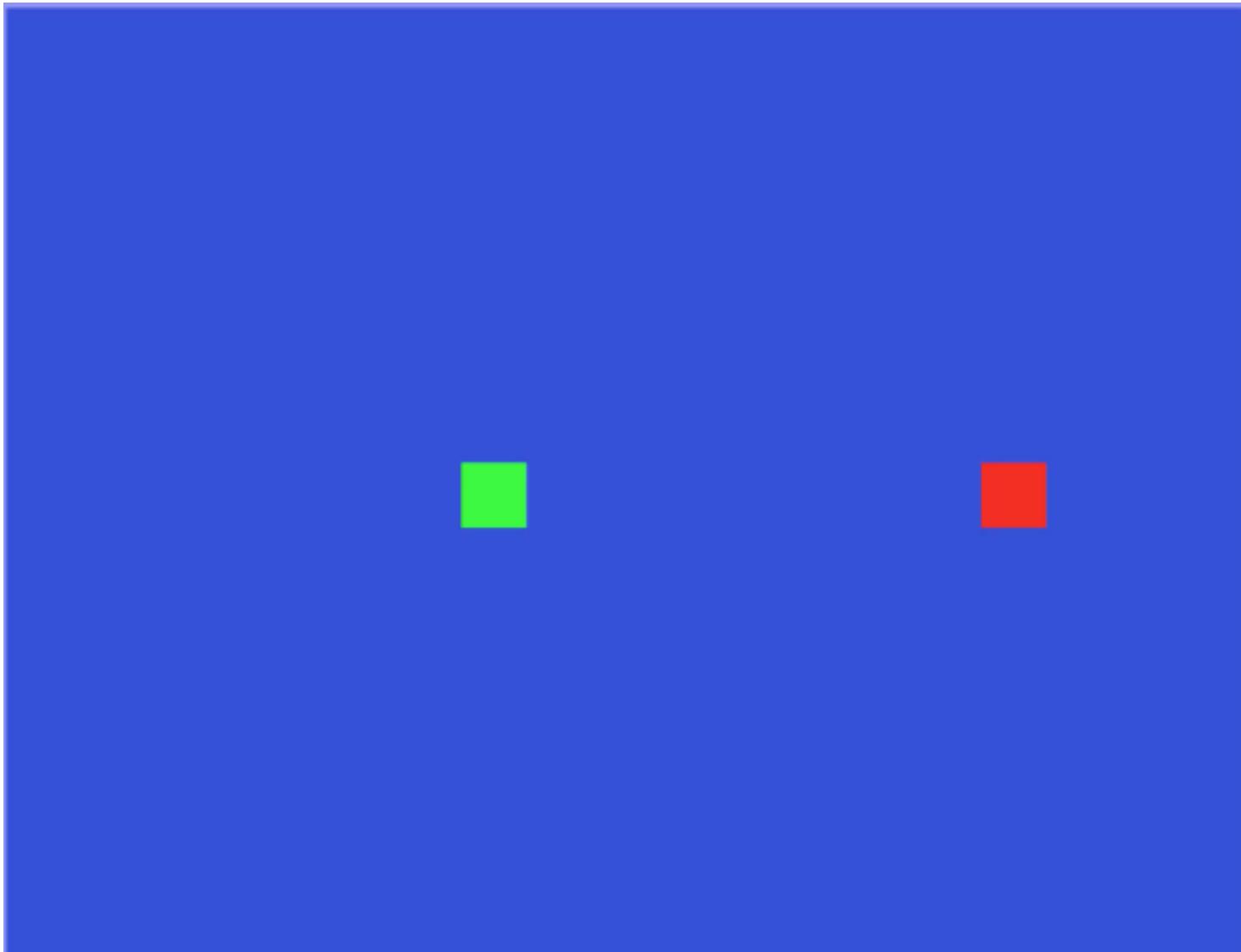


# BF vs DF

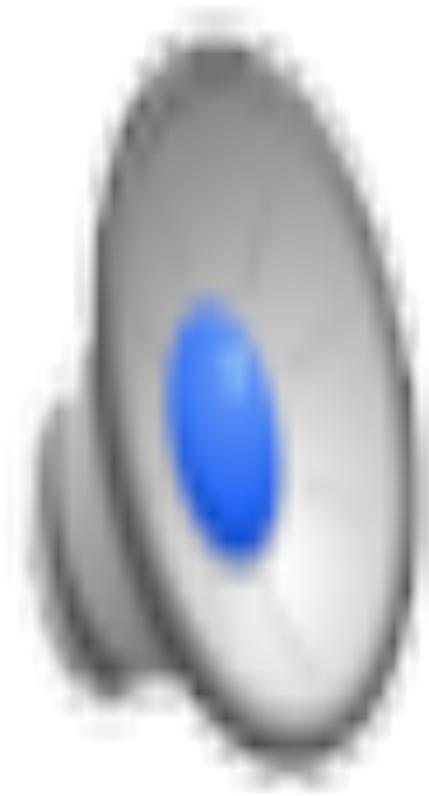


- Când BF găsește o soluție mai bună decât DF?
- Când DF găsește o soluție mai bună decât BF?

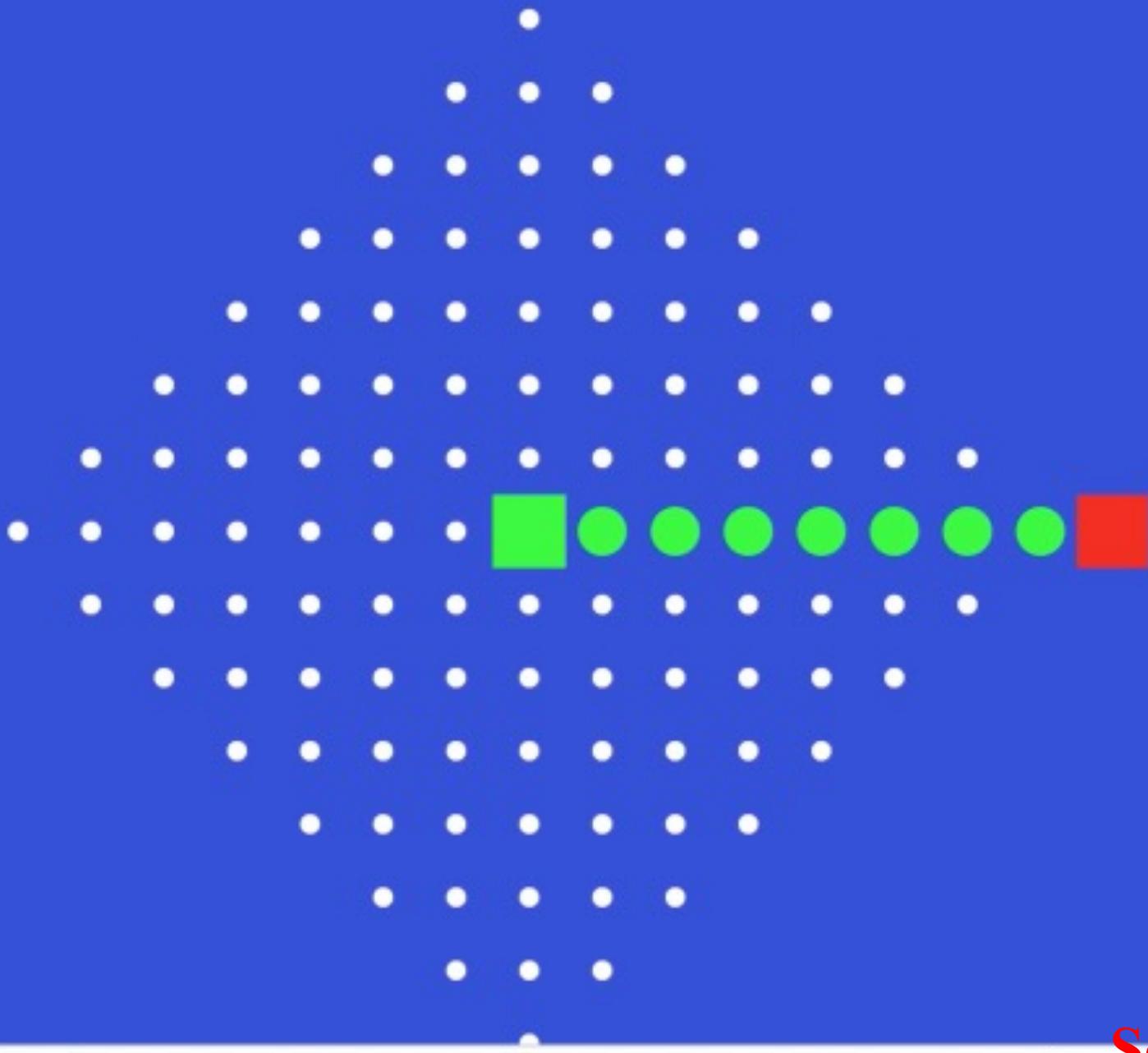
# BF vs DF



# BF vs DF



**Soluția BF**

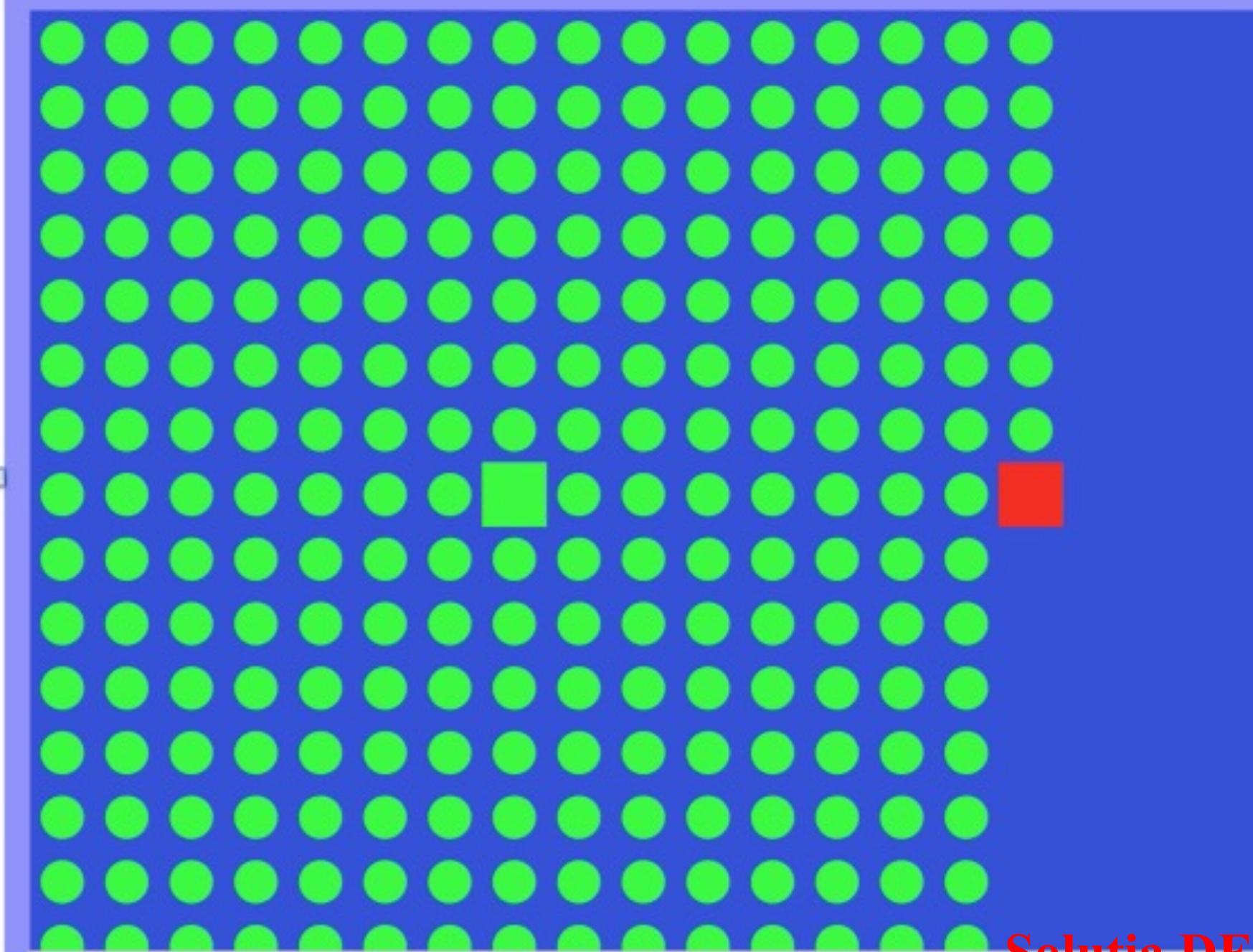


Soluția BF

# BF vs DF



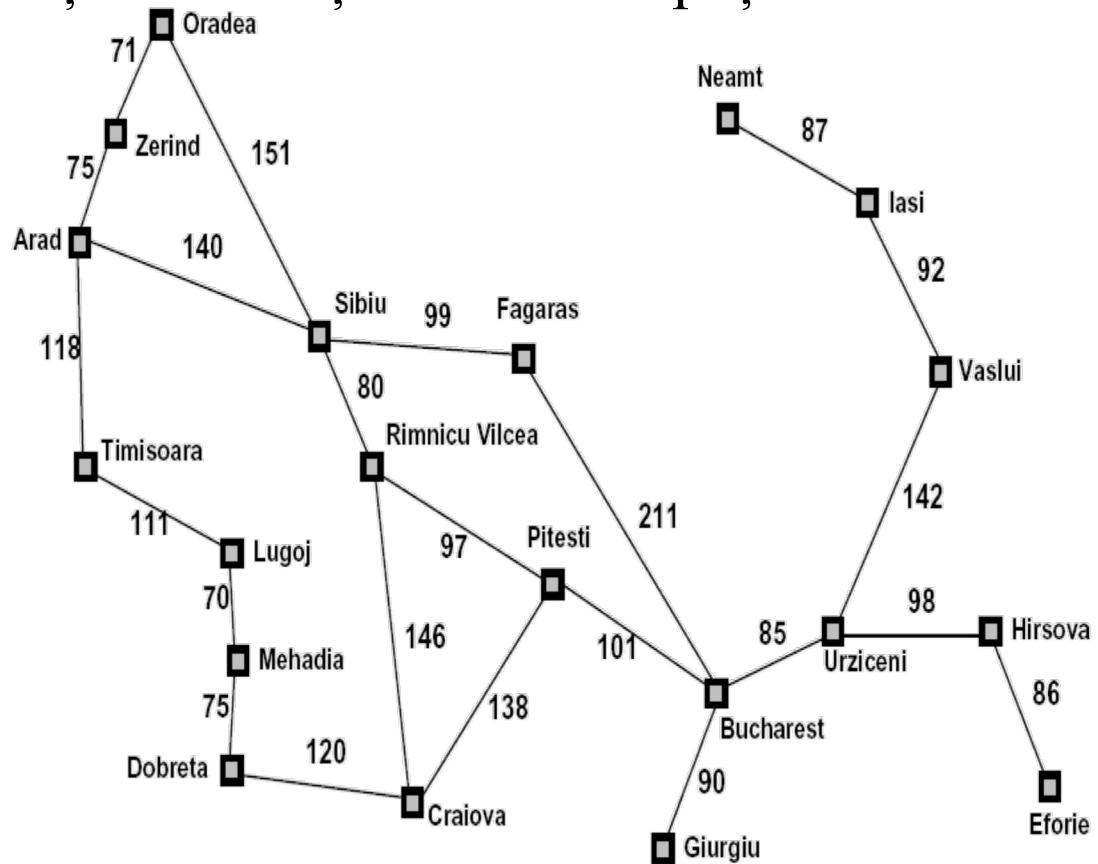
**Soluția DF**



Soluția DF

# Căutare în adâncime limitată

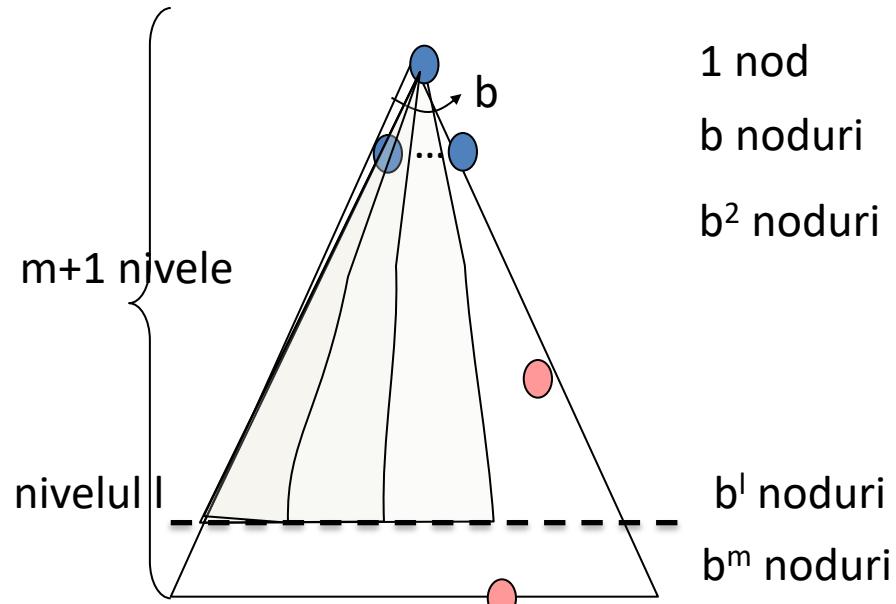
- căutarea în adâncime poate eșua pentru spațiu de stări infinit
- harta are 20 de orașe
- adâncimea maximă poate fi considerată 19
- pot ajunge din orice oraș în alt oraș în maxim 9 pași
- diametru = 9
- limitează DF la 9



# Căutare în adâncime limitată

Căutare în adâncime până la un anumit nivel 1 (depth-limited DL).

- Ce fel de noduri expandează DL?
  - partea stângă prefix din arbore, numai până la nivelul 1
  - complexitate timp  $O(b^l)$
- Cât de mult spațiu necesită DL?
  - fiii de la rădăcină spre frunză + nodurile neexplorate, deci  $O(bl)$
- Completitudine?
  - NU, poate rata soluția dacă l este prea mic
- Optimalitate?
  - NU, găsește “cea mai din stânga” soluție de adâncime maximă l, indiferent de cost



# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 10

# Recapitulare – cursul trecut

## 1. Rezolvarea problemelor prin căutare

- Graful stărilor
- Arborele de căutare

## 2. Căutare neinformată

- Căutare în lățime (Bread-Fist Search)
- Căutare în adâncime (Depth-Fist Search)
- Căutare în adâncime limitată (depth-limited search)

# Strategii de căutare neinformată

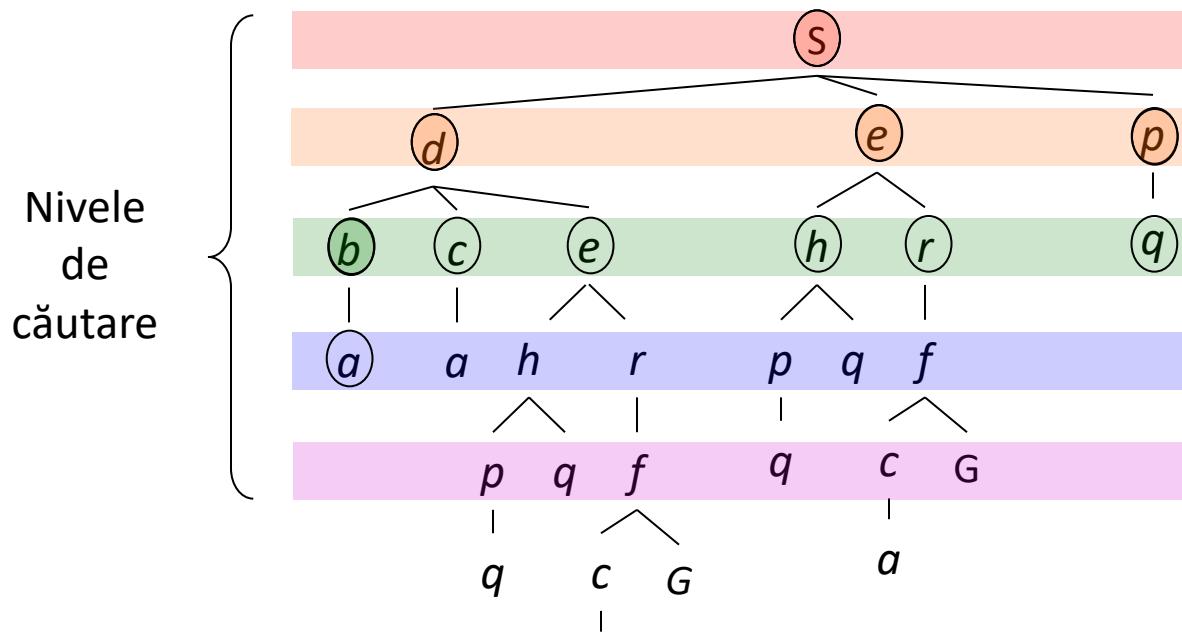
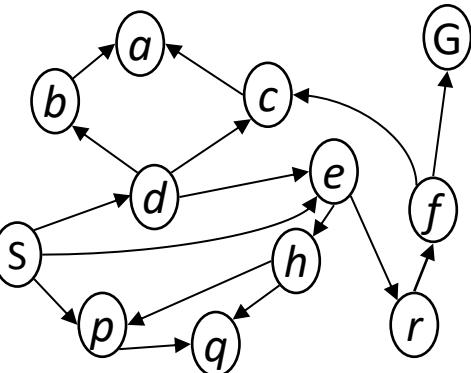
Strategii neinformate – folosesc numai informație disponibilă din definirea problemei. Pot genera succesiuni și distinge dacă o stare este scop sau. Din acest motiv se mai numesc și strategii de căutare ***oarbe***.

- căutare în lățime (bread-first search)
- căutare în adâncime (depth-first search)
- căutare cu cost uniform (uniform-cost search)
- căutare cu adâncime limitată (depth-limited search)
- căutare cu adâncime incrementală (iterative deepening search)

# Căutare în lățime (bread-first = BF)

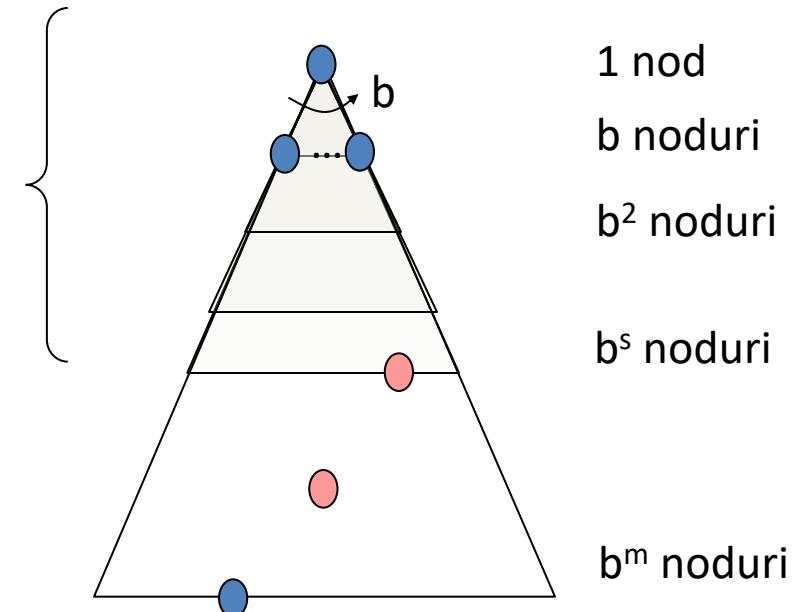
Strategie: expandează succesiv toate nodurile în ordinea adâncimii începând cu nodul rădăcină

Frontiera este implementată cu o coadă (FIFO)



# Proprietățile căutării în lățime

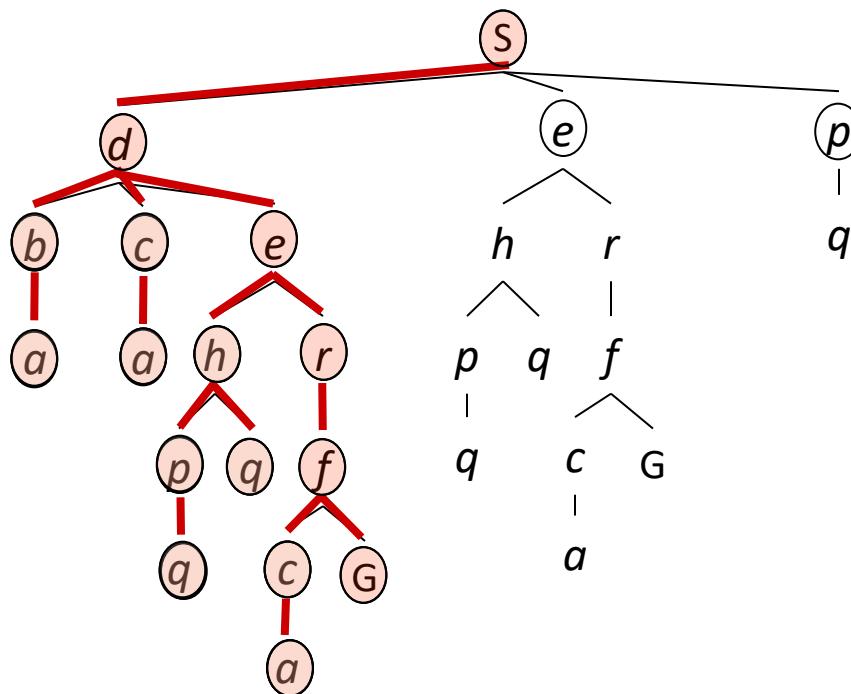
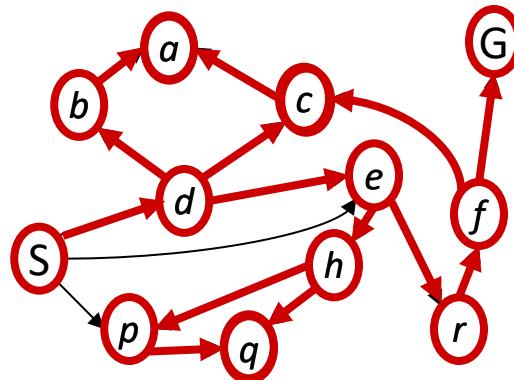
- Ce fel de noduri expandează BF?
  - procesează toate nodurile deasupra soluției de adâncime minimă
  - fie  $s =$  adâncimea minimă a unui nod scop
  - complexitate timp  $O(b^s)$
- Cât de mult spațiu necesită BF?
  - este spațiul pentru memorarea frontierei  $O(b^s)$   
+ memorarea nodurilor explorate  $O(b^{s-1})$
  - limită superioară dată de ultimul nivel
  - complexitate  $O(b^s)$
- Completitudine?
  - dacă o soluție există atunci s este finit, deci DA!
- Optimalitate?
  - dacă costurile fiecărei muchii sunt egale ( $= 1$ ) atunci DA!



# Căutarea în adâncime (depth-first = DF)

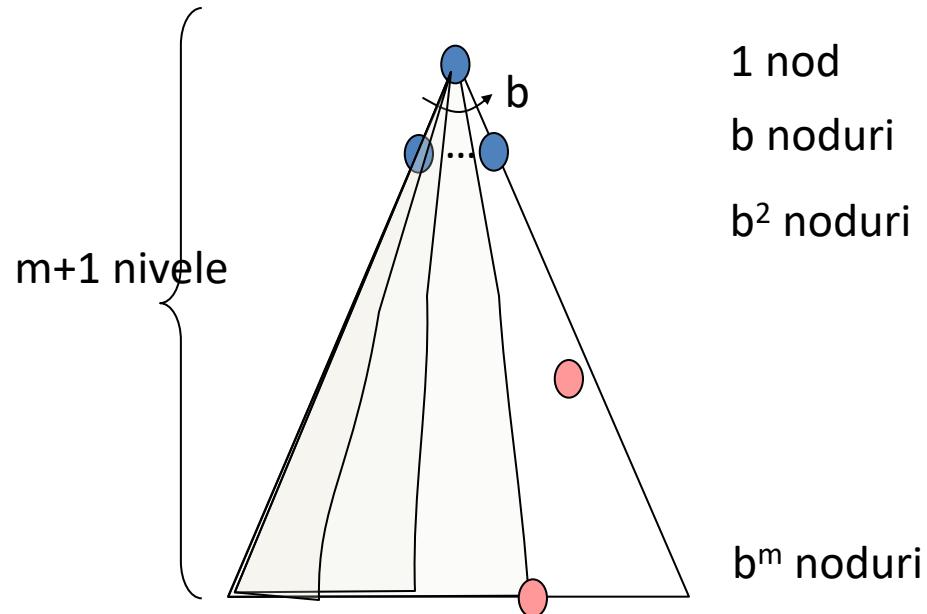
Strategie: expandează nodul curent de adâncime maximă

Frontiera este implementată cu o stivă (LIFO)



# Proprietățile căutării în adâncime

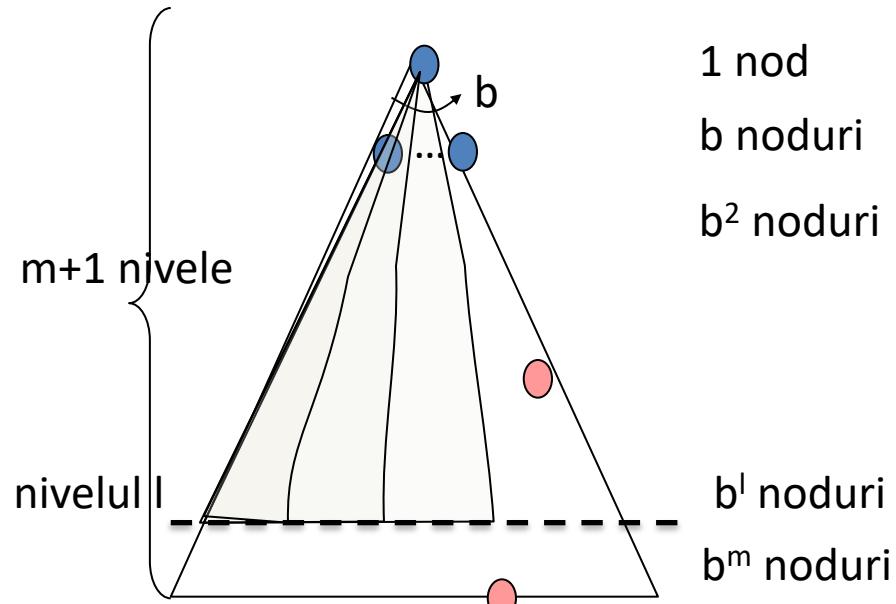
- Ce fel de noduri expandează DF?
  - partea stângă prefix din arbore
  - poate procesa întreg arborele!
  - complexitate timp  $O(b^m)$
- Cât de mult spațiu necesită DF?
  - numai fiii de la rădăcină spre frunză neexpandați, deci  $O(bm)$
- Completitudine?
  - $m$  ar putea fi infinit, în acest caz NU!
  - $m$  finit + dacă se țin minte nodurile vizitate DA!
- Optimalitate?
  - NU, găsește “cea mai din stânga” soluție, indiferent de adâncime sau cost



# Căutare în adâncime limitată

Căutare în adâncime până la un anumit nivel 1 (depth-limited DL).

- Ce fel de noduri expandează DL?
  - partea stângă prefix din arbore, numai până la nivelul 1
  - complexitate timp  $O(b^l)$
- Cât de mult spațiu necesită DL?
  - fiii de la rădăcină spre frunză + nodurile neexplorate, deci  $O(bl)$
- Completitudine?
  - NU, poate rata soluția dacă l este prea mic
- Optimalitate?
  - NU, găsește “cea mai din stânga” soluție de adâncime maximă l, indiferent de cost



# Cuprinsul cursului de azi

## 1. Căutare neinformată

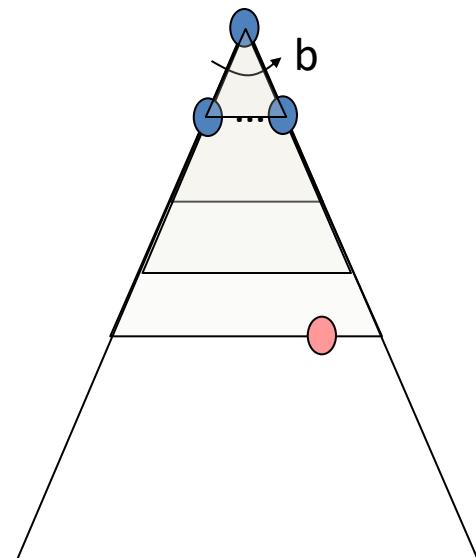
- Căutare în adâncime incrementală (iterative deepening search)
- Căutare uniformă după cost (uniform-cost search)

## 2. Căutare informată

- căutare Greedy
- algoritmul A\*
- euristici admisibile, dominante și banale

# Căutare incrementală în adâncime

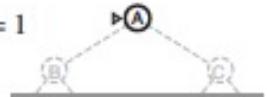
- Combină căutarea în adâncime (DF) cu căutarea în lățime (BF):
  - Rulează DF cu adâcimea limitată la 1.  
Dacă nu găsește o soluție ...
  - Rulează DF cu adâcimea limitată la 2.  
Dacă nu găsește o soluție ...
  - Rulează DF cu adâcimea limitată la 3.  
Dacă nu găsește o soluție ...
  - Rulează DF cu adâcimea limitată la 4....



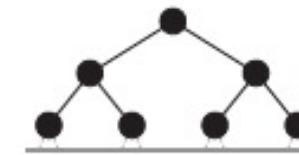
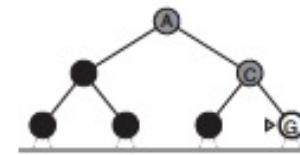
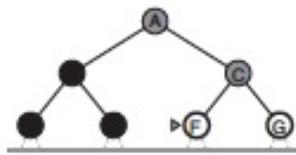
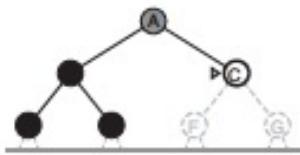
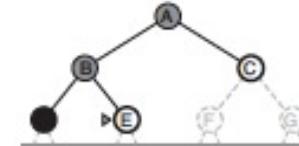
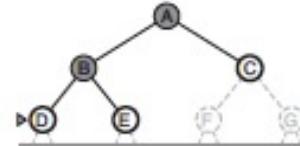
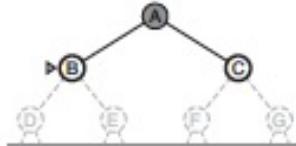
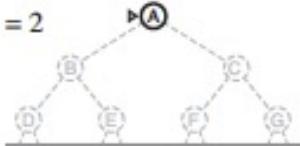
Limit = 0



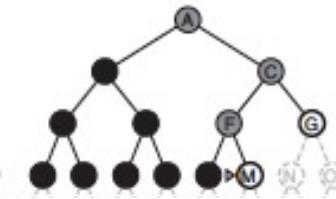
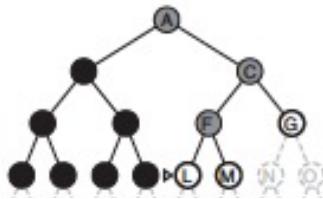
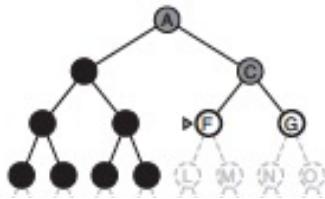
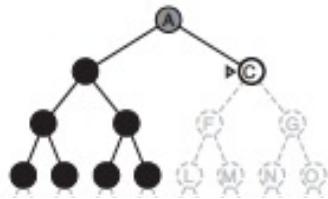
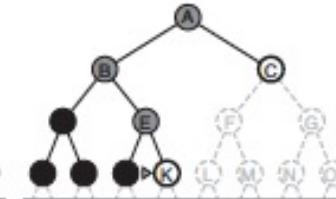
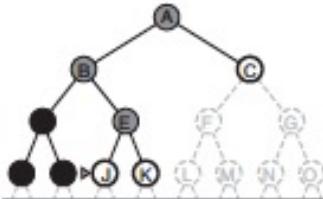
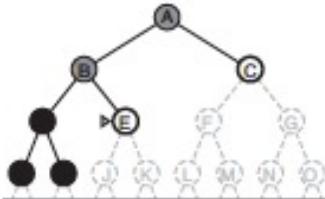
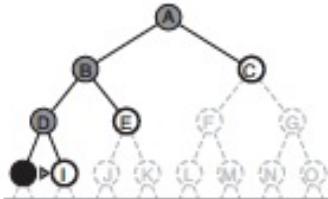
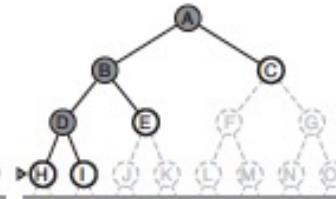
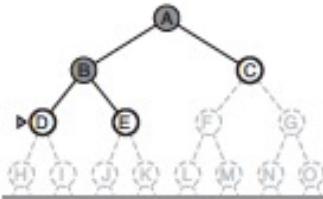
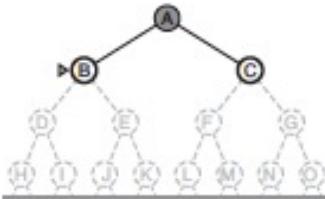
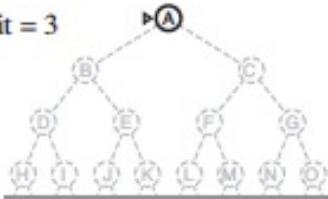
Limit = 1



Limit = 2



Limit = 3



# Căutare incrementală în adâncime

Ce fel de noduri expandează IDS?

BF + DF până la nivelul s al soluției de adâncime minimă

Complexitate timp  $O(b^s)$

$N(IDS) = s * b + (s-1) * b^2 + \dots + b^s$  (noduri vizitate de IDS)

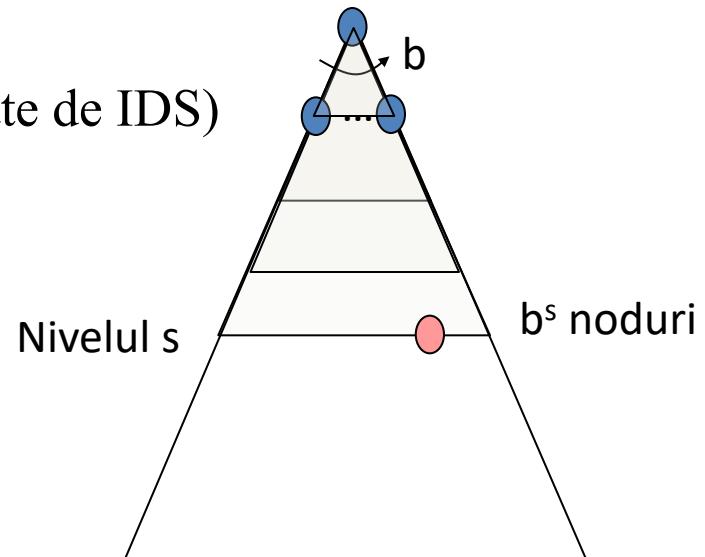
$N(BF) = b + b^2 + \dots + b^s$  (noduri vizitate de BF)

Pentru  $b = 10$ ,  $s = 5$ :

$N(IDS) = 50 + 400 + \dots + 100000 = 123450$

$N(BF) = 10 + 100 + \dots + 100000 = 111110$

*Nu este foarte redundant*



Cât de mult spațiu necesită DL?

fii de la rădăcină spre frunză + cele neexplorate, deci  $O(bs)$

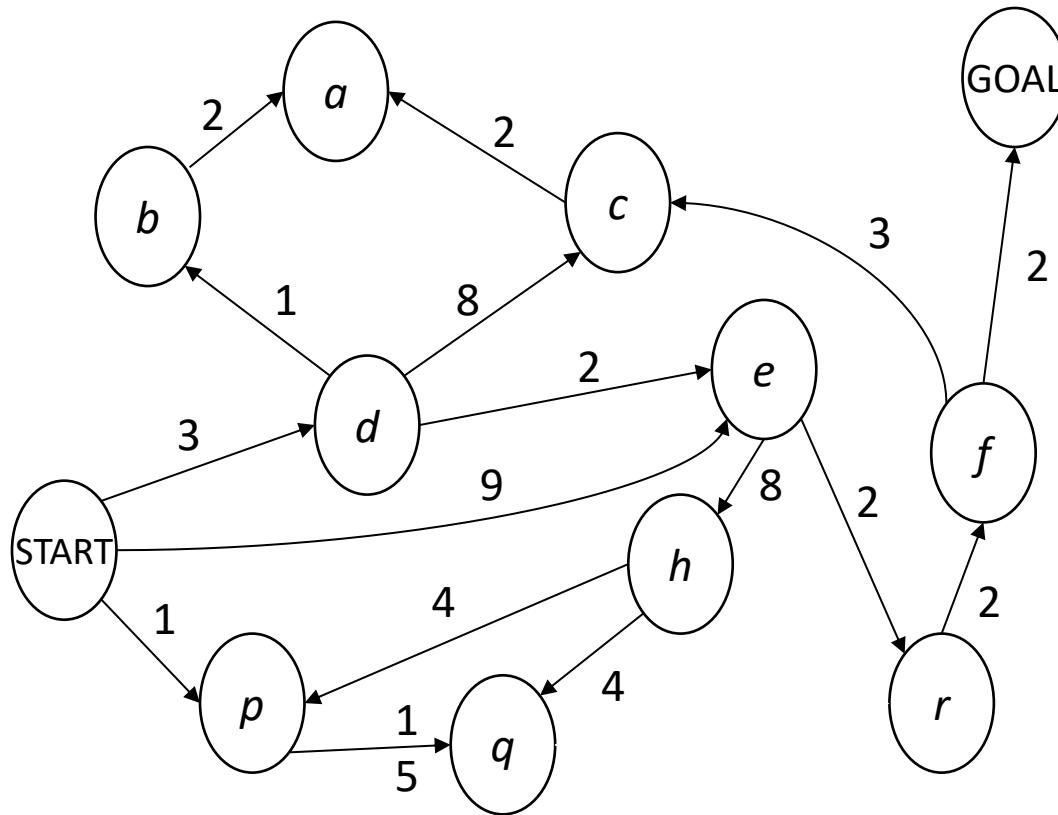
Compleitudine? (de la BF)

dacă o soluție există atunci s este finit, deci DA!

Optimalitate? (de la BF)

dacă costurile fiecărei muchii sunt egale (= 1) atunci DA!

# Căutare în funcție de cost

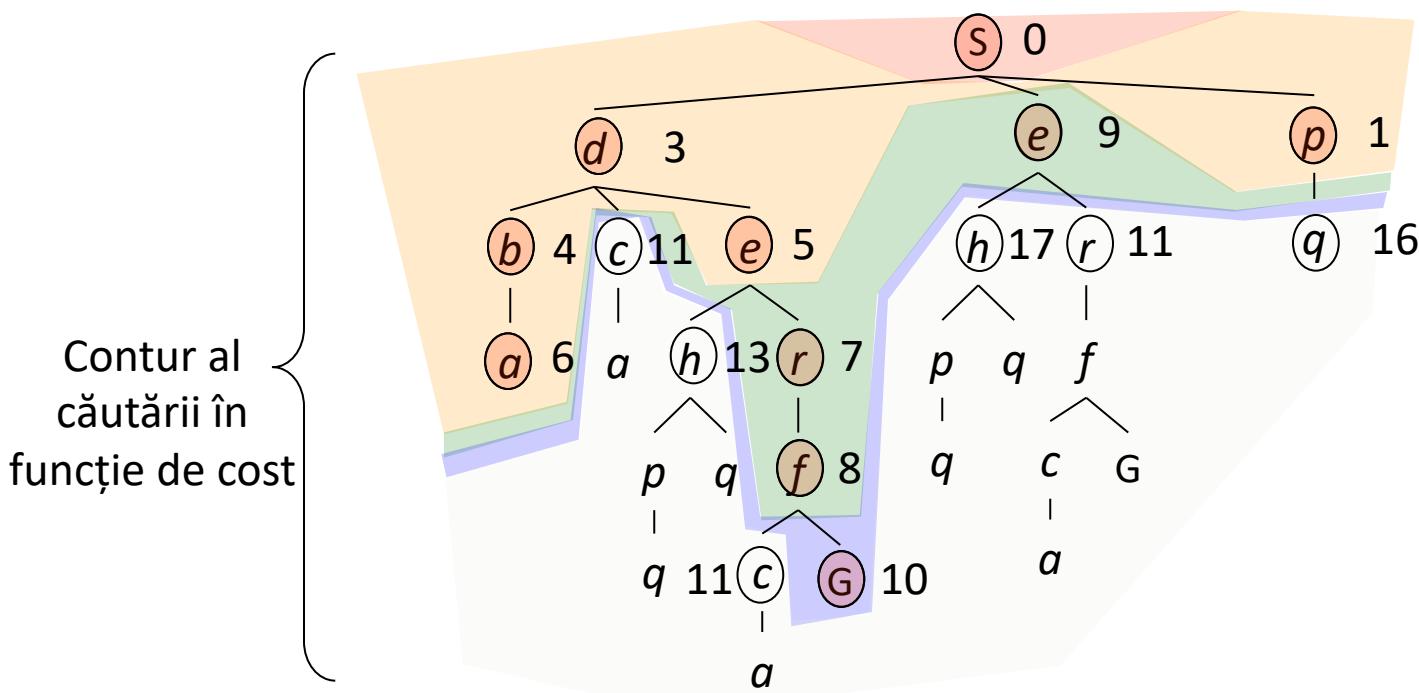
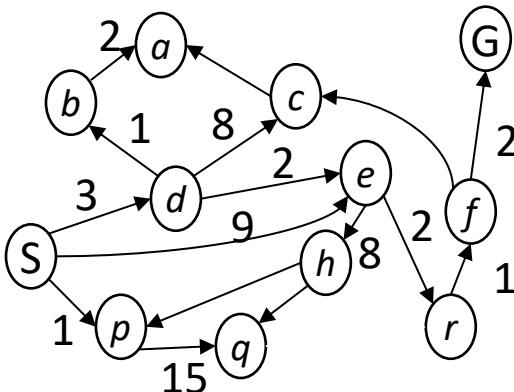


BF găsește cel mai scurt drum relativ la numărul de arce (acțiuni). Nu ia în considerare nici un cost (implicit fiecare arc are același cost). Studiem algoritmi de căutare pe bază de cost.

# Căutare uniformă după cost (uniform cost search – UCS)

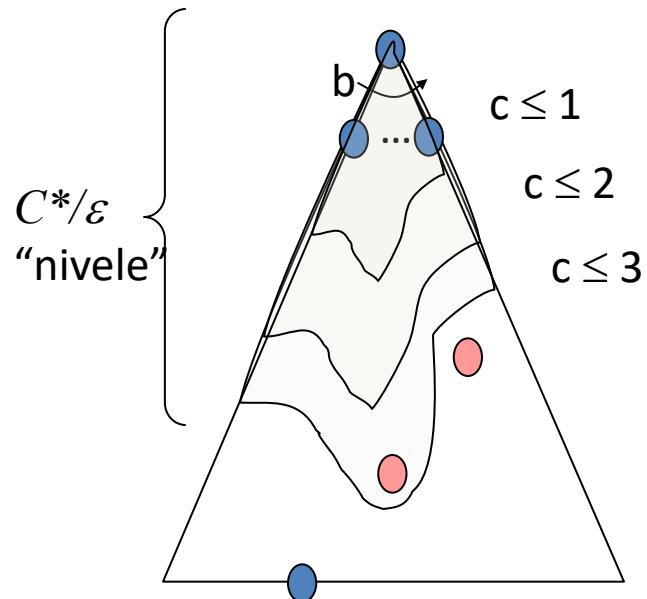
*Strategie: expandează nodul care face parte dintr-un drum de cost minim la pasul curent.*

*Frontiera este implementată cu o coadă de priorități (prioritate: costul minim al unui drum curent)*



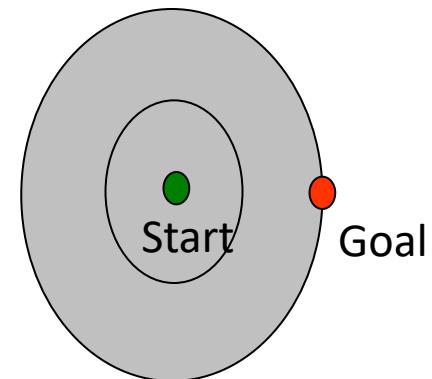
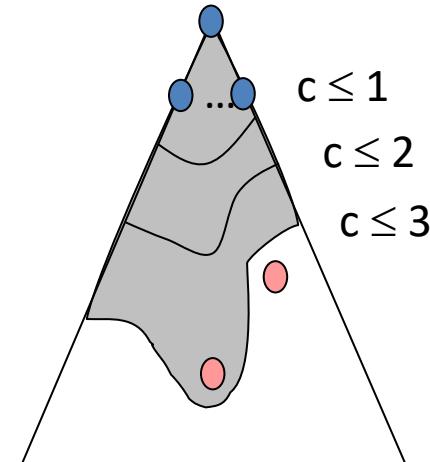
# Proprietăți ale căutării uniforme după cost

- Ce noduri expandază UCS?
  - procesează toate nodurile cu cost mai mic decât soluția de cost minim!
  - dacă soluția de cost minim are costul  $C^*$  iar fiecare arc costă cel puțin  $\varepsilon$ , atunci ajungem la o adâncime în jur de  $C^*/\varepsilon$
  - complexitate timp  $O(b^{C^*/\varepsilon})$  (exponențială în adâncime)
- Complexitate spațiu (a memoriei pentru frontieră)?
  - La fel ca la BF, aproximativ reține toate nodurile de pe ultimul nivel, deci  $O(b^{C^*/\varepsilon})$
- Completitudine?
  - DA (dacă soluția cea mai bună are cost finit iar costurile arcelor sunt pozitive)
- Optimalitate?
  - DA! (demonstrația se leagă de algoritmul A\*)

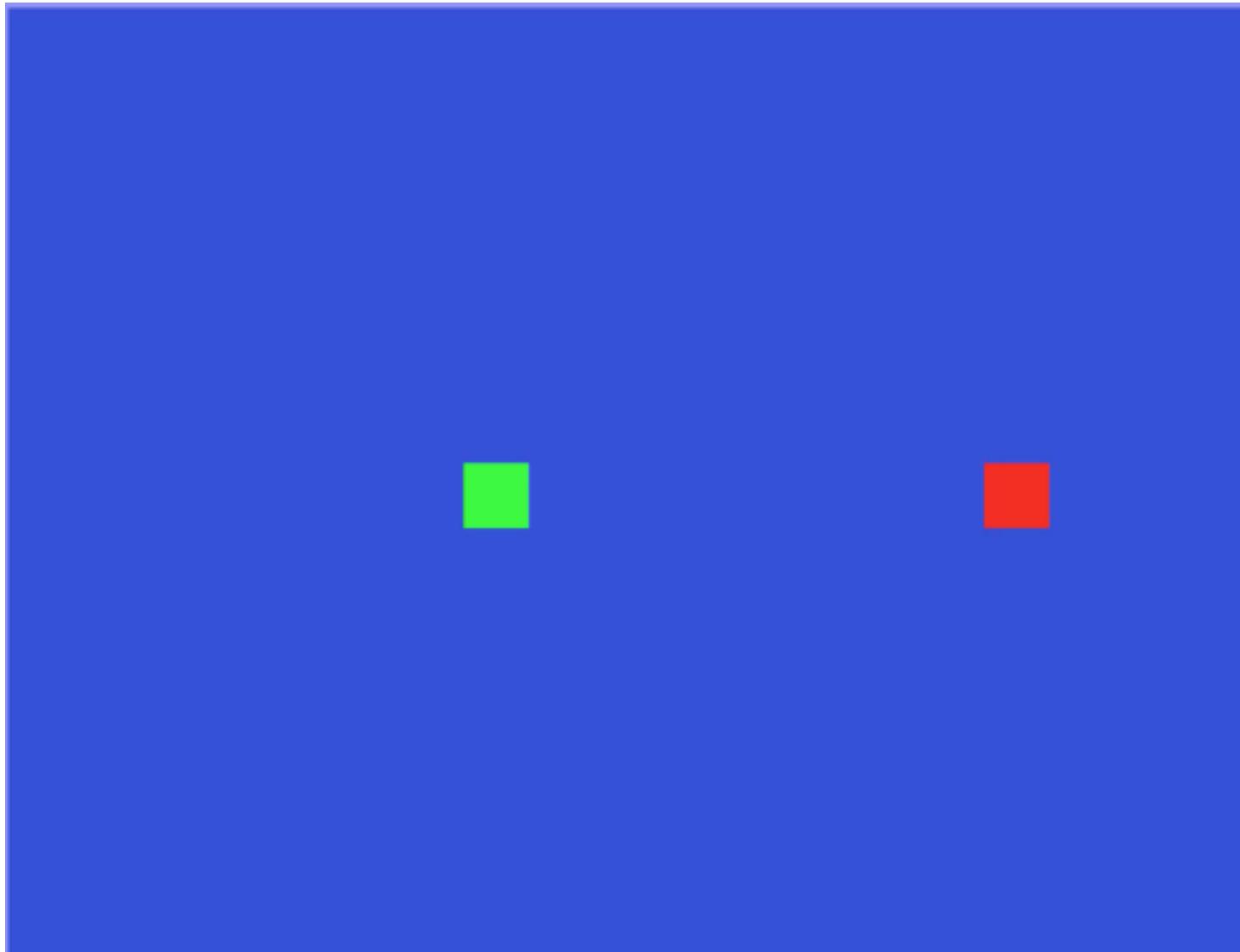


# Avantaje/dezavantaje ale căutării uniforme după cost

- UCS explorează drumuri în ordinea costurilor
- Avantaje: UCS este complet și optimal!
- Dezavantaje:
  - explorează drumurile în toate direcțiile (căutare uniformă)
  - nu are nicio informație despre poziția nodului scop



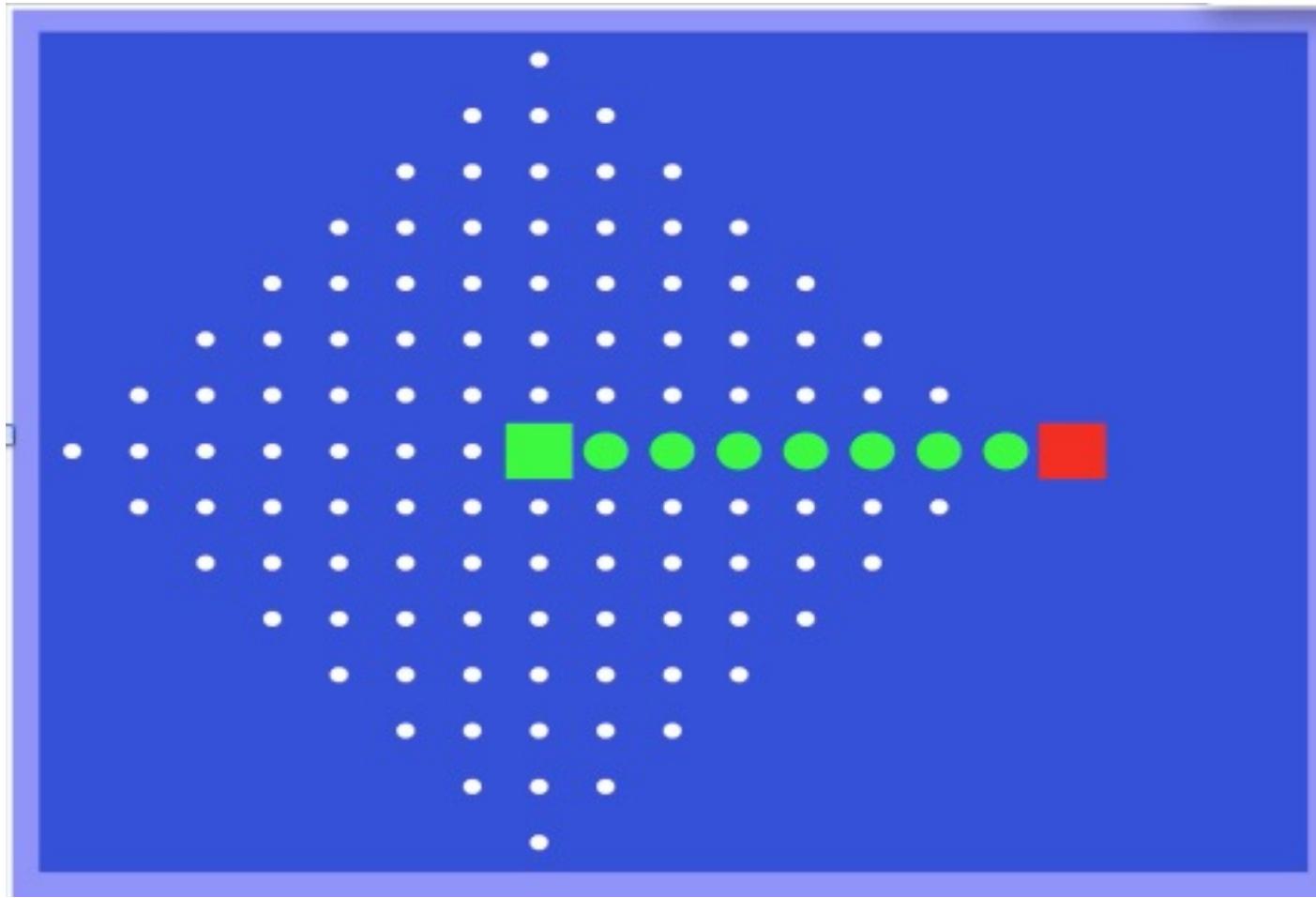
# UCS vs BF vs DF



# Căutare uniformă după cost - demo

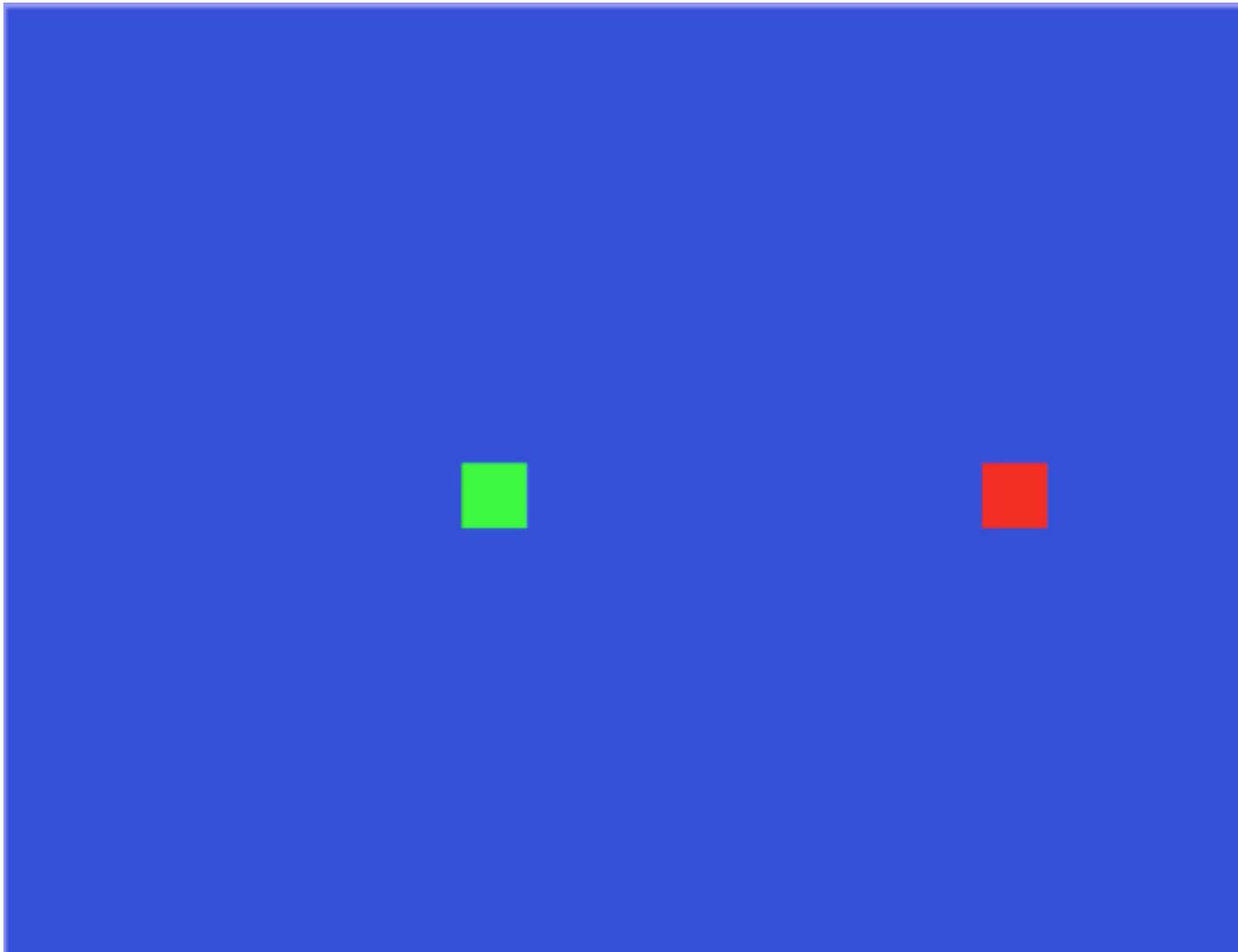


# Căutare uniformă după cost - demo



**Soluția UCS**

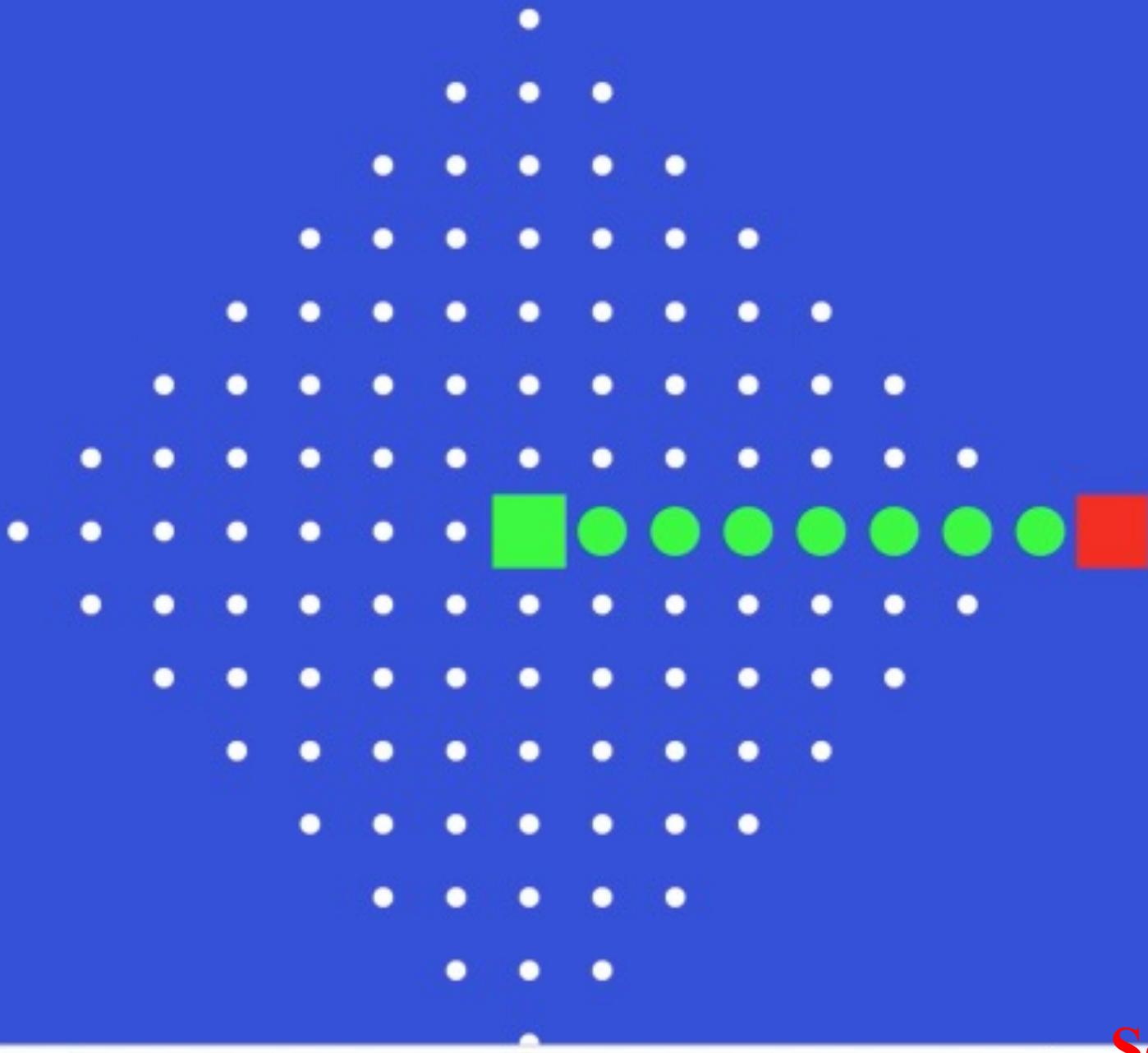
# BF vs DF



# BF vs DF



**Soluția BF**

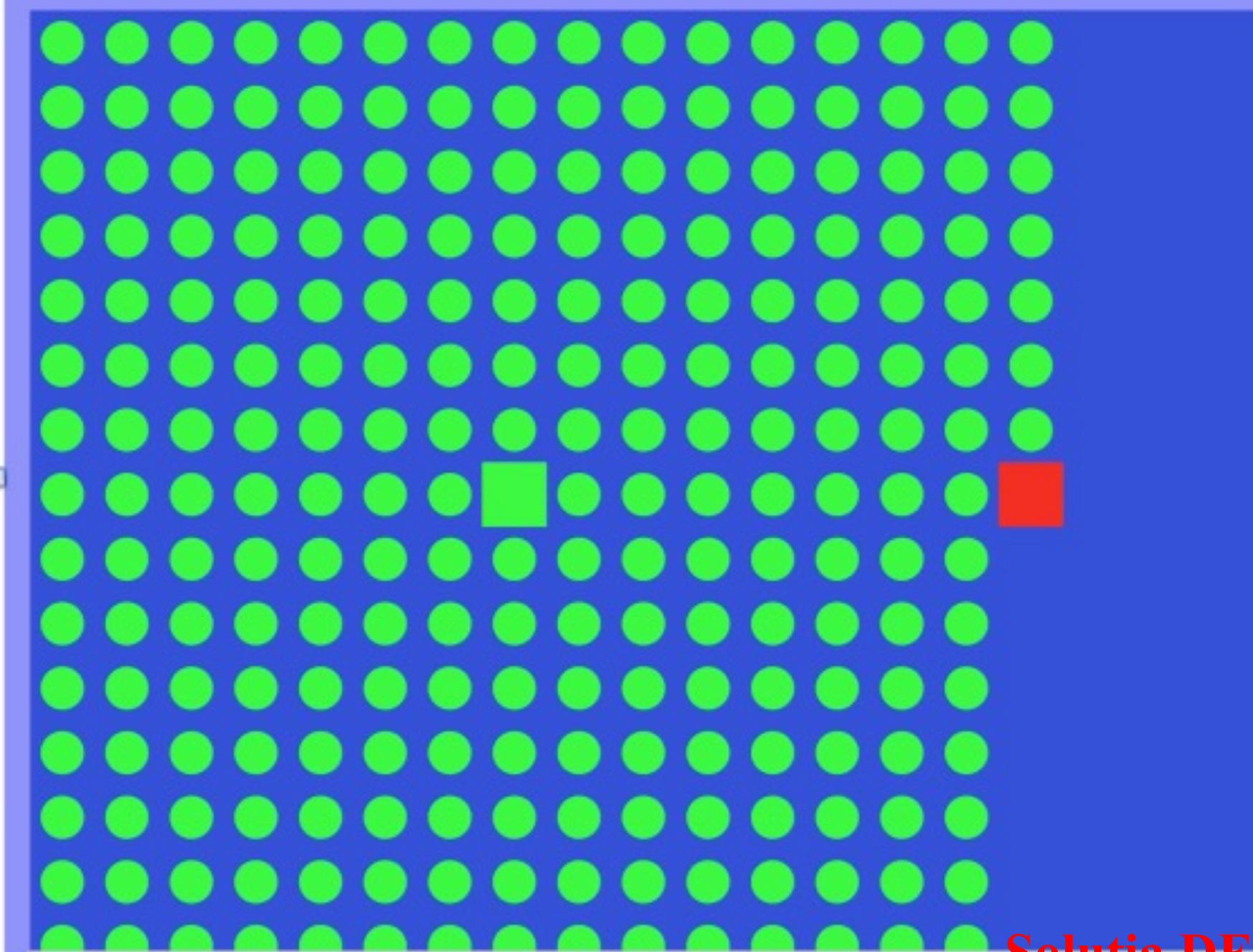


Soluția BF

# BF vs DF



**Soluția DF**



Soluția DF

# Compararea strategiilor de căutare neinformată

Criteriu	Căutare în lățime (BF)	Căutare uniformă pe bază de cost (UCS)	Căutare în adâncime (DF)	Căutare în adâncime limitată (DL)	Căutare iterativă în adâncime (ID)
Completitudine	DA <sup>a</sup>	DA <sup>a,b</sup>	NU	NU	DA <sup>a</sup>
Timp	O(b <sup>d</sup> )	O(b <sup>C*/ε</sup> )	O(b <sup>m</sup> )	O(b <sup>l</sup> )	O(b <sup>d</sup> )
Spațiu	O(b <sup>d</sup> )	O(b <sup>C*/ε</sup> )	O(bm)	O(bl)	O(bd)
Optimalitate	DA <sup>c</sup>	DA	NU	NU	DA <sup>c</sup>

b – branching factor (factor de ramificare)

d – adâncimea minimă a unei soluții

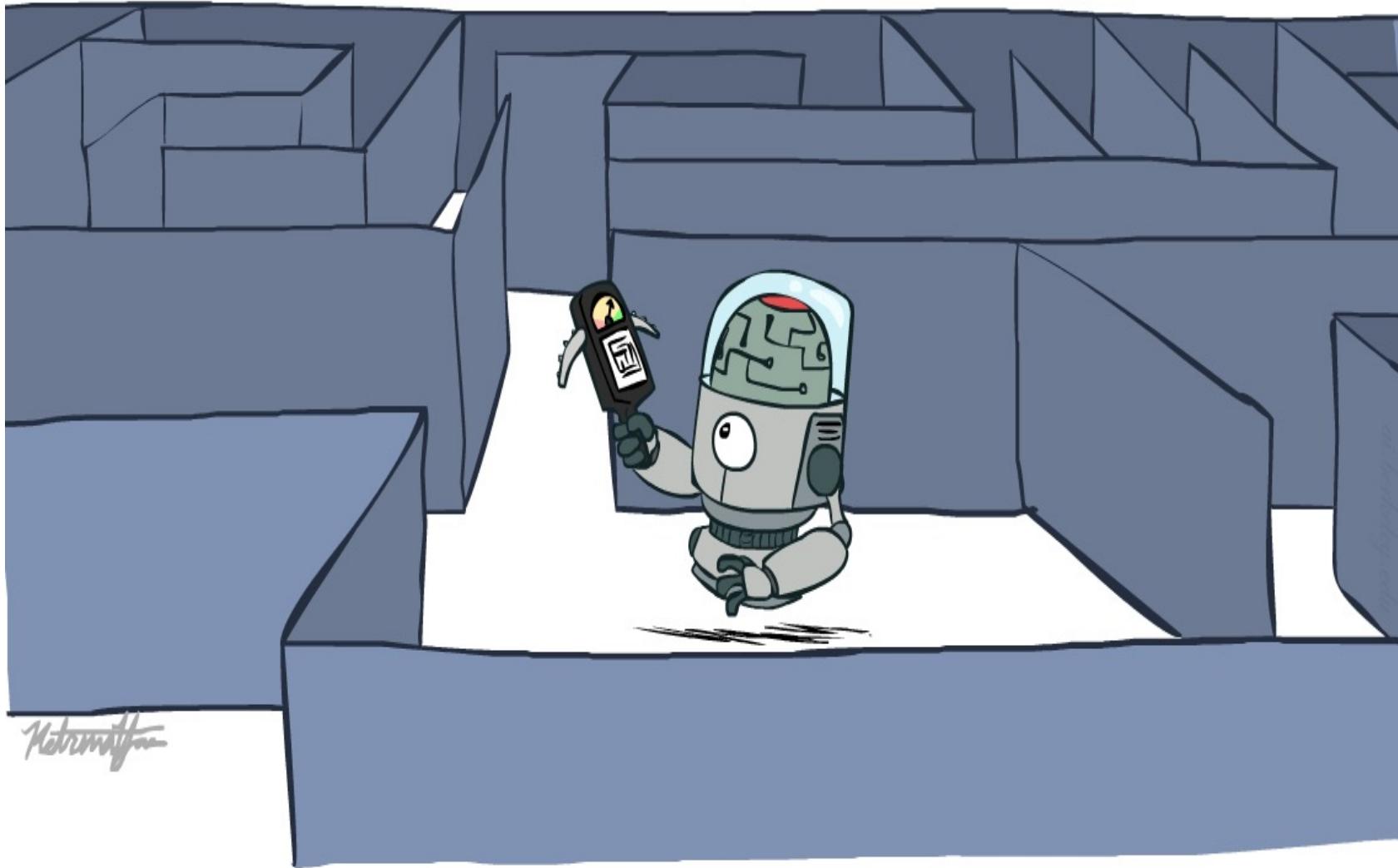
m – adâncimea maximă în arborele de căutare

DA<sup>a</sup> – complet dacă b este finit

DA<sup>b</sup> – complet dacă fiecare arc are costul  $\geq \varepsilon$

DA<sup>c</sup> – optimal dacă toate costurile sunt egale = 1

# Căutare informată



# Arbore de căutare general

funcția **ARBORE-CAUTARE** (problemă, strategie) returnează o soluție sau eșec

initializează **frontiera** folosind **starea inițială a problemei**

ciclează

dacă **frontiera** este vidă **atunci returnează** eșec

alege un **nod frunză** conform **strategiei** și elimină-l din **frontieră**

dacă **nodul** conține o **stare scop**

**atunci returnează** soluția corespunzătoare

expandează nodul ales, adăugând nodurile succesor generate în **frontieră**

**Frontiera** = structură de date ce păstrează toate nodurile care mai trebuie expandate

**Strategia de căutare** = precizează ordinea în care expandăm/explorăm nodurile

# Arbore de căutare general

funcția **ARBORE-CAUTARE** (problemă, strategie) returnează o soluție sau eșec

initializează **frontiera** folosind **starea inițială a problemei**  
initializează multimea de noduri explorate cu multimea vidă  
**ciclează**

dacă **frontiera** este vidă **atunci returnează** eșec

alege un **nod frunză** conform **strategiei** și elimină-l din **frontieră**

dacă **nodul** conține o **stare scop**

**atunci returnează** soluția corespunzătoare

adaugă nodul curent multimii de noduri explorate (vizitate)

expandează nodul ales, adăugând nodurile succesor generate în **frontieră**

dacă nodurile nu sunt în **frontieră** sau în multimea de noduri explorate

**Frontiera** = structură de date ce păstrează toate nodurile care mai trebuie expandate

**Strategia de căutare** = precizează ordinea în care expandăm/explorăm nodurile

# Căutare informată vs căutare neinformată

**Frontiera** = structură de date ce păstrează toate nodurile care mai trebuie expandate

**Strategia de căutare** = precizează ordinea în care expandăm/explorăm nodurile

- funcție de evaluare  $f(n)$  – evaluatează nodul  $n$
- alege un nod  $n$  pe baza lui  $f(n)$

## Căutare neinformată

- $f(n)$  nu codează informații specifice despre problemă
- BF:  $f(n) = \text{primul nod dintr-o coadă}$ , nod de adâncime minimă
- DF:  $f(n) = \text{primul nod dintr-o stivă}$ , nod de adâncime maximă
- UCS:  $f(n) = \text{costul nodului}$ , listă de priorități

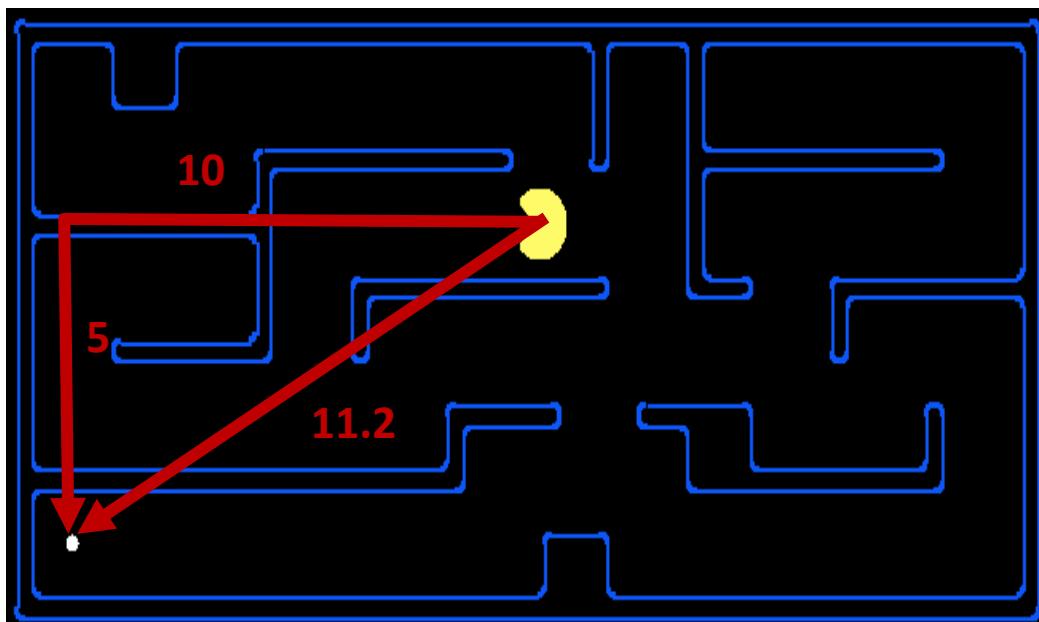
## Căutare informată

- $f(n)$  codează informații specifice despre problemă
- includem în  $f(n)$  și euristici (nu le avem la căutare neinformată)

# Euristici de căutare

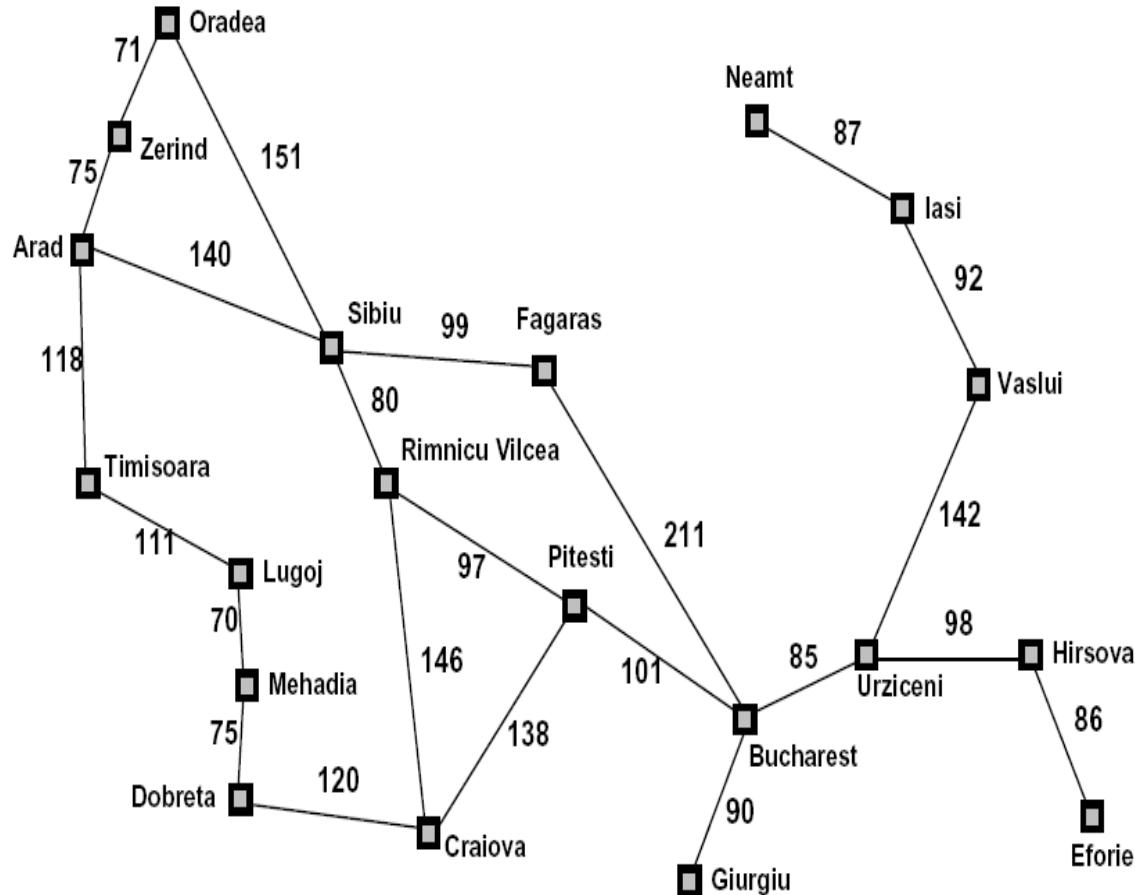
O euristică este:

- o funcție  $h$  care *estimează* cât de aproape suntem de o stare-scop;
- $h(n) \geq 0$ , pentru orice stare  $n$ ;
- $h(\text{stare-scop}) = 0$ ;
- specifică pentru fiecare problemă;
- exemple: distanța Manhattan, distanța Euclideană pentru găsirea celui mai scurt drum;



# Exemplu de funcție euristică

$h(x)$  = distanța în linie dreaptă între orașul  $x$  și București



Straight-line distance  
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

$h(x)$

# Exemplu de funcție euristică

- Pentru problema 8-puzzle putem folosi câteva euristici:
  - $h_1$ : numărul de piese așezate greșit față de starea-scop
  - $h_2$ : suma distanțelor pieselor față de poziția lor în starea scop (suma distanțelor Manhattan)

2		3
1	8	4
7	6	5

Stare inițială

$$h_1 = 4$$

$$h_2 = 1 + 1 + 0 + 0 + 0 + 0 + 0 + 1 + 1 = \\ 4$$

1	2	3
8		4
7	6	5

Stare scop

# Exemplu de funcție heuristică

1	2	3
8		4
7	6	5

Stare scop

2		3
1	8	4
7	6	5

$$h_1 = 4$$
$$h_2 = 4$$

STÂNGA

	2	3
1	8	4
7	6	5

$$h_1 = 3$$
$$h_2 = 4$$

JOS

2		3
1		4
7	6	5

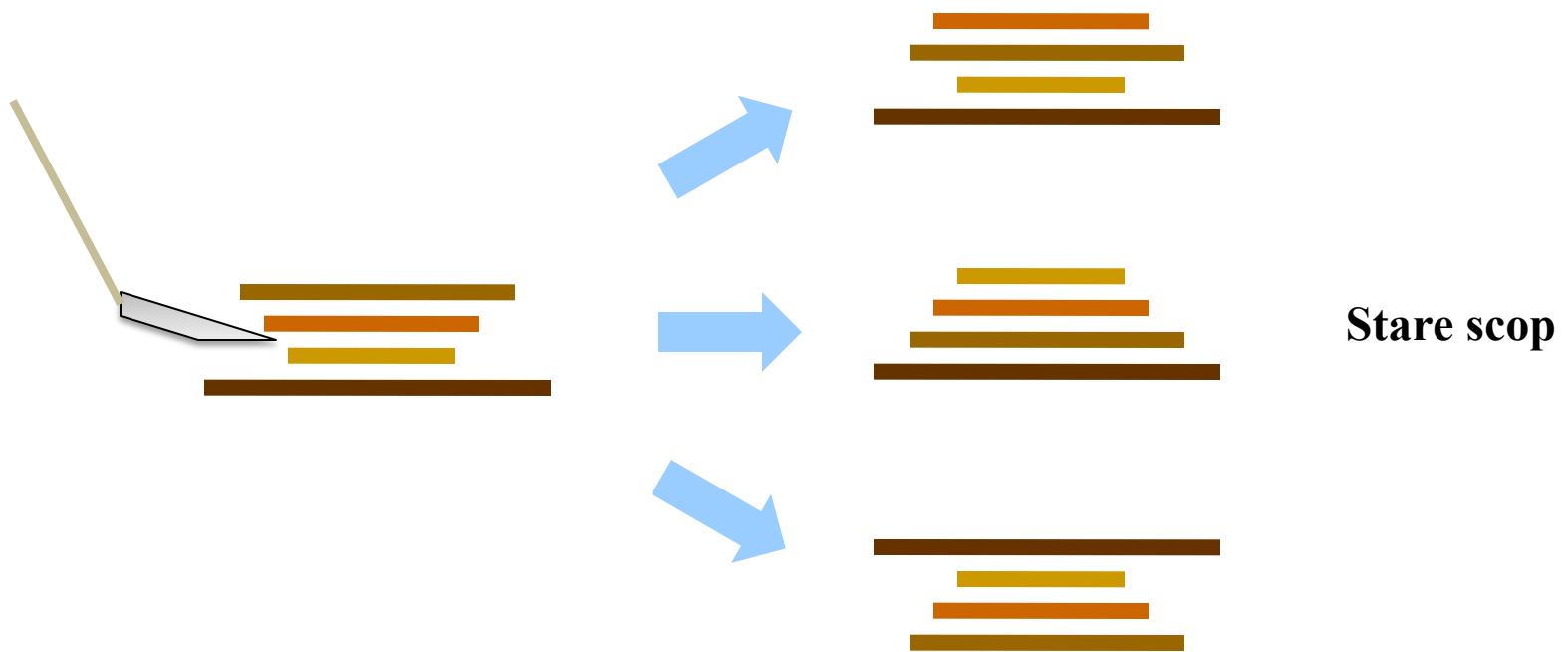
$$h_1 = 3$$
$$h_2 = 4$$

DREAPTA

2	3	
1	8	4
7	6	5

$$h_1 = 5$$
$$h_2 = 6$$

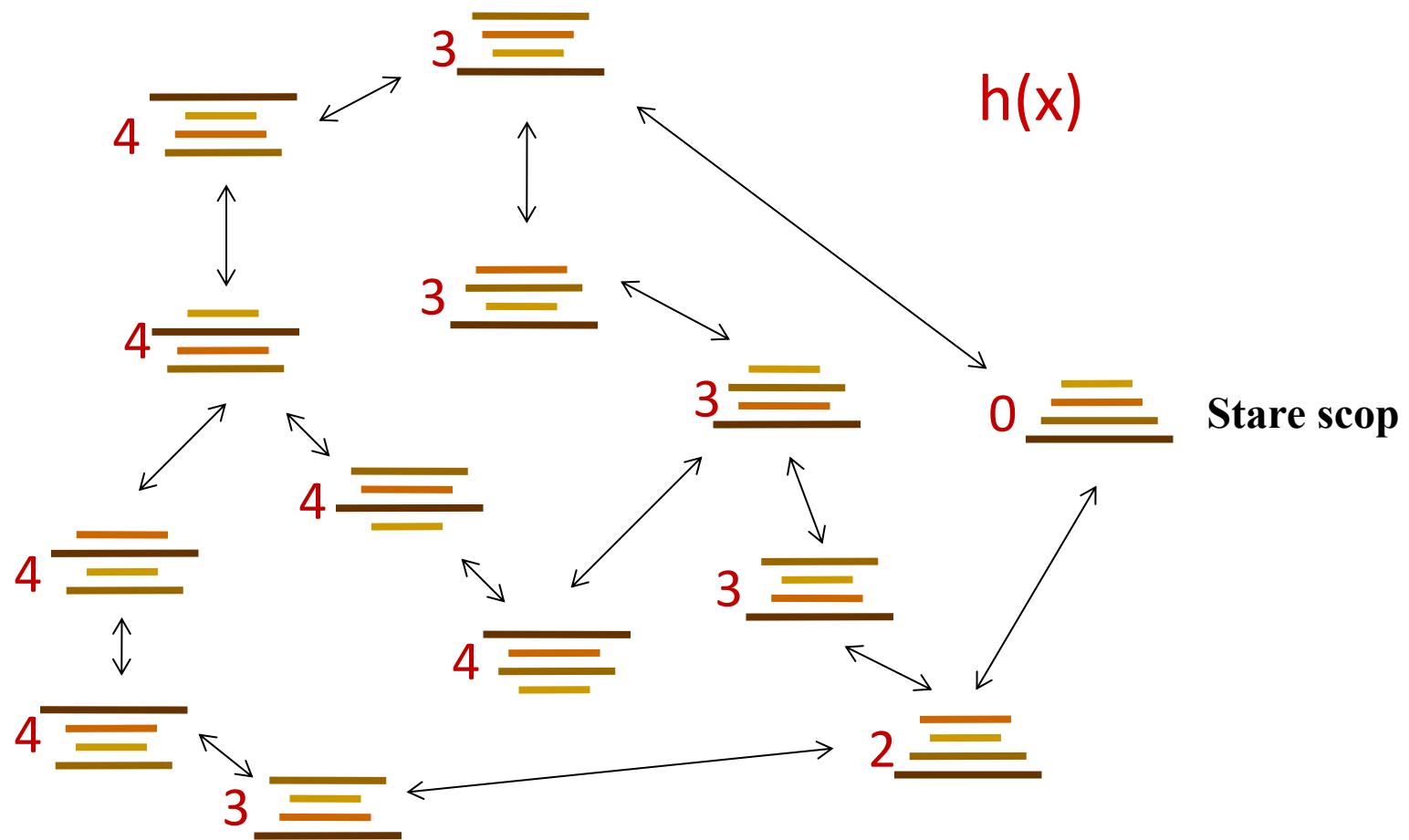
# Problema sortării clătitelor



Cost: numărul total de clătite întoarse

# Exemplu de funcție euristică

$h(x)$ : care este clătită de diametru cel mai mare din configurația  $x$  care nu este la locul ei?

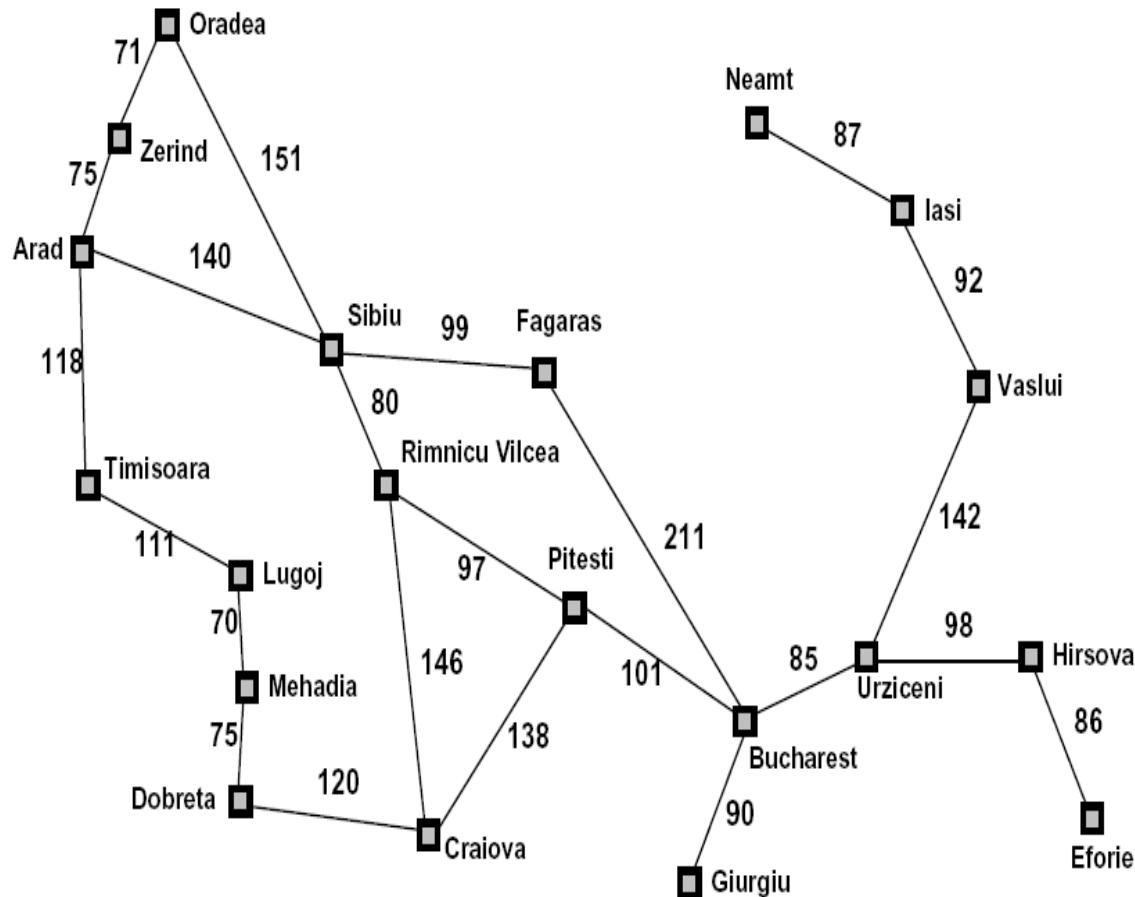


# Căutare Greedy (best-first)



# Exemplu de funcție euristică

$h(x)$  = distanța în linie dreaptă între orașul  $x$  și București



Straight-line distance  
to Bucharest

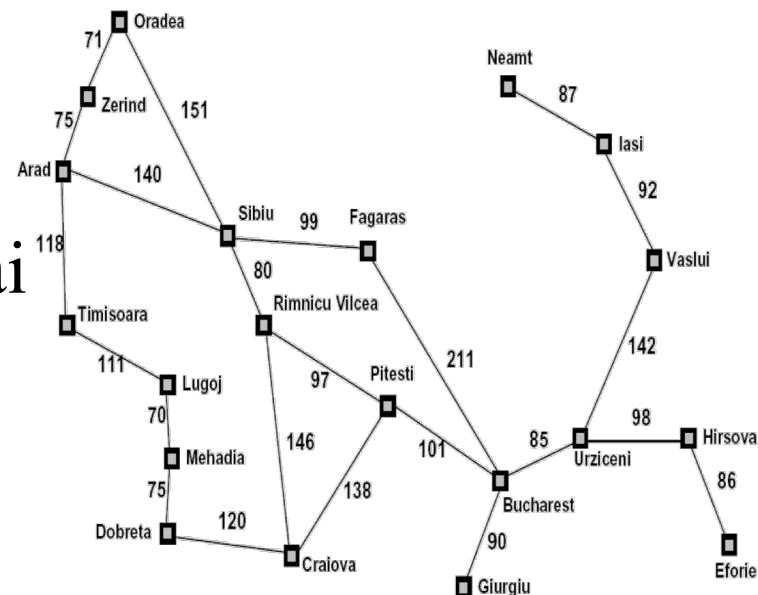
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

$h(x)$

# Căutare Greedy

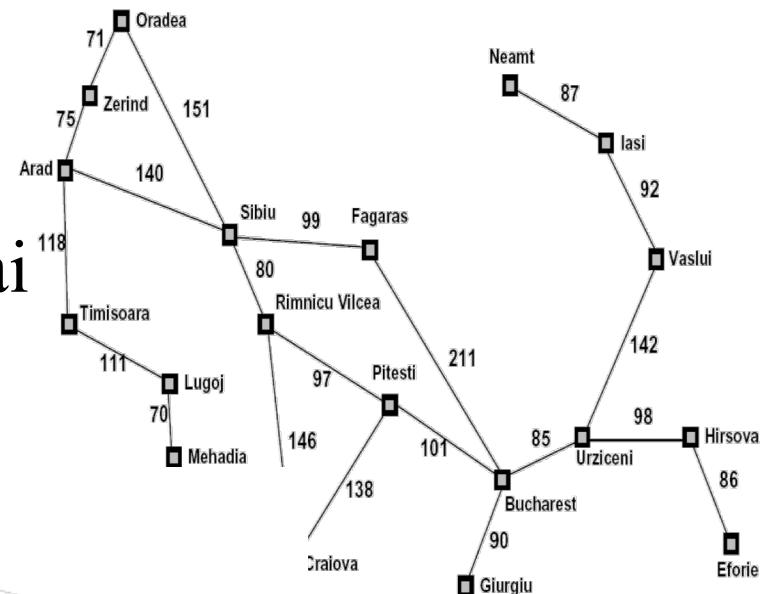
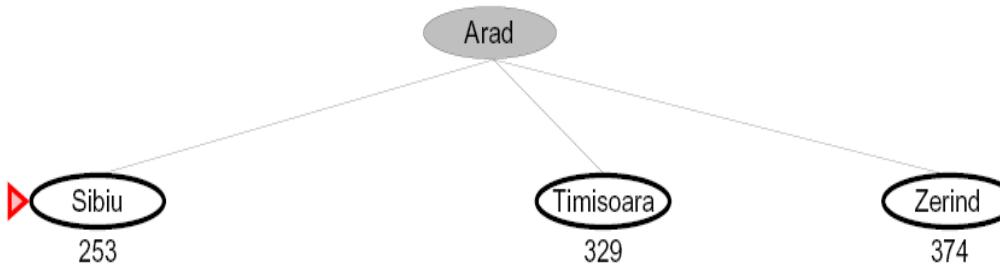
- Explorează nodul care pare cel mai aproape de un nod-scop (dpdv al euristicii, ea ghidează căutarea)

Arad  
366



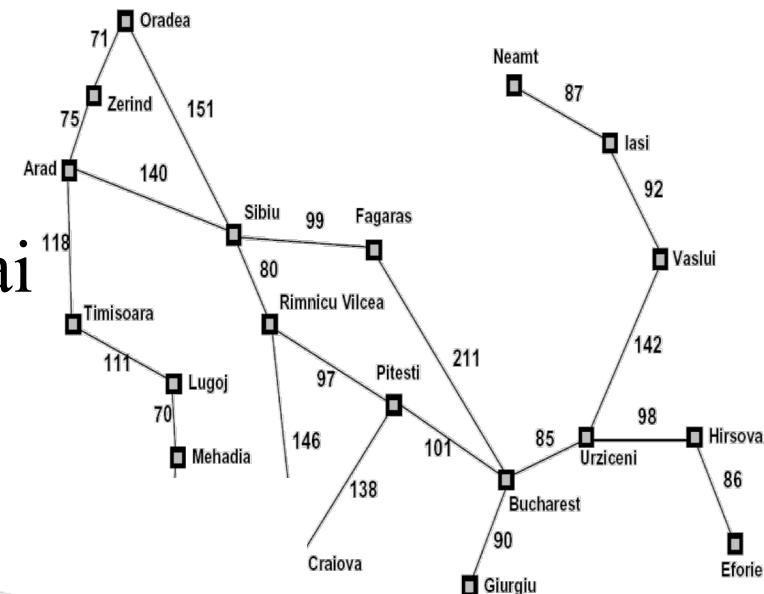
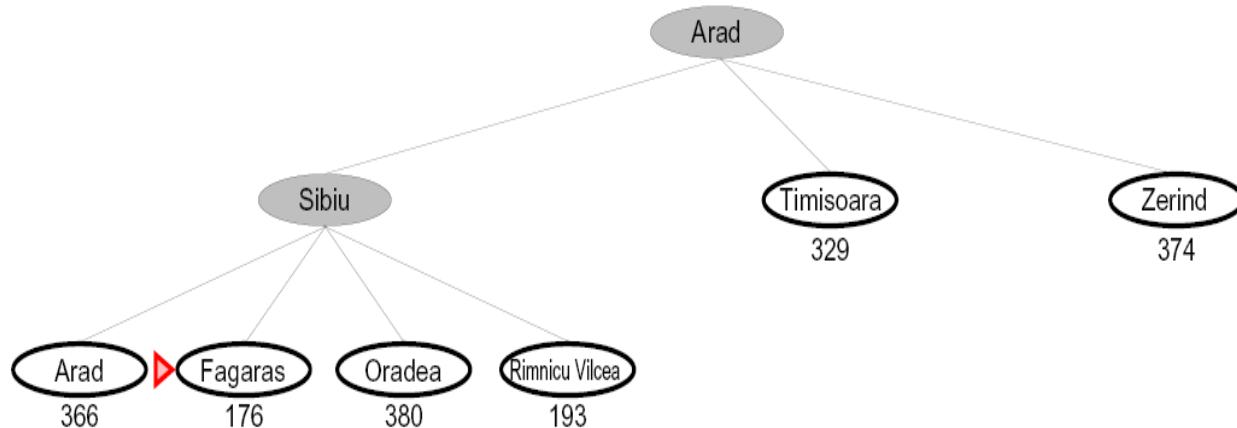
# Căutare Greedy

- Explorează nodul care pare cel mai aproape de un nod-scop (dpdv al euristicii)



# Căutare Greedy

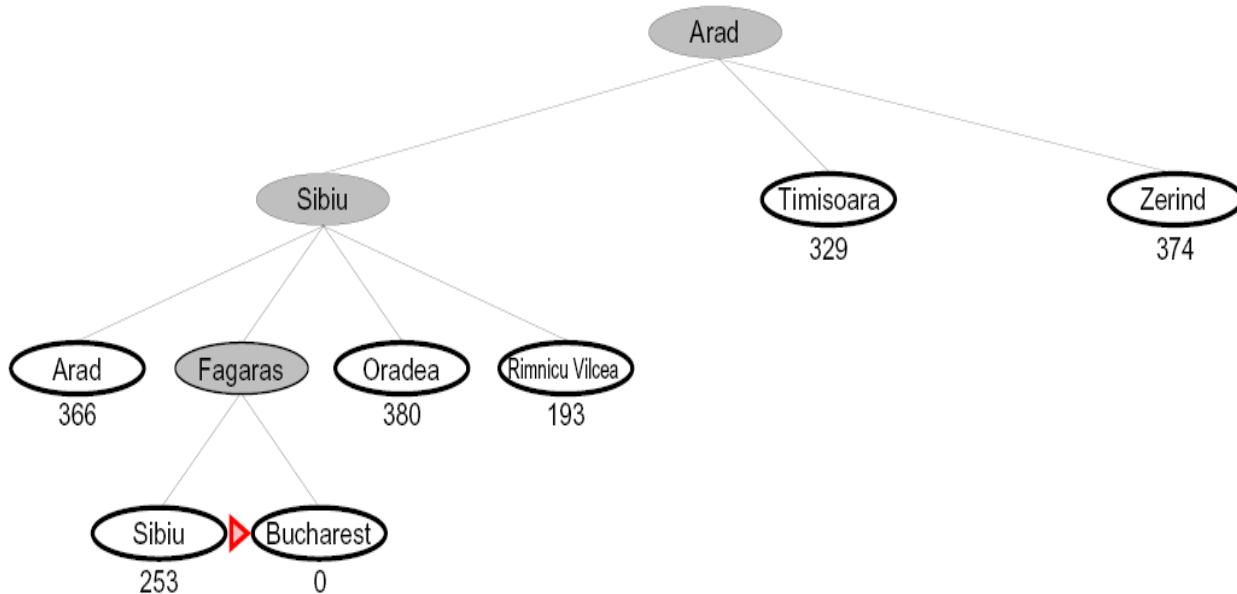
- Explorează nodul care pare cel mai aproape de un nod-scop (dpdv al euristicii)



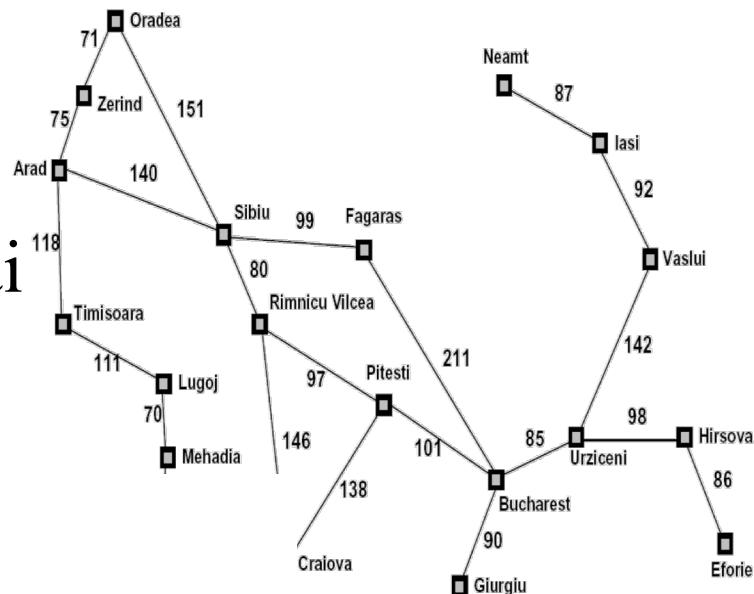
Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

# Căutare Greedy

- Explorează nodul care pare cel mai aproape de un nod-scop (dpdv al euristicii)

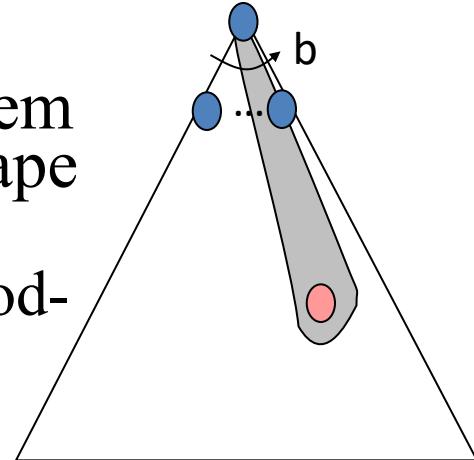


- Ce se poate întâmpla?
  - să obținem o soluție suboptimă: Arad – Sibiu – Făgăraș - București, cu un cost mai mare decât soluția optimă: Arad – Sibiu – Râmnicu-Vâlcea-Pitești-București

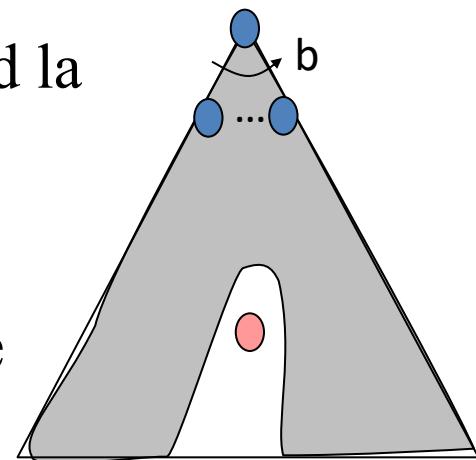


# Căutare Greedy

- Strategia: expandează un nod care credem (dpdv al euristicii) că este cel mai aproape de un nod-scop
  - euristică: distanță estimată față de un nod-scop pentru fiecare nod curent



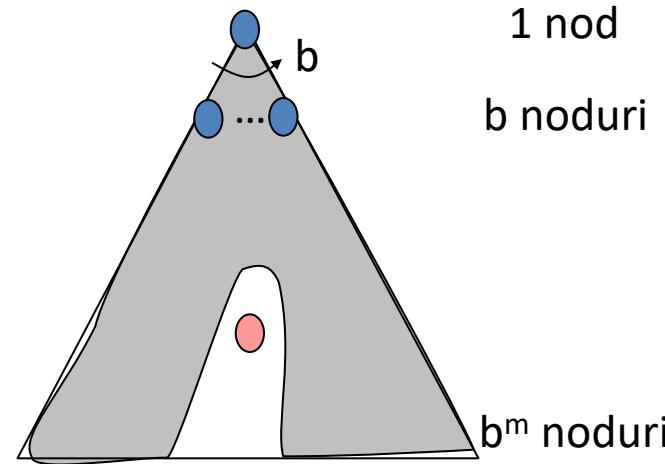
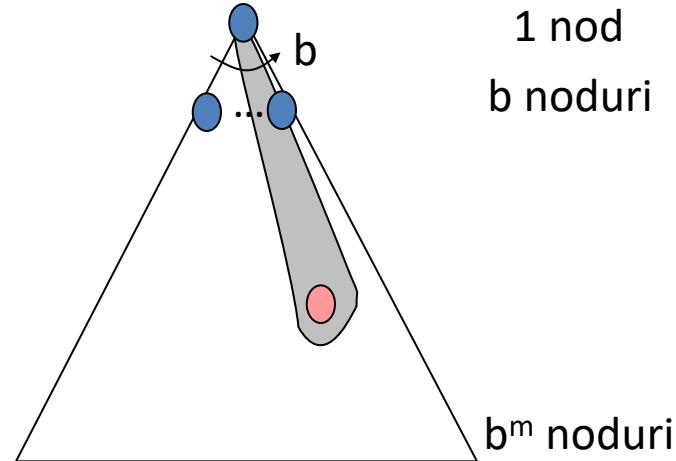
- Caz ușual:
  - strategia Greedy (best-first) ajunge rapid la o soluție, dar este suboptimă



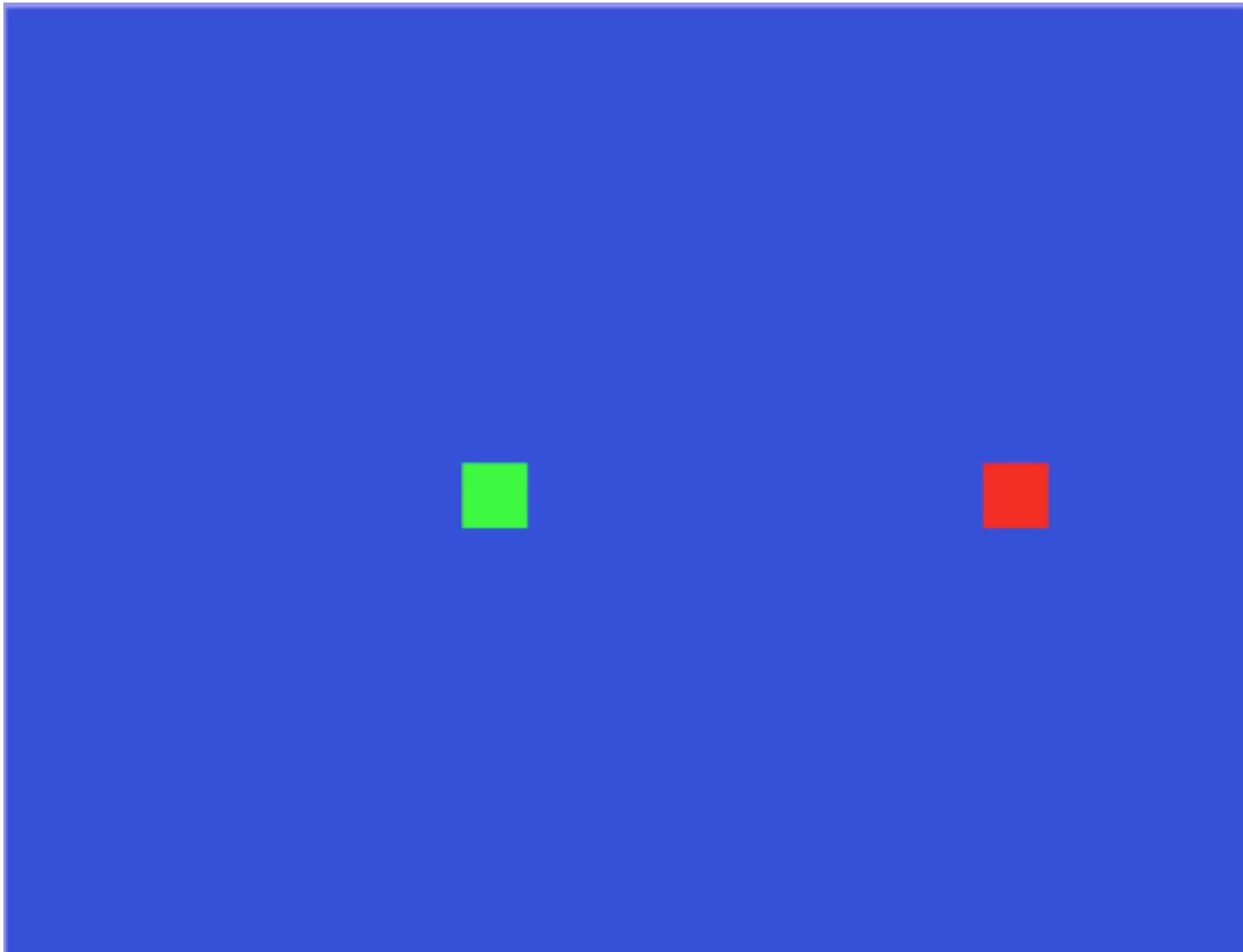
- Caz defavorabil: căutare în adâncime prost ghidată

# Proprietățile căutării Greedy

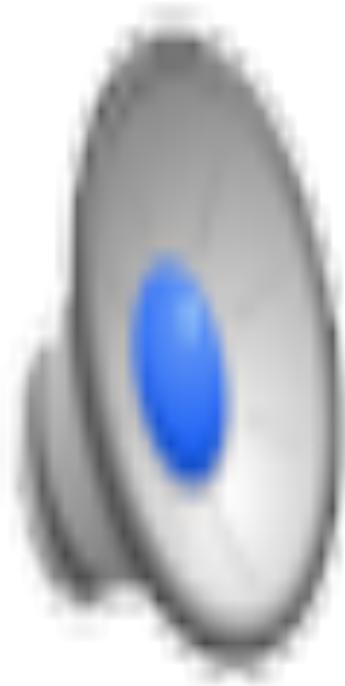
- Ce fel de noduri expandează Greedy?
  - poate procesa toate nodurile
  - complexitate timp  $O(b^m)$
- Cât de mult spațiu necesită Greedy?
  - este spațiul pentru memorarea frontierei
  - complexitate  $O(b^m)$
- Completitudine?
  - dacă o soluție există atunci DA! (dacă ține minte pe unde a fost și nu intră în circuite)
- Optimalitate?
  - NU



# Căutare Greedy - demo



# Căutare Greedy - demo

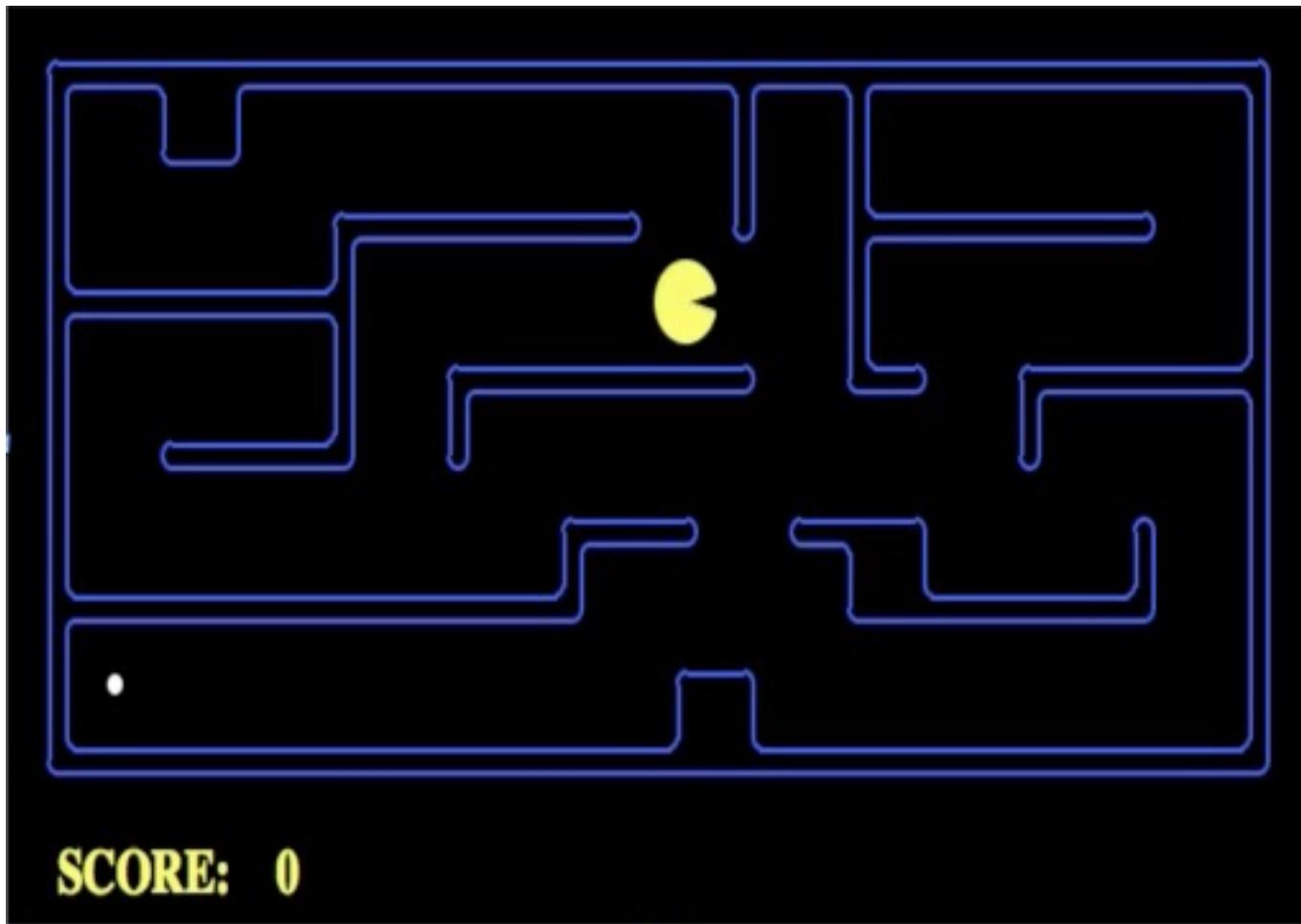


# Căutare Greedy - demo



Soluția Greedy  
Este optimă

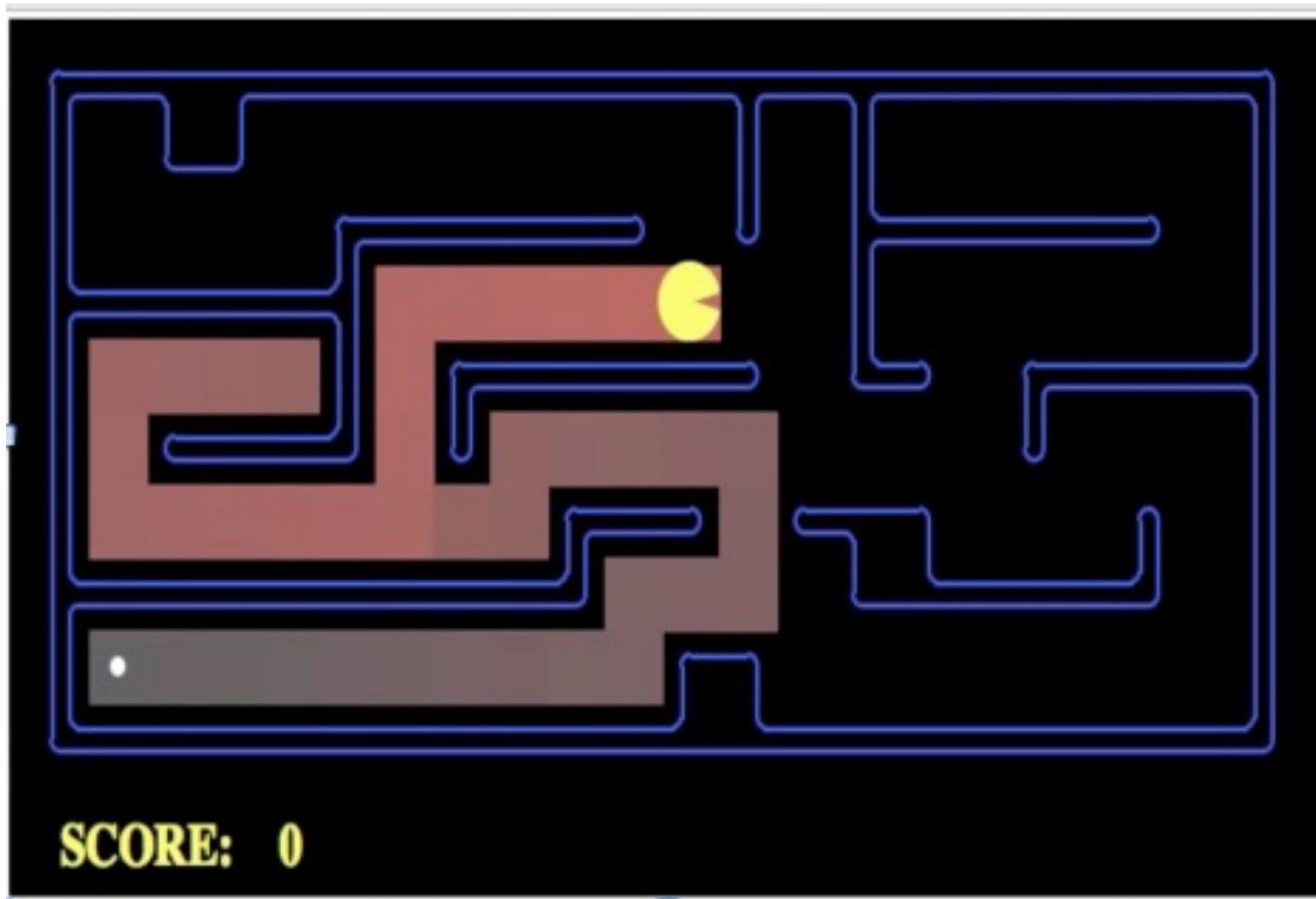
# Căutare Greedy - demo



# Căutare Greedy - demo



# Căutare Greedy - demo



**Soluția Greedy  
Nu este optimă**

# Algoritmul de căutare A\*



# Algoritmul de căutare A\*

1

# Algoritmul de căutare A\*



UCS



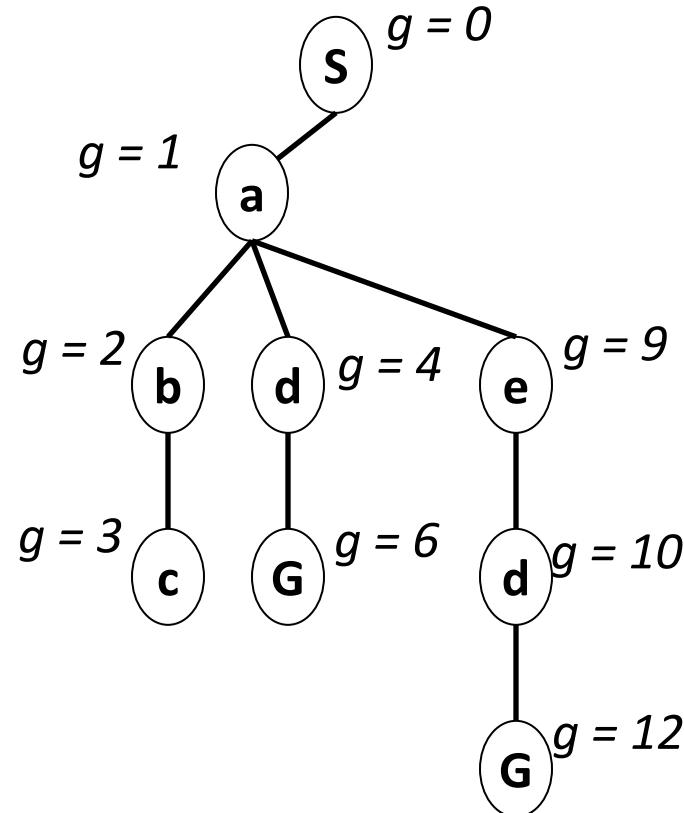
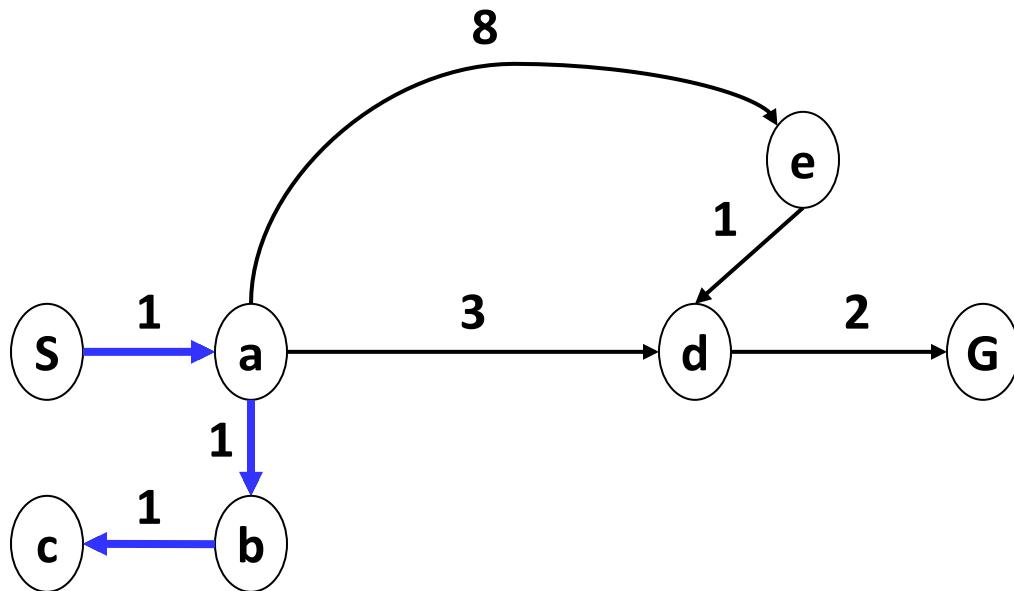
Greedy



A\*

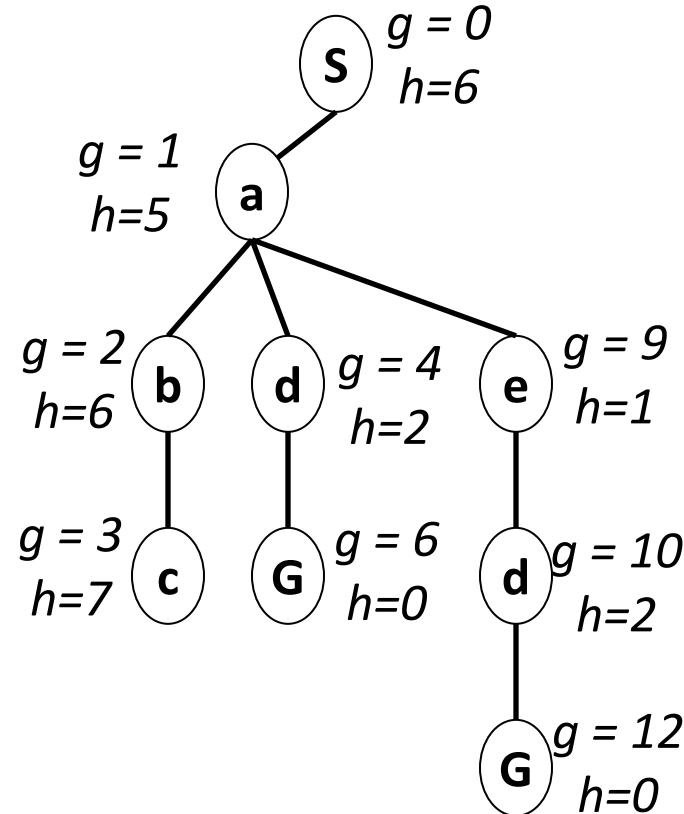
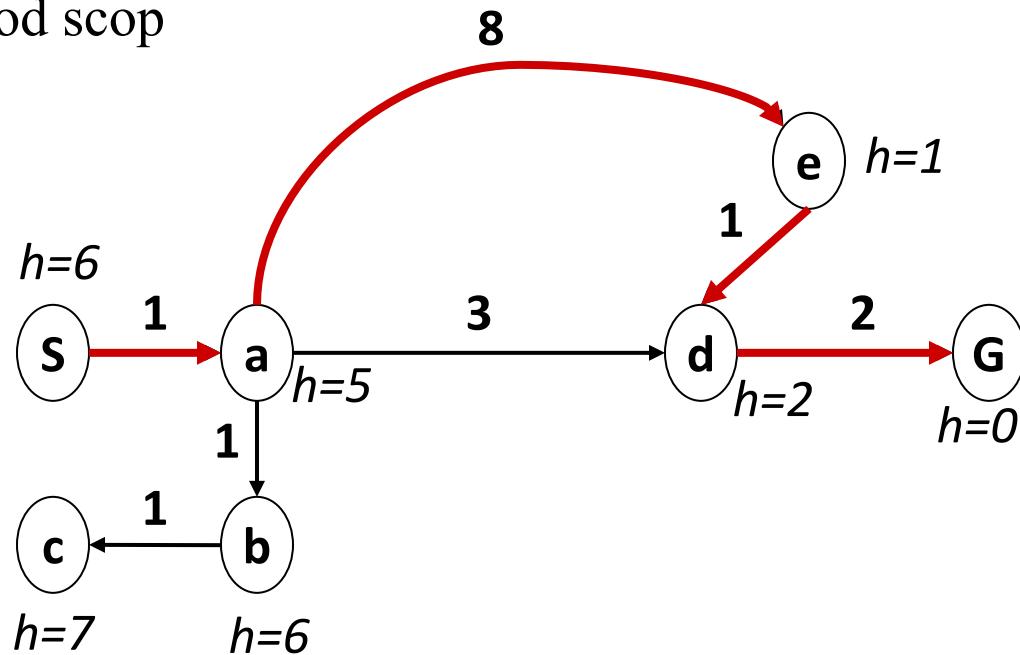
# Combinăm UCS + Greedy

- Căutarea uniformă bazată pe cost preferă să exploreze soluții parțiale (drumuri) de cost curent minim (se bazează pe un calcul real, nu există estimări)  
 $g(n)$  = **backward cost** = cât costă drumul de la nodul initial la nodul curent
- Găsește în final soluția de cost minim 6:  $s \rightarrow a \rightarrow d \rightarrow G$



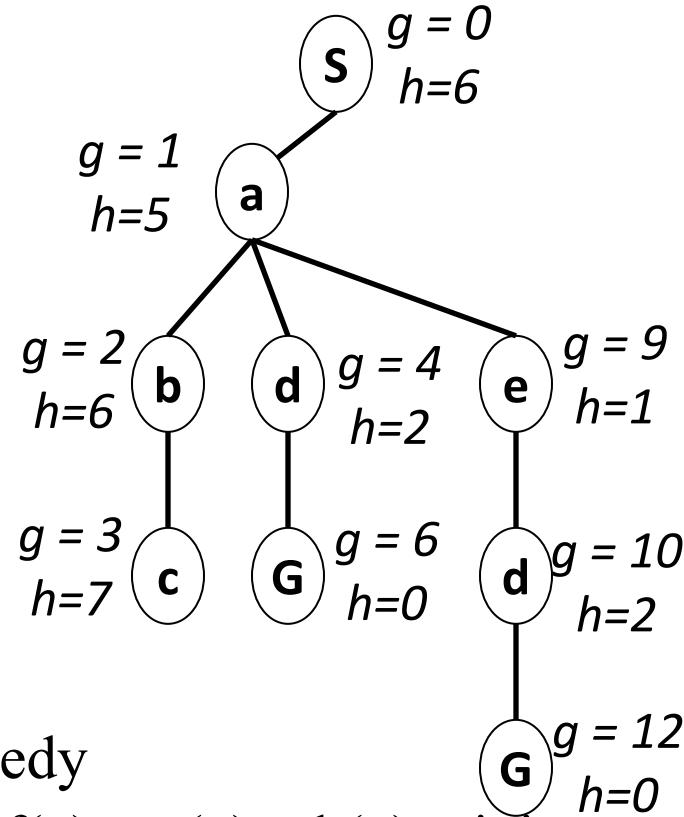
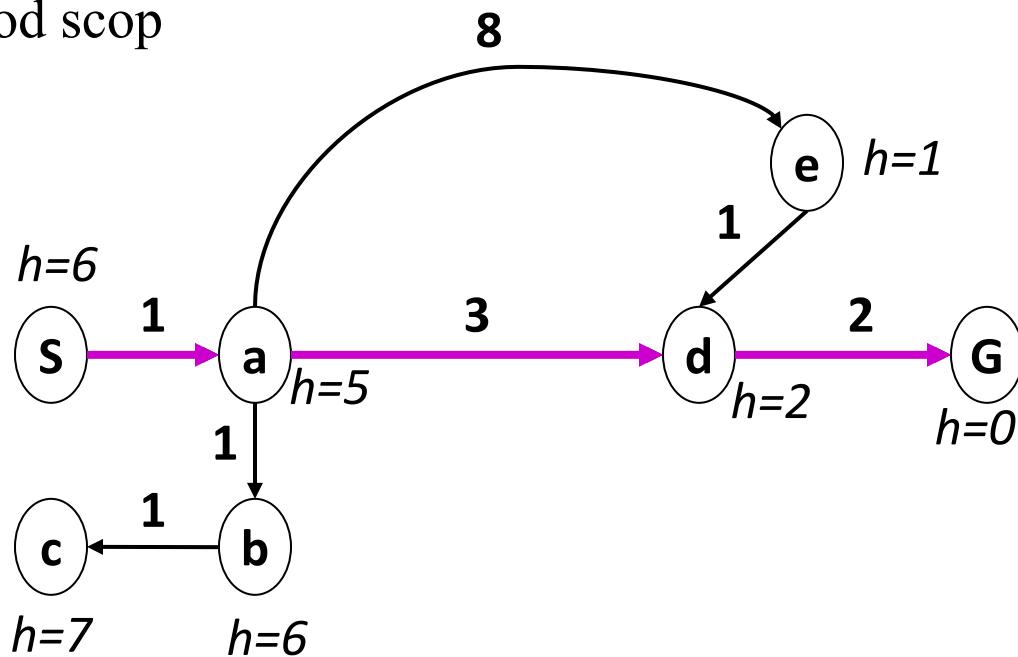
# Combinăm UCS + Greedy

- **Căutarea uniformă bazată pe cost** preferă să exploreze soluții parțiale (drumuri) de cost curent minim (se bazează pe un calcul real, nu există estimări)  
 $g(n)$  = **backward cost** = cât costă drumul de la nodul inițial la nodul curent
- **Căutarea Greedy** preferă să exploreze soluții parțiale (drumuri) pentru care nodul curent este cel mai apropiat de un nod scop (se bazează pe o estimare)  
 $h(n)$  = **forward cost** = cât estimez că va costa drumul de la nodul curent la un nod scop



# Combinăm UCS + Greedy

- **Căutarea uniformă bazată pe cost** preferă să exploreze soluții parțiale (drumuri) de cost curent minim (se bazează pe un calcul real, nu există estimări)  
 $g(n)$  = **backward cost** = cât costă drumul de la nodul inițial la nodul curent
- **Căutarea Greedy** preferă să exploreze soluții parțiale (drumuri) pentru care nodul curent este cel mai apropiat de un nod scop (se bazează pe o estimare)  
 $h(n)$  = **forward cost** = cât estimez că va costa drumul de la nodul curent la un nod scop

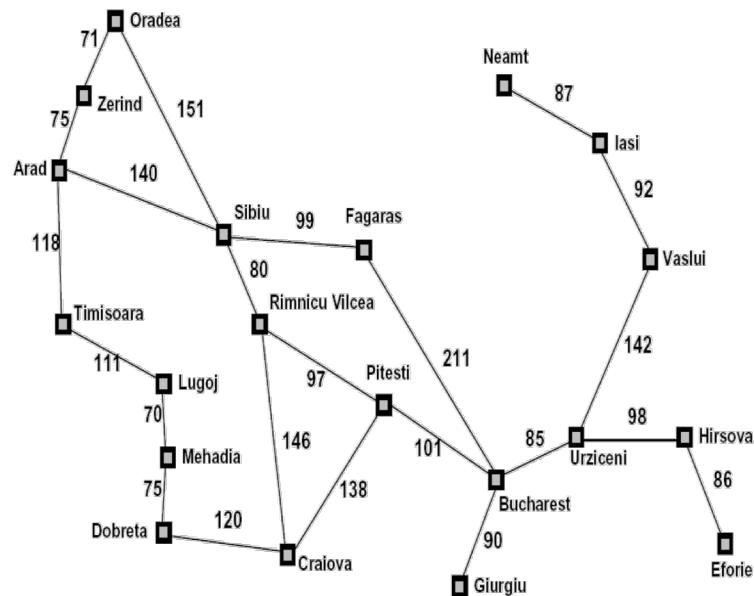


- **Algoritmul de căutare A\*** combină UCS + Greedy
  - Preferă să exploreze soluțiile parțiale de cost  $f(n) = g(n) + h(n)$  minim

# Exemplu de căutare A\*

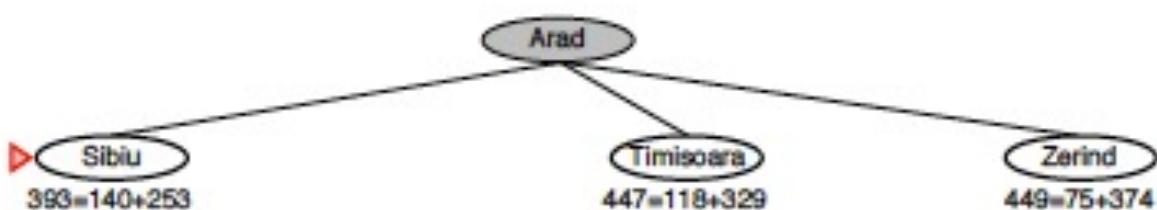
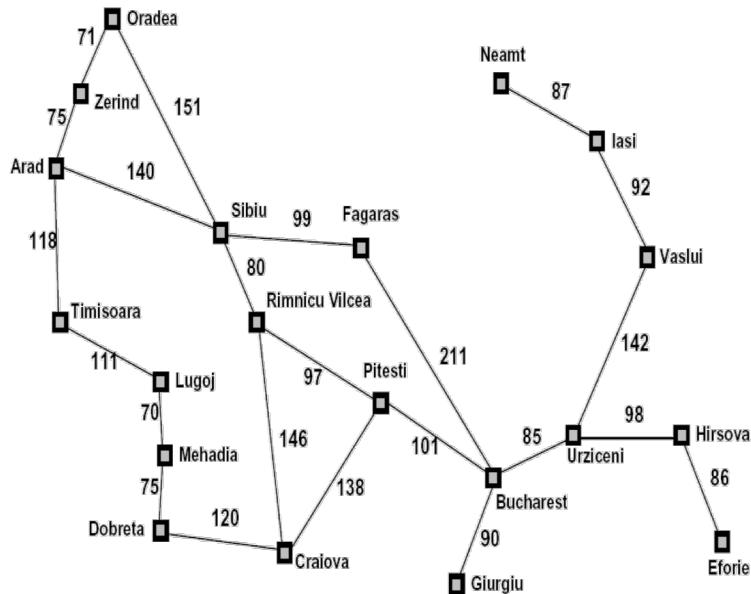
- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$

trecut      viitor  
(exact)      (estimat)



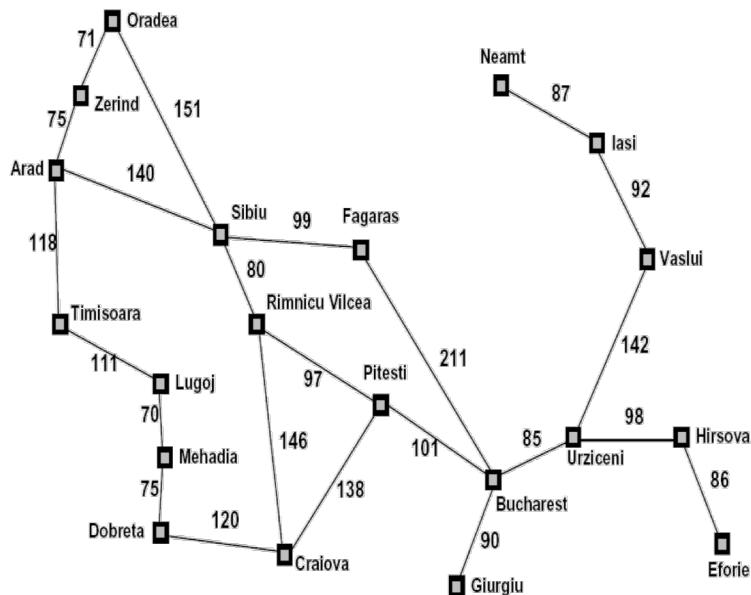
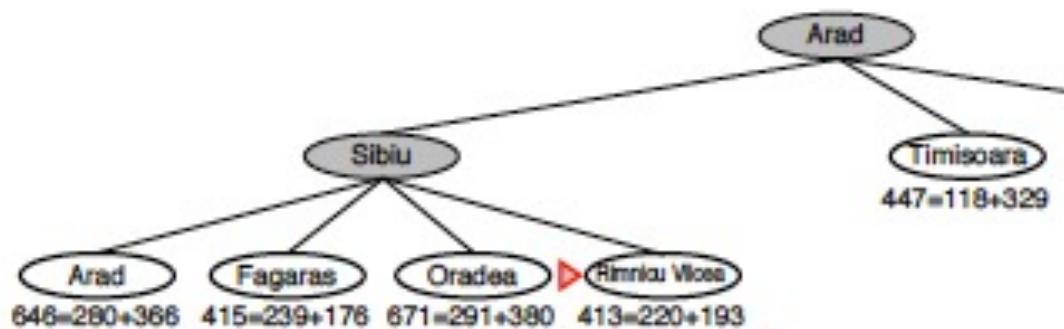
# Exemplu de căutare A\*

- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$



# Exemplu de căutare A\*

- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$

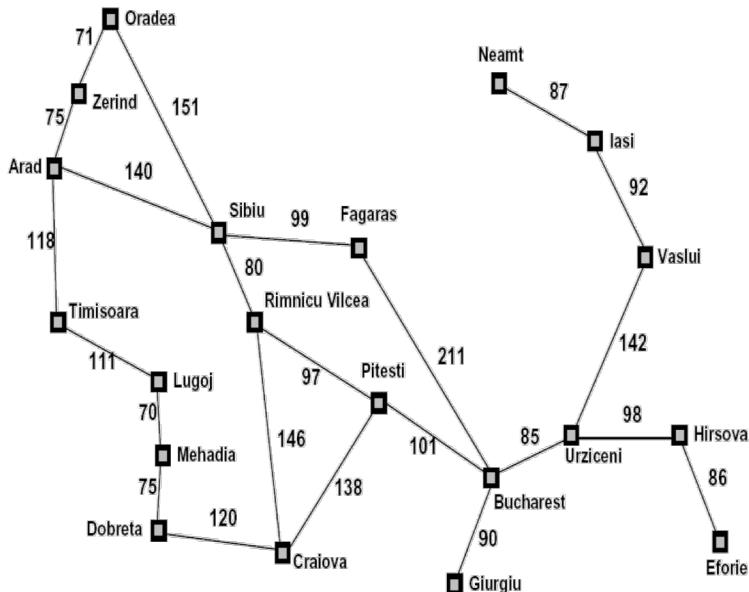


Straight-line distance  
to Bucharest

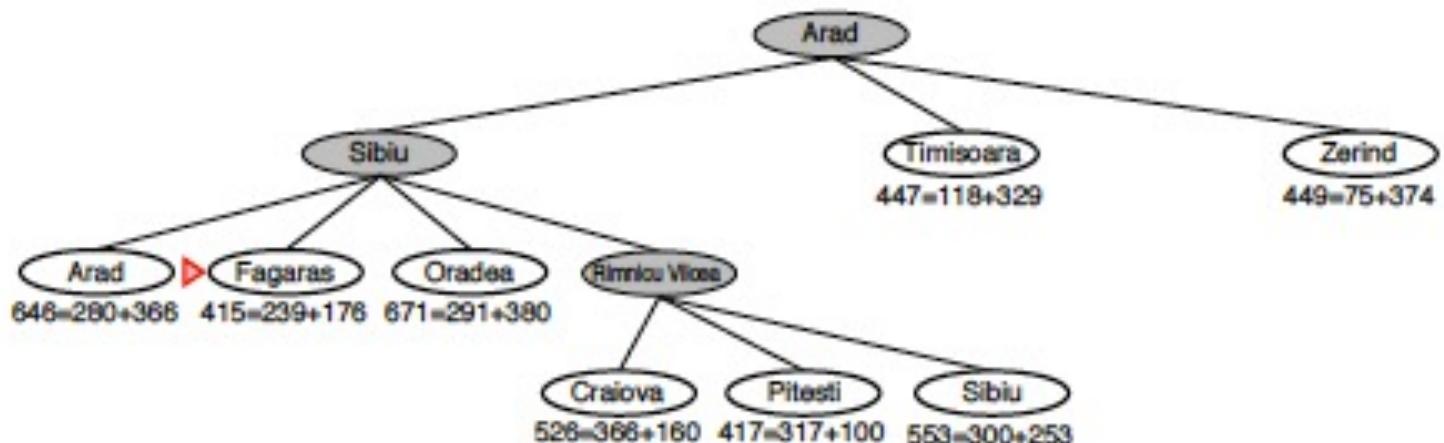
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

# Exemplu de căutare A\*

- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$



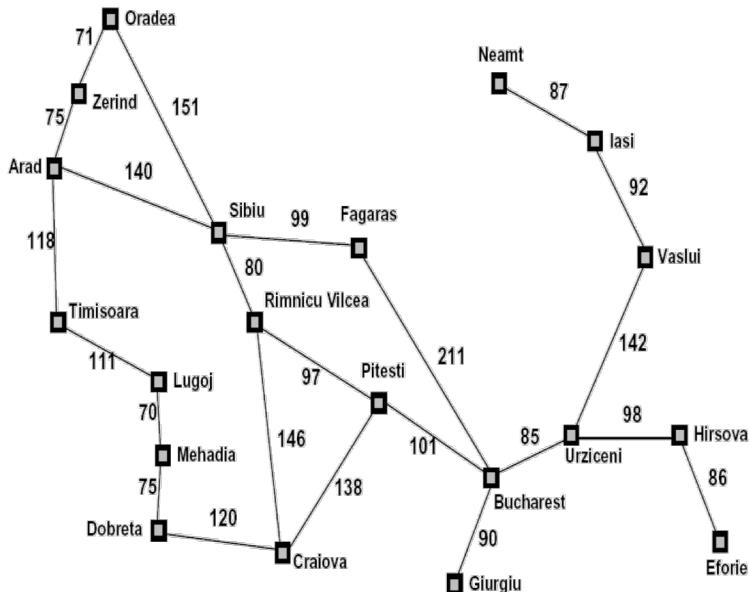
Straight-line distance  
to Bucharest



Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

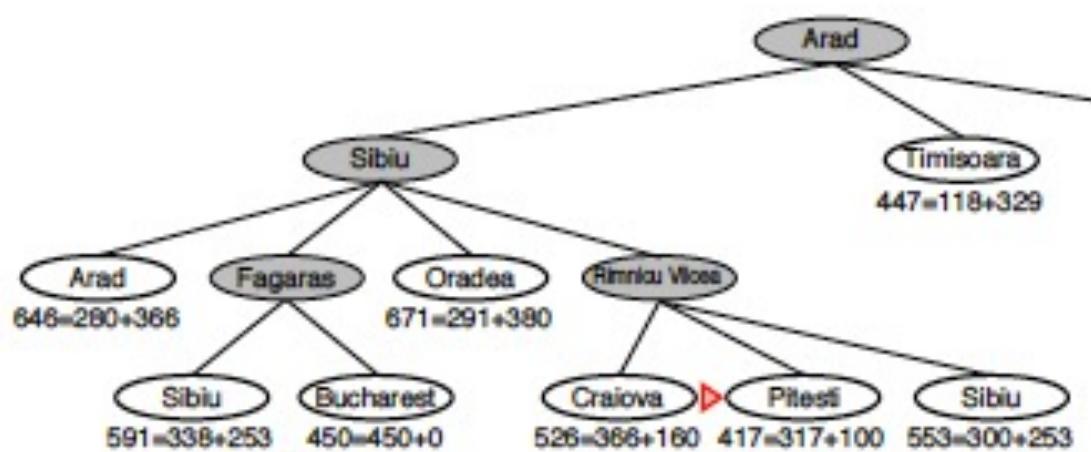
# Exemplu de căutare A\*

- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$



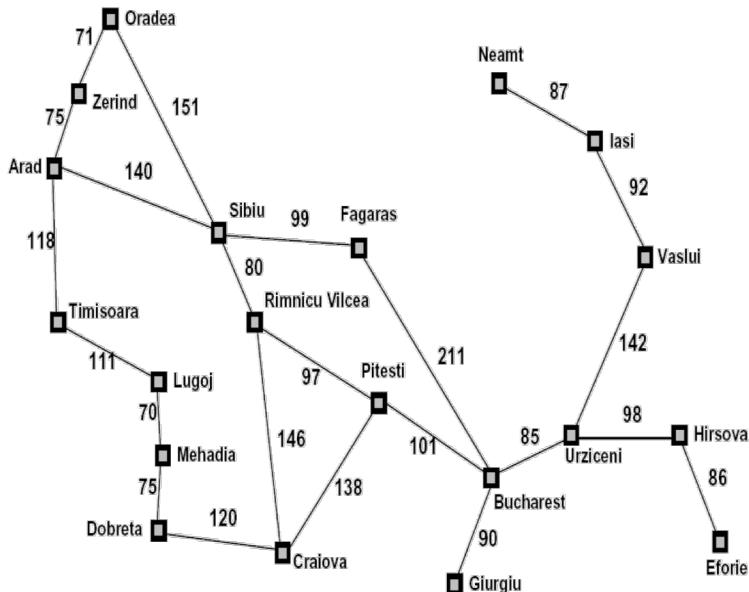
Straight-line distance  
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

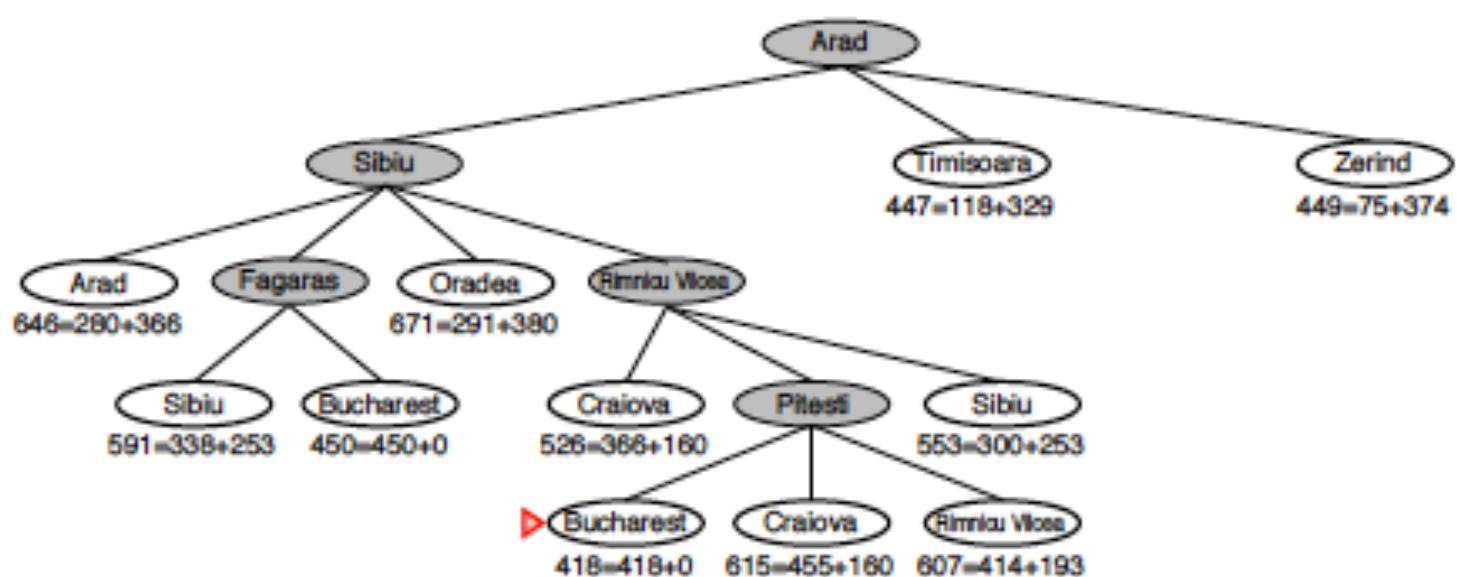


# Exemplu de căutare A\*

- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$



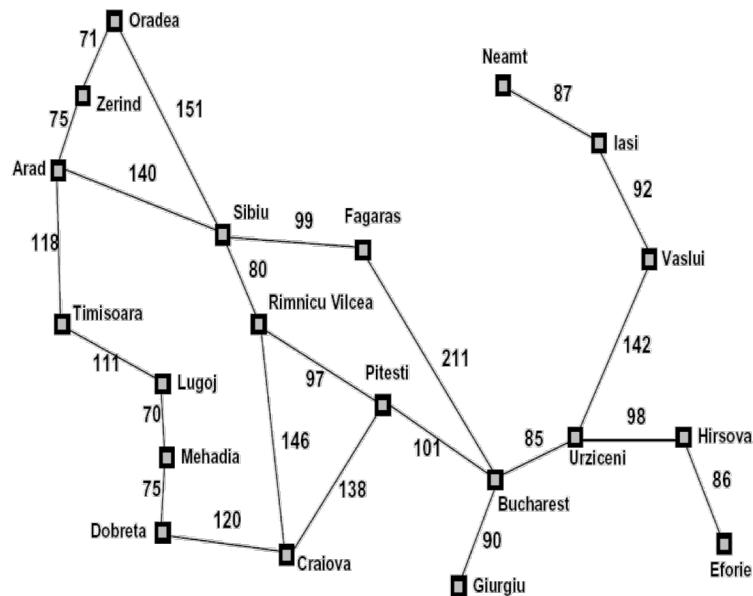
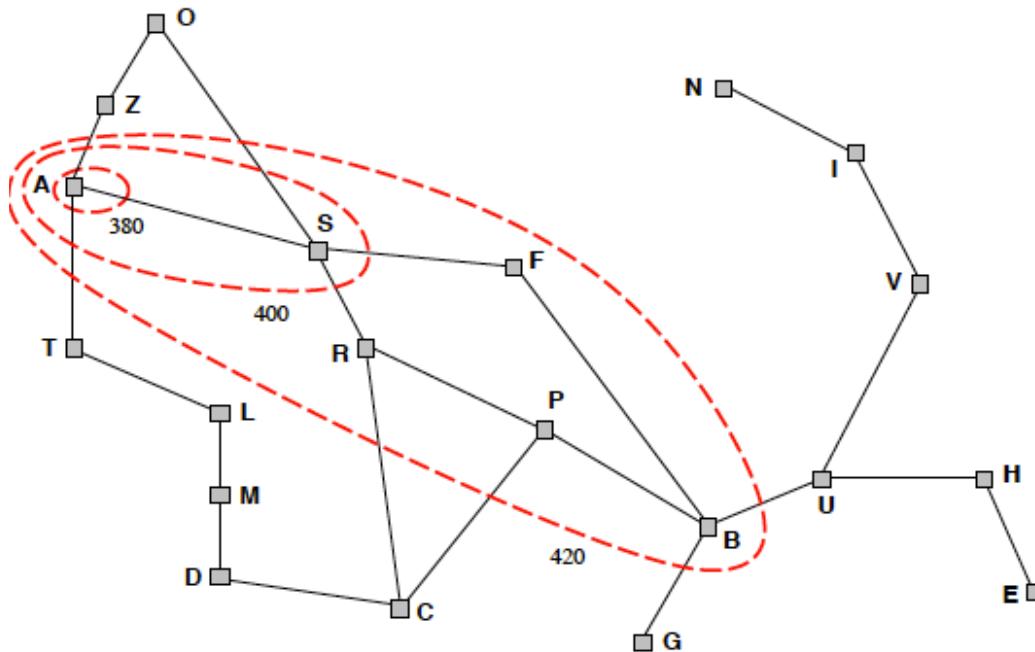
Straight-line distance  
to Bucharest



Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

# Exemplu de căutare A\*

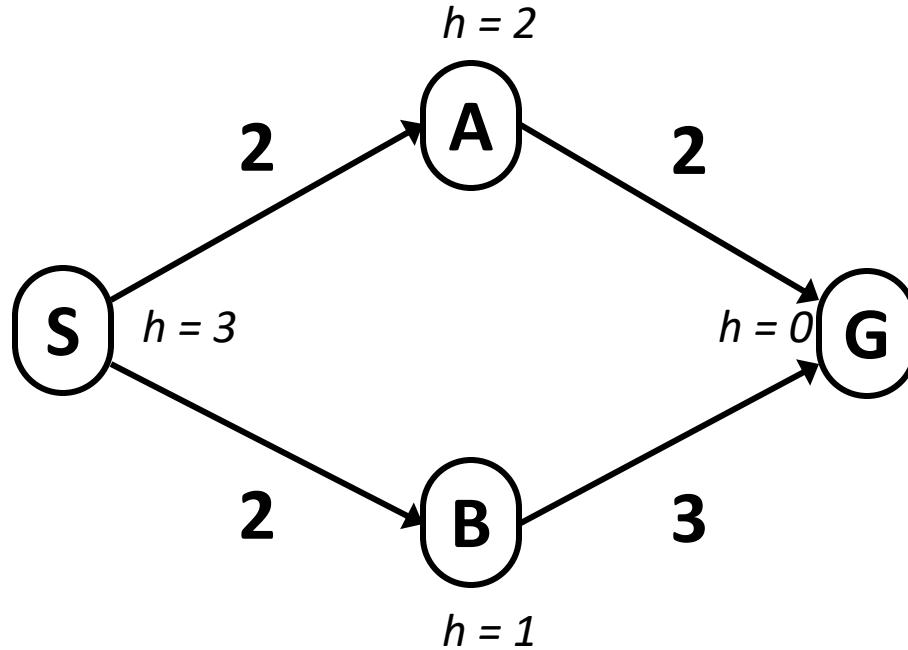
- Explorează nodul care pare cel mai promițător din perspectiva  $f(n) = g(n) + h(n)$
- Rezultat teoretic: A\* explorează pe rând noduri în ordinea crescătoare a funcției f asociate



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

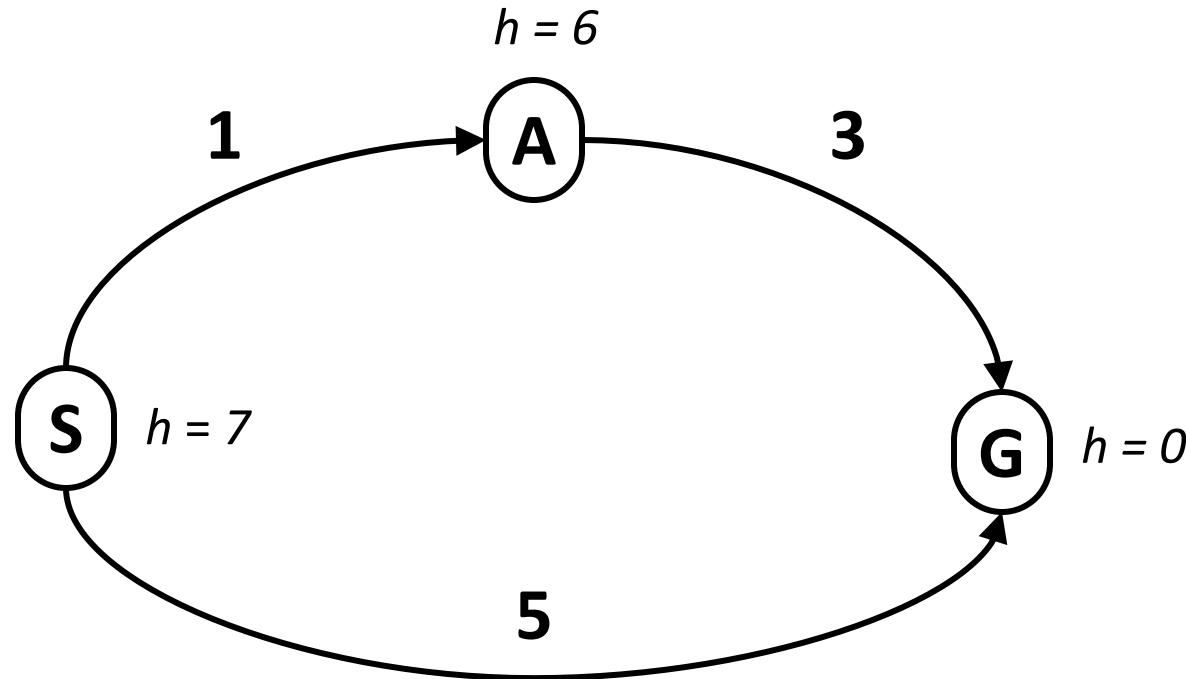
# Când trebuie să ne oprim cu algoritmul A\*?

- Ne oprim când ajungem la o stare –scop?



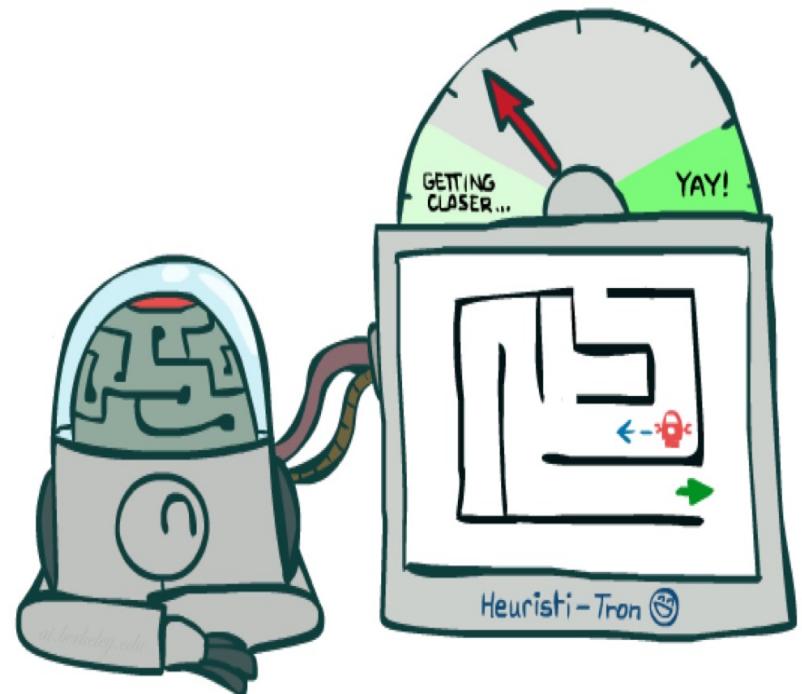
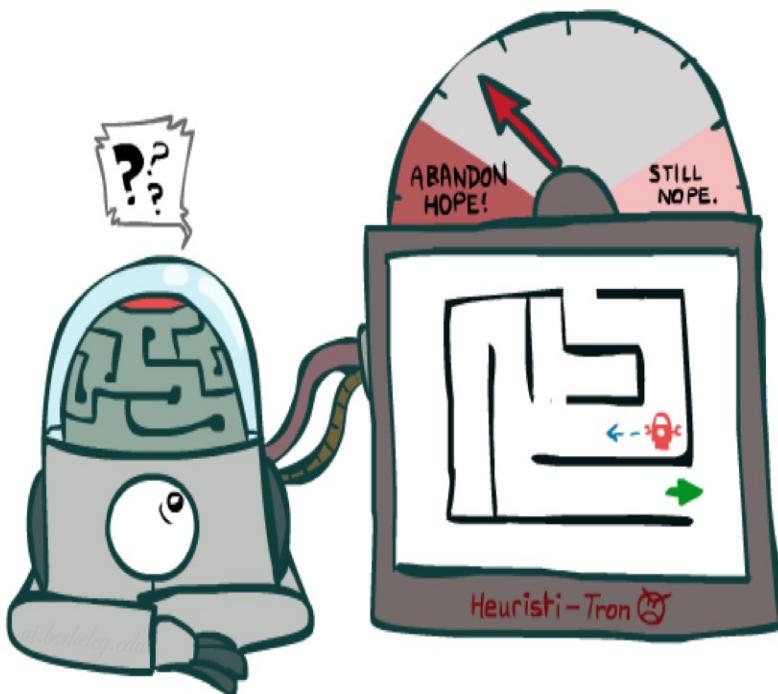
- NU: ne oprim numai după ce explorăm toate nodurile care au şansa să conducă la soluţie optimă

# Este algoritmul A\* optim?



- Avem nevoie de estimări (date de euristici) care să fie mai mici decât costurile reale (euristici optimiste)
- Optimalitate condiționată de euristici cu o asemenea caracteristică: A\* este optim dacă avem euristici optimiste

# Admisibilitatea euristicilor



Euristici inadmisibile (pesimiste): supraestimează costul unui drum, nu conduc la soluții optime, stopează explorarea unor drumuri de cost minim

Euristici admisibile (optimiste): subestimează costul unui drum, încetinesc explorarea unor drumuri de cost mare, nu supraestimează costurile drumurilor

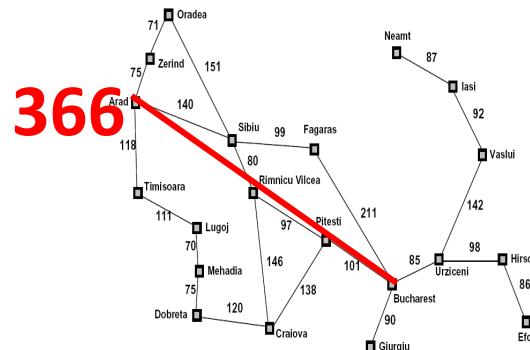
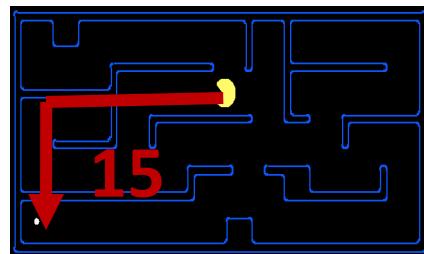
# Euristici admisibile

- O euristică  $h$  este *admisibilă* (optimistă) dacă:

$$0 \leq h(n) \leq h^*(n)$$

unde  $h^*(n)$  este costul real către cel mai apropiat nod scop

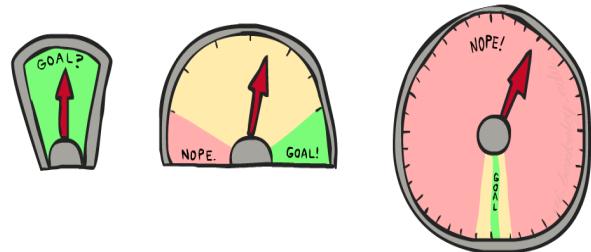
- Exemple:



- Găsirea de euristici admisibile este partea cea mai grea în aplicarea algoritmului A\*.

# Euristici pentru A\*

- de ce să nu folosim ca euristică *costul real* până la un nod scop?
  - este o asemenea euristică admisibilă?
  - explorăm mai multe sau mai puține noduri?
  - există vreun dezavantaj?
- compromis între calitatea estimării dată de euristică și numărul de noduri exploreate:
  - pe măsură ce o euristică se apropii mai bine de costul real, A\* va explora mai puține noduri, însă ia mai mult timp să calculeze valoarea euristică la nodul curent



# Euristici dominante și banale

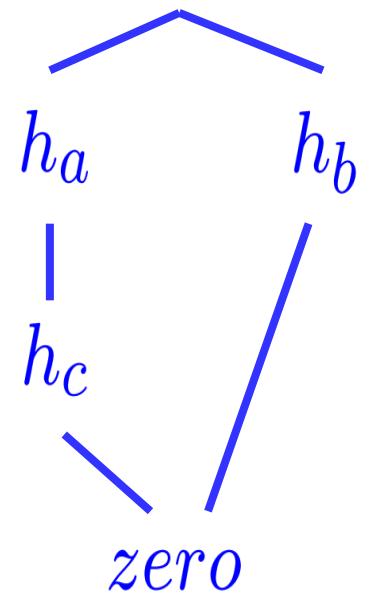
- dominanță:  $h_a \geq h_c$  dacă

$$\forall n : h_a(n) \geq h_c(n)$$

*exact*

|

*max( $h_a, h_b$ )*



- euristicile formează o semi-latice:

- maximum a două euristică admisibile este o euristică admisibilă

$$h(n) = \max(h_a(n), h_b(n))$$

- euristică banală:

- euristică zero (la ce conduce în A\*?)
  - euristică exactă

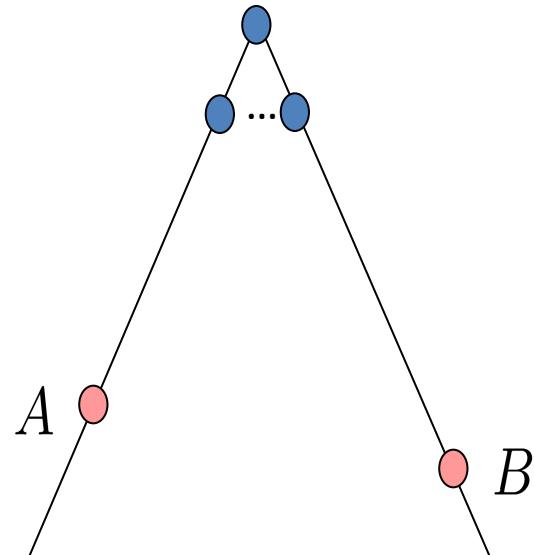
# Optimalitatea algoritmului A\*

Presupunem:

- A este un nod-scop optim global (nu există alt nod-scop cu cost mai mic)
- B este un nod-scop suboptimal (are cost mai mare decât soluția optimă)
- $h$  este euristică admisibilă

Afirmație:

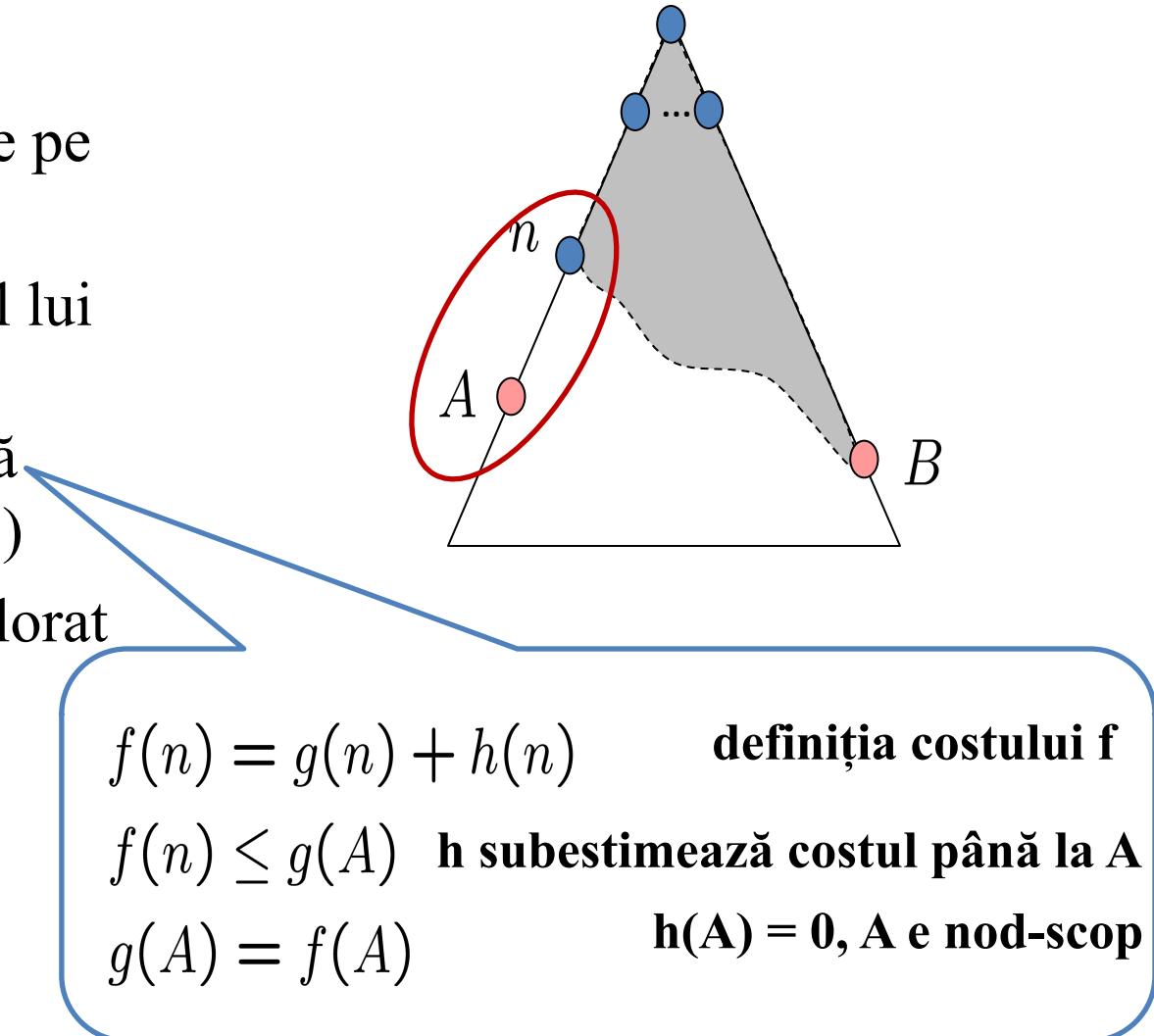
- A va fi explorat de algoritmul A\* înaintea lui B



# Optimalitatea algoritmului A\*

Demonstrație:

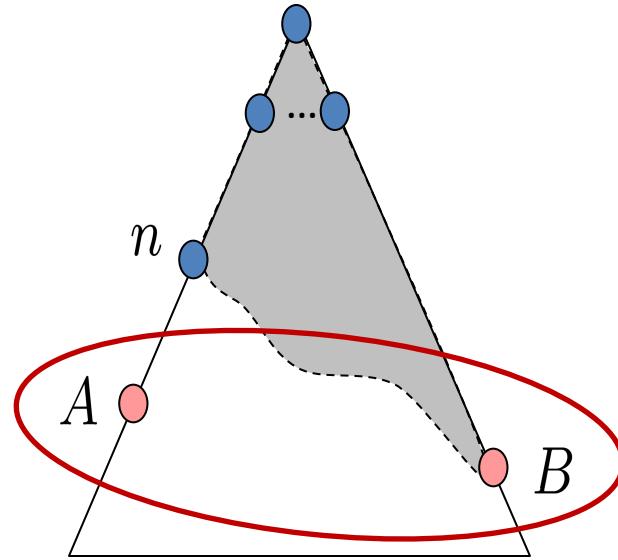
- presupunem că B este pe frontieră
- un nod ascendent  $n$  al lui A (strămoș) este de asemenea pe frontieră (poate chiar nodul A!)
- afirmație:  $n$  va fi explorat înaintea nodului B
  1.  $f(n) \leq f(A)$



# Optimalitatea algoritmului A\*

Demonstrație:

- presupunem că B este pe frontieră
- un nod ascendent  $n$  al lui A (strămoș) este de asemenea pe frontieră (poate chiar nodul A!)
- afirmație:  $n$  va fi explorat înaintea nodului B
  1.  $f(n) \leq f(A)$
  2.  $f(A) < f(B)$



$$g(A) < g(B)$$

$$f(A) < f(B)$$

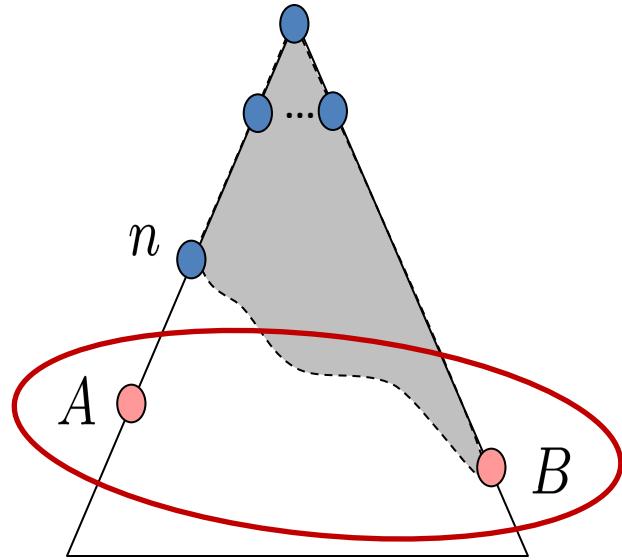
**B este nod suboptimal**

**$h(A) = 0, h(B) = 0$**   
admisibilitatea lui  $h$

# Optimalitatea algoritmului A\*

Demonstrație:

- presupunem că B este pe frontieră
- un nod ascendent  $n$  al lui A (strămoș) este de asemenea pe frontieră (poate chiar nodul A!)
- afirmație:  $n$  va fi explorat înaintea nodului B
  1.  $f(n) \leq f(A)$
  2.  $f(A) < f(B)$
  3.  $n$  este explorat înaintea lui B

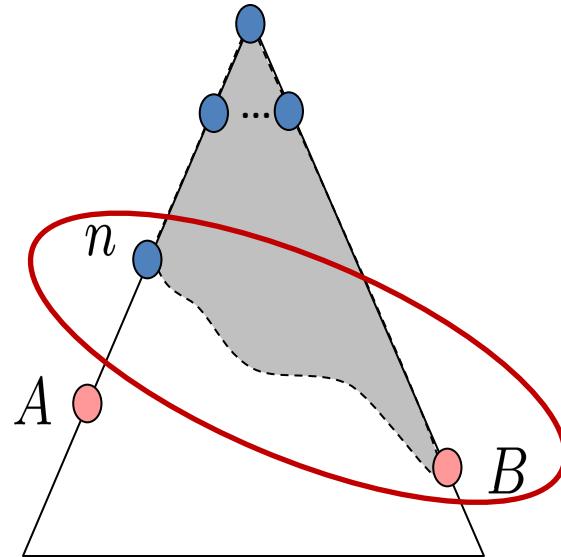


$$f(n) \leq f(A) < f(B)$$

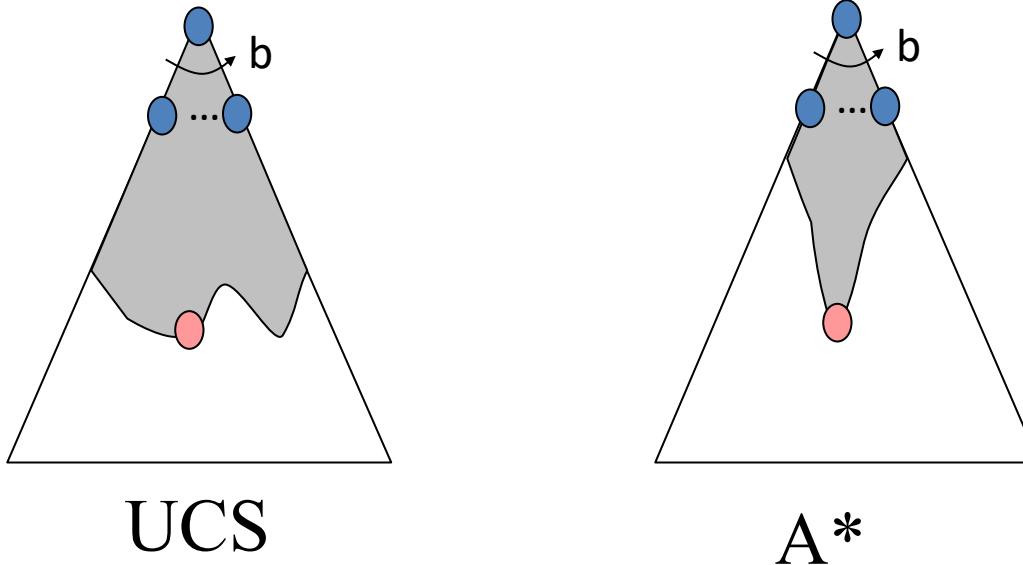
# Optimalitatea algoritmului A\*

Demonstrație:

- presupunem că B este pe frontieră
- un nod ascendent  $n$  al lui A (strămoș) este de asemenea pe frontieră (poate chiar nodul A!)
- afirmație:  $n$  va fi explorat înaintea nodului B
  1.  $f(n) \leq f(A)$
  2.  $f(A) < f(B)$
  3.  $n$  este explorat înaintea lui B
- toți ascendenții lui A sunt explorați înaintea lui B
- A este explorat înaintea lui B
- algoritmul A\* este optimal



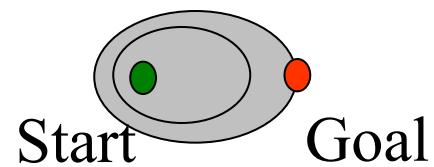
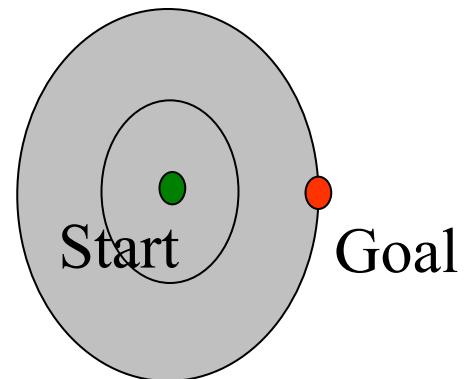
# Proprietățile algoritmului A\*



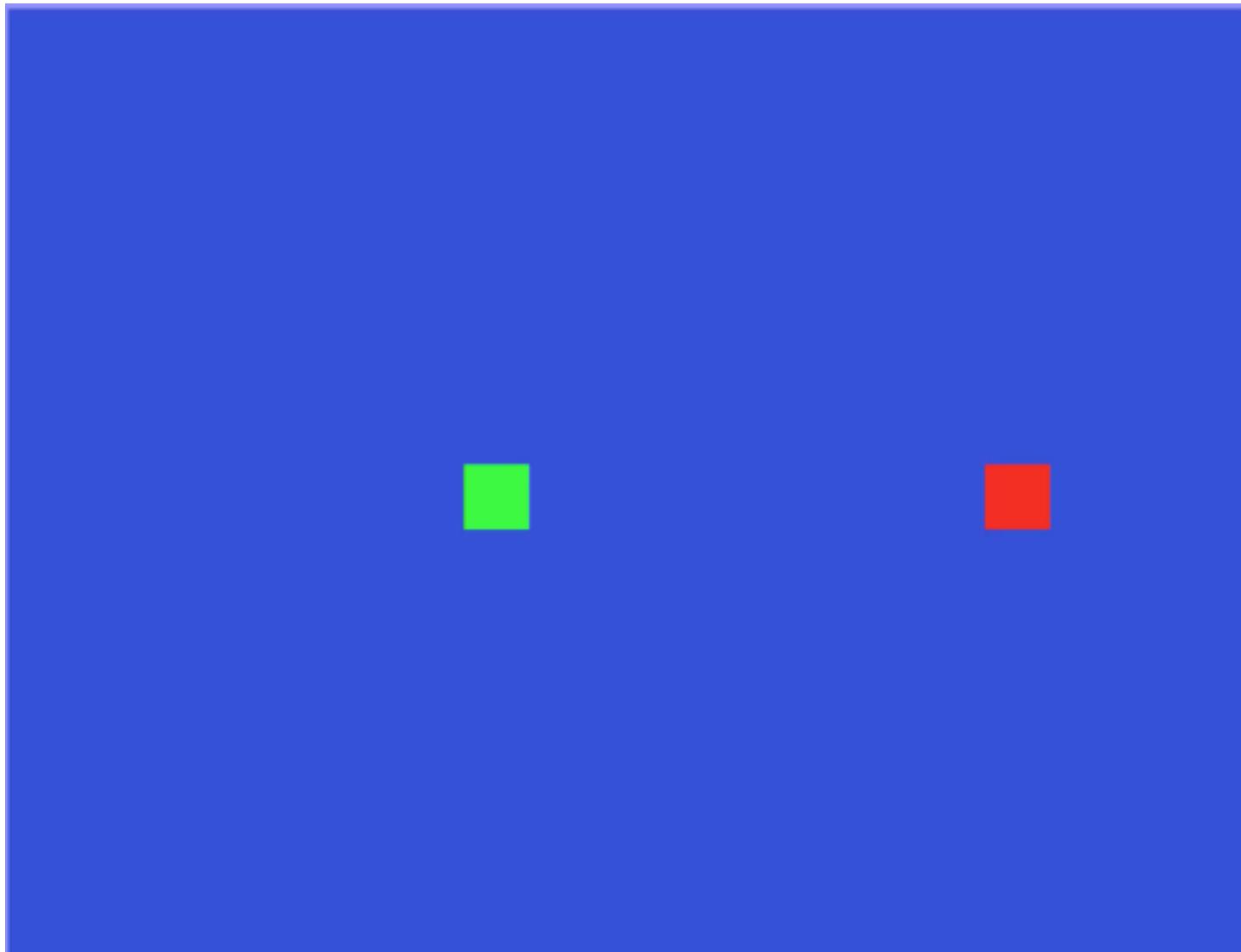
Căutare uniformă pe bază de cost explorează drumuri de cost minim, indiferent de cât de îndepărtate sunt de soluția-scop. Algoritmul A\* beneficiază de euristica  $h$  care restricționează căutarea.

# UCS vs A\* - explorarea arborelui de căutare

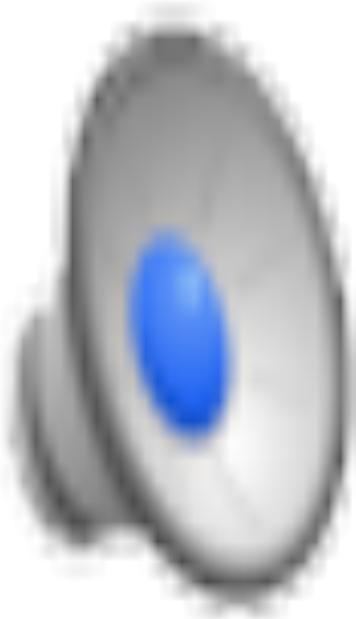
- Căutarea uniformă pe bază de cost (UCS) explorează soluții uniform în toate direcțiile.
- Algoritmul A\* explorează soluții în mare în direcția soluției, totuși păstrează și soluții în alte direcții (pentru păstrarea optimalității).



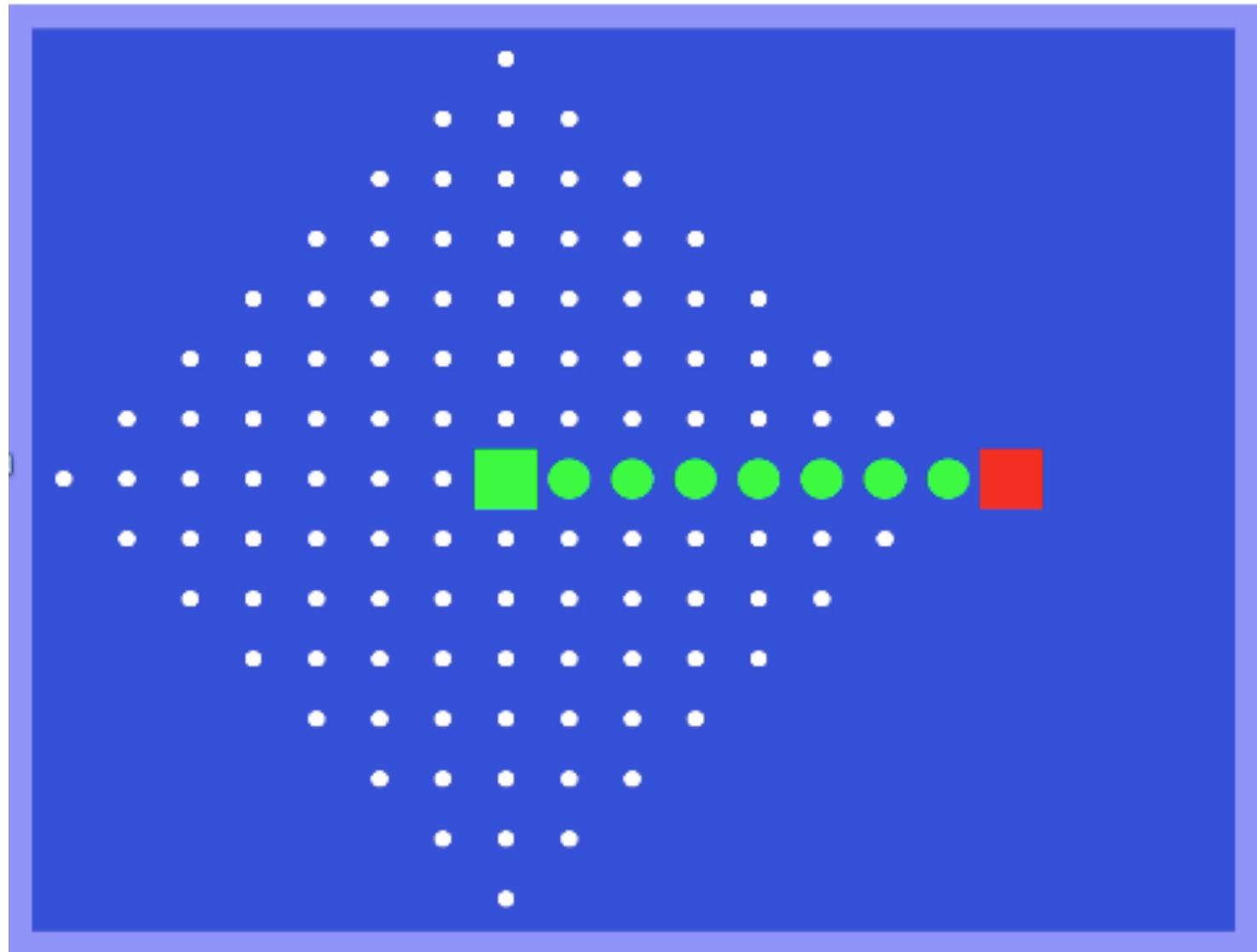
# Explorarea arborelui de căutare - demo



# Explorarea arborelui de căutare – demo UCS

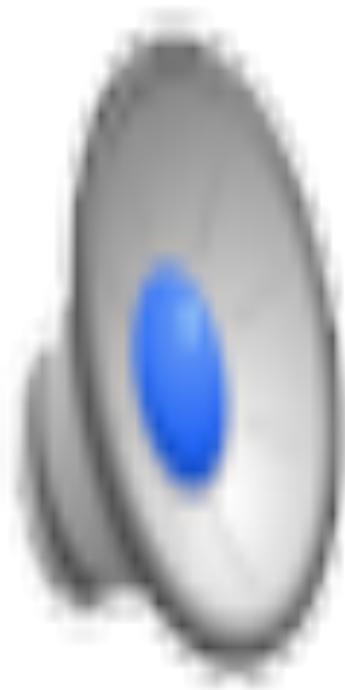


# Explorarea arborelui de căutare – demo UCS

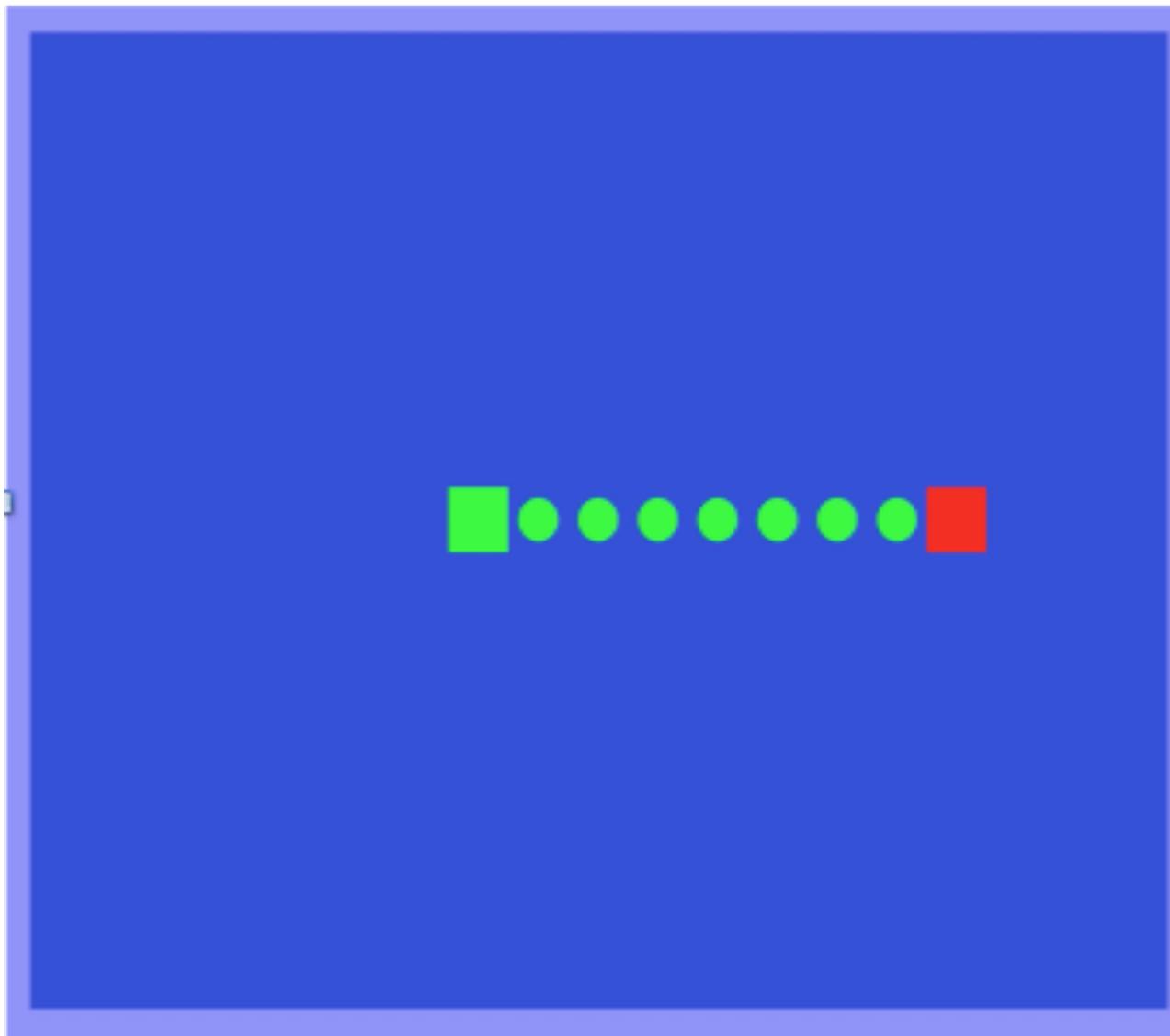


Soluția UCS

# Explorarea arborelui de căutare – demo Greedy

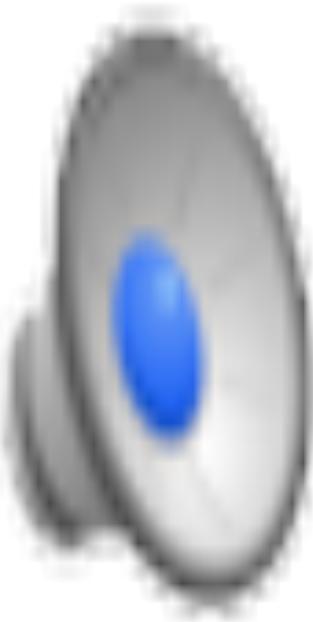


# Explorarea arborelui de căutare – demo Greedy

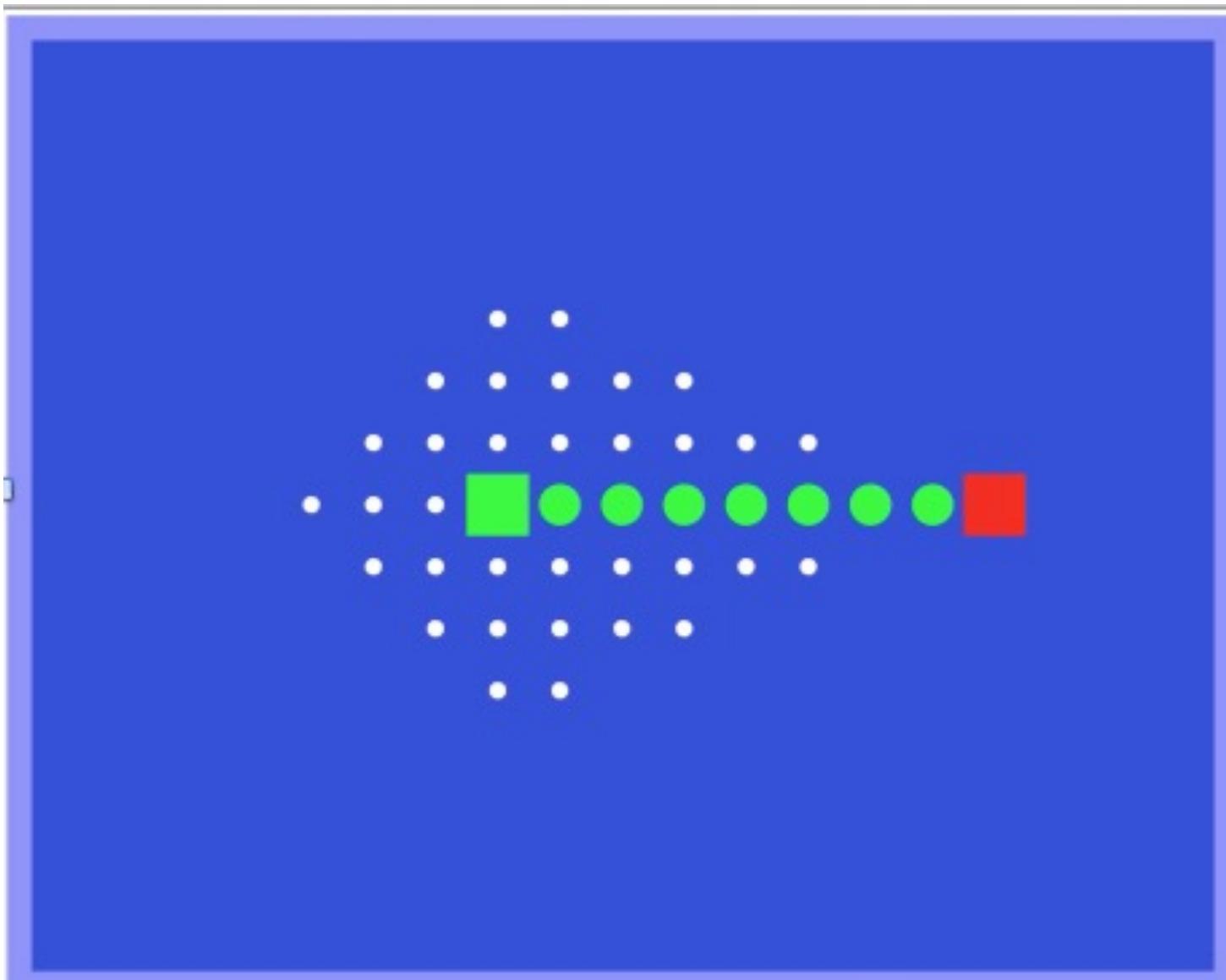


Soluția Greedy

# Explorarea arborelui de căutare – demo A\*



# Explorarea arborelui de căutare – demo A\*

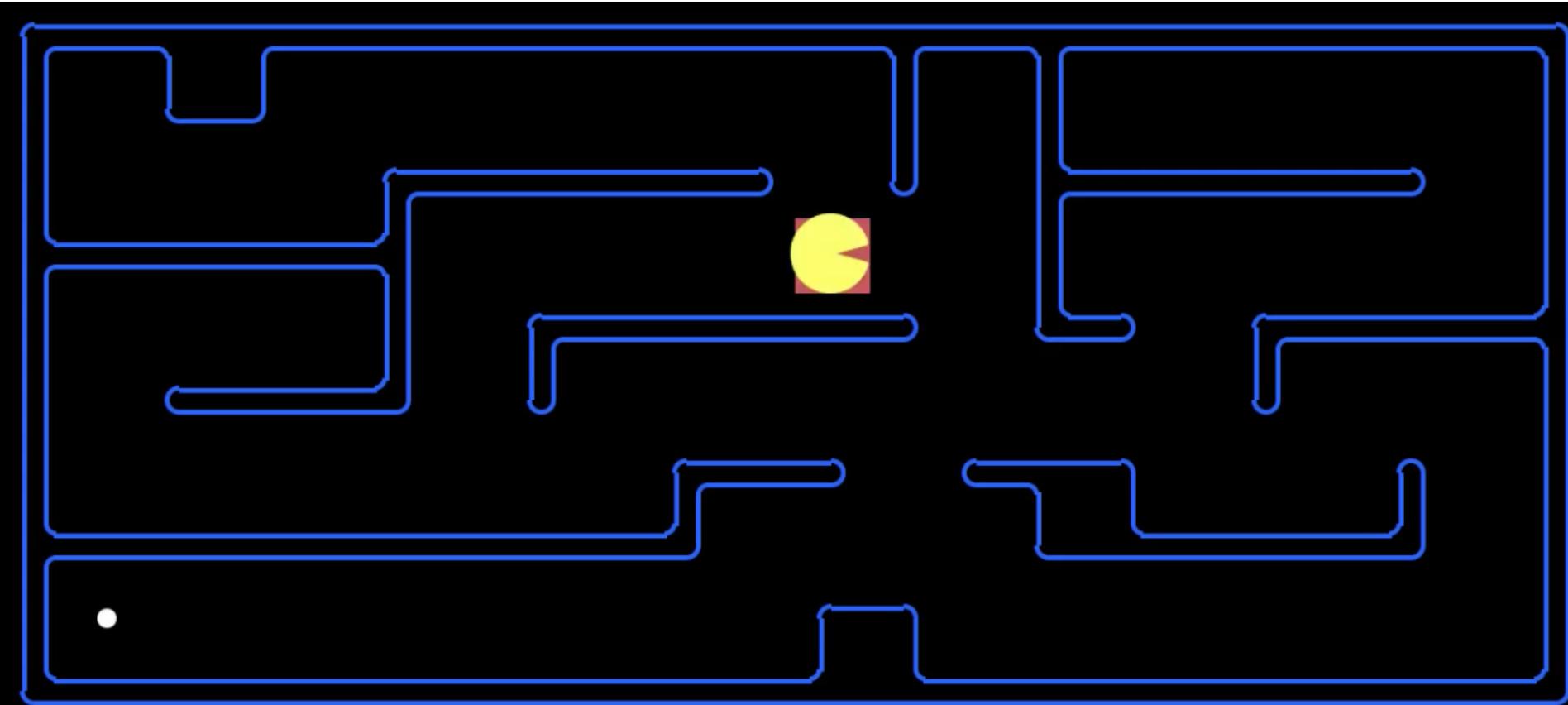


Soluția A\*

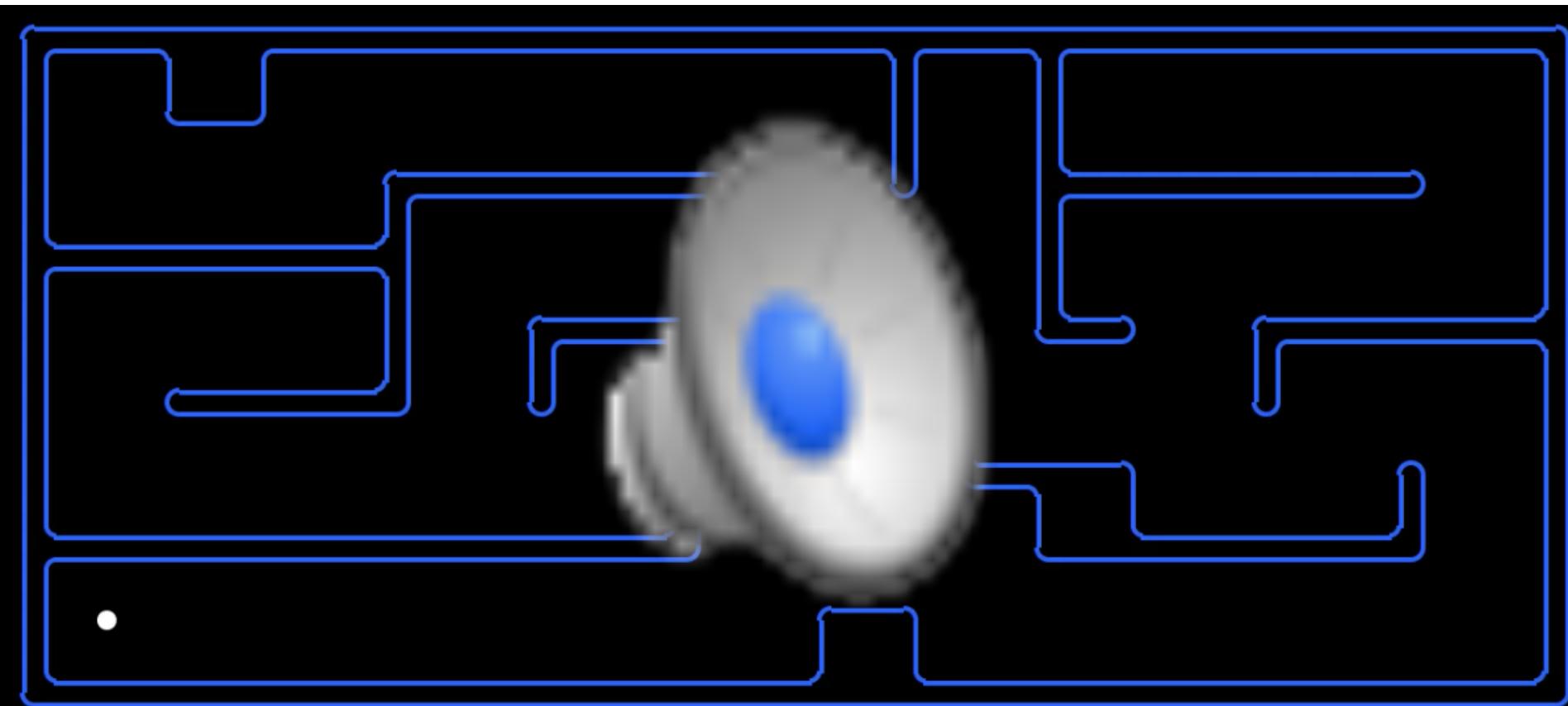
# Demo A<sup>\*</sup> - explorarea arborelui de căutare



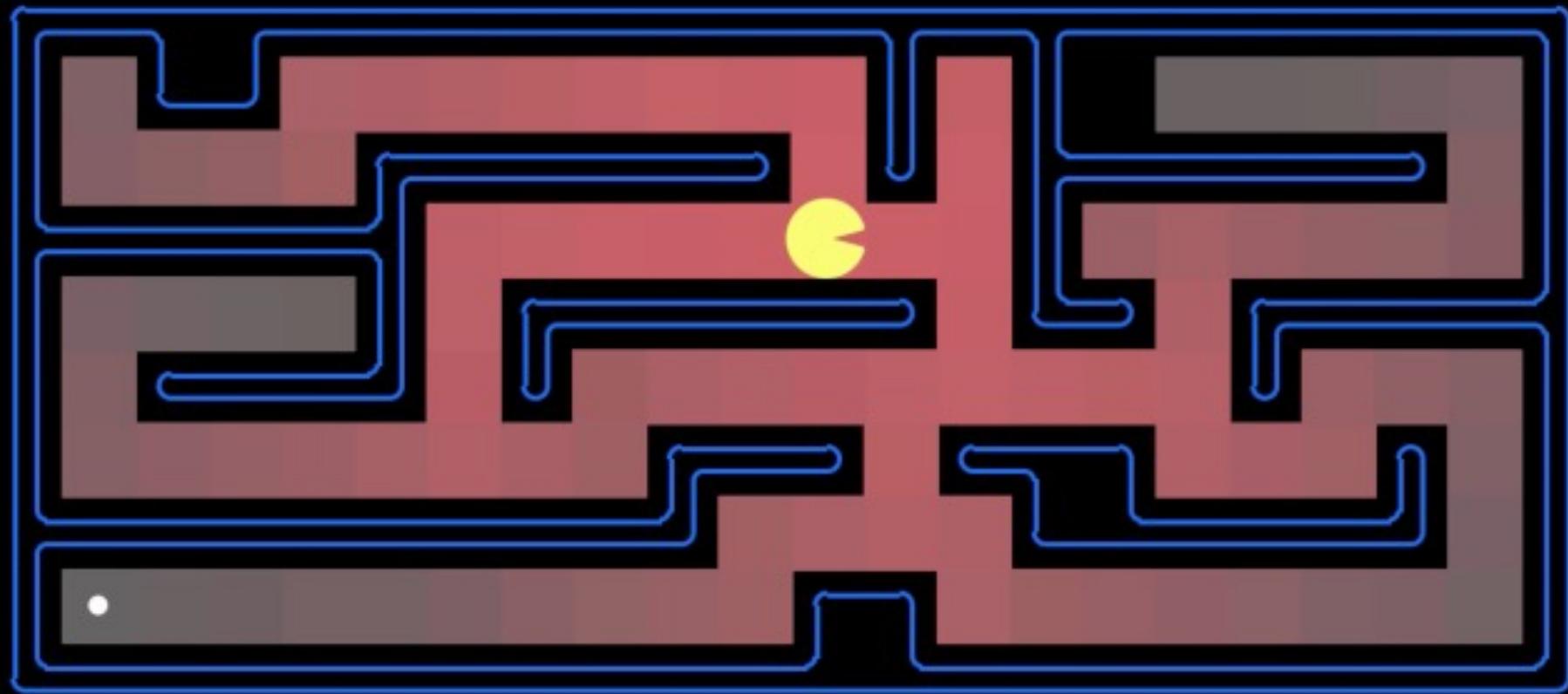
# Explorarea arborelui de căutare - demo



# Explorarea arborelui de căutare – demo UCS



# Explorarea arborelui de căutare – demo UCS



**SCORE: 0**

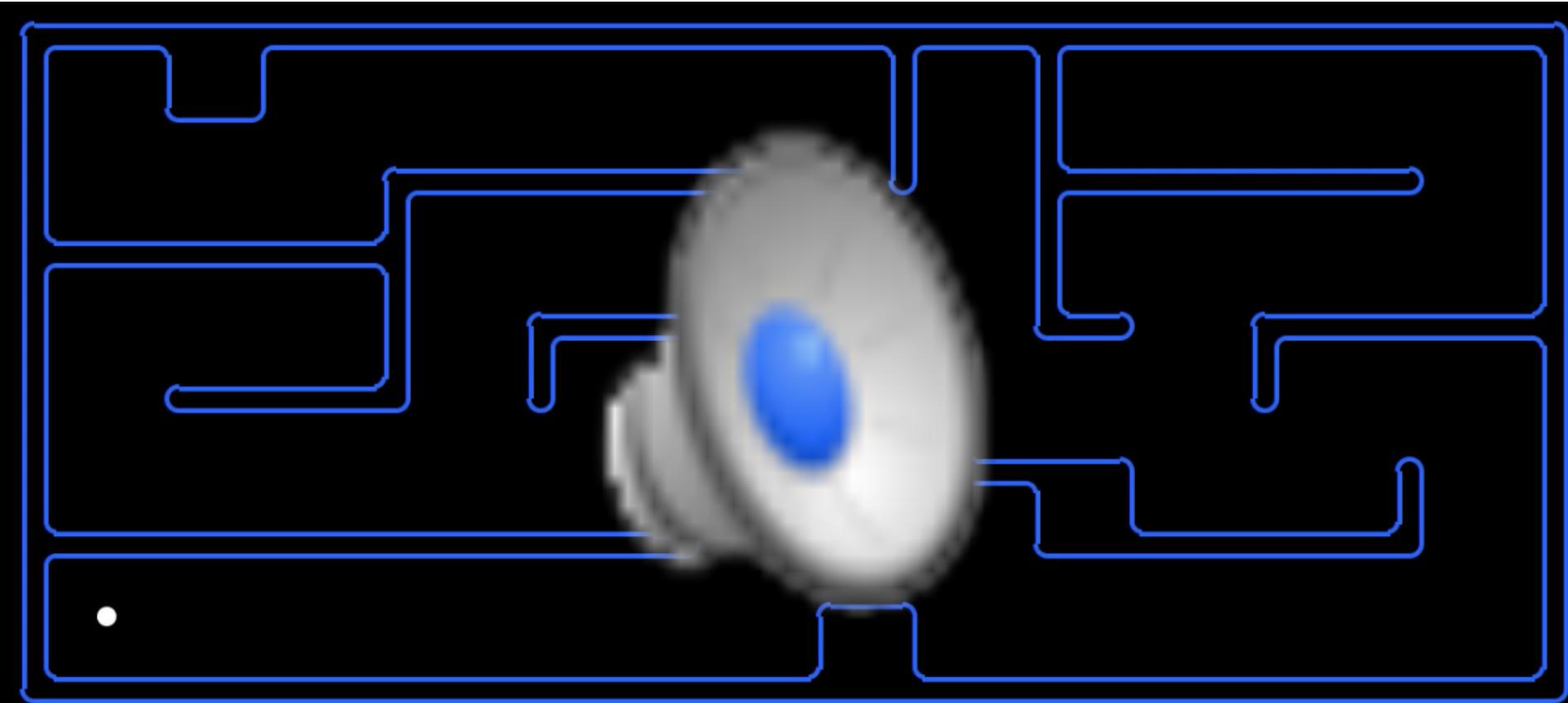
Ordinea de explorare a nodurile este dată de culoare:

**Roșu intens – noduri exploreate la început**

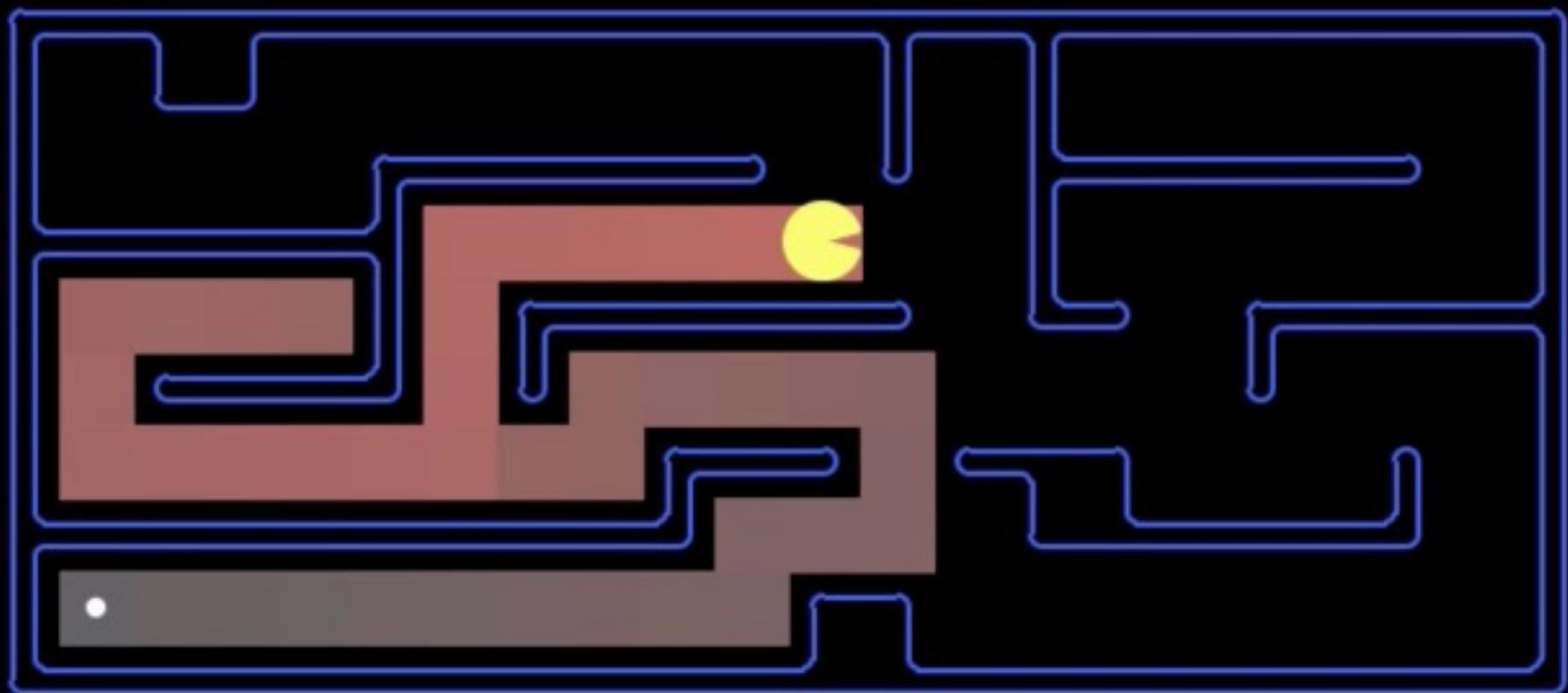
**Gri – noduri exploreate la sfârșit**

**Soluția UCS**

# Explorarea arborelui de căutare – demo Greedy



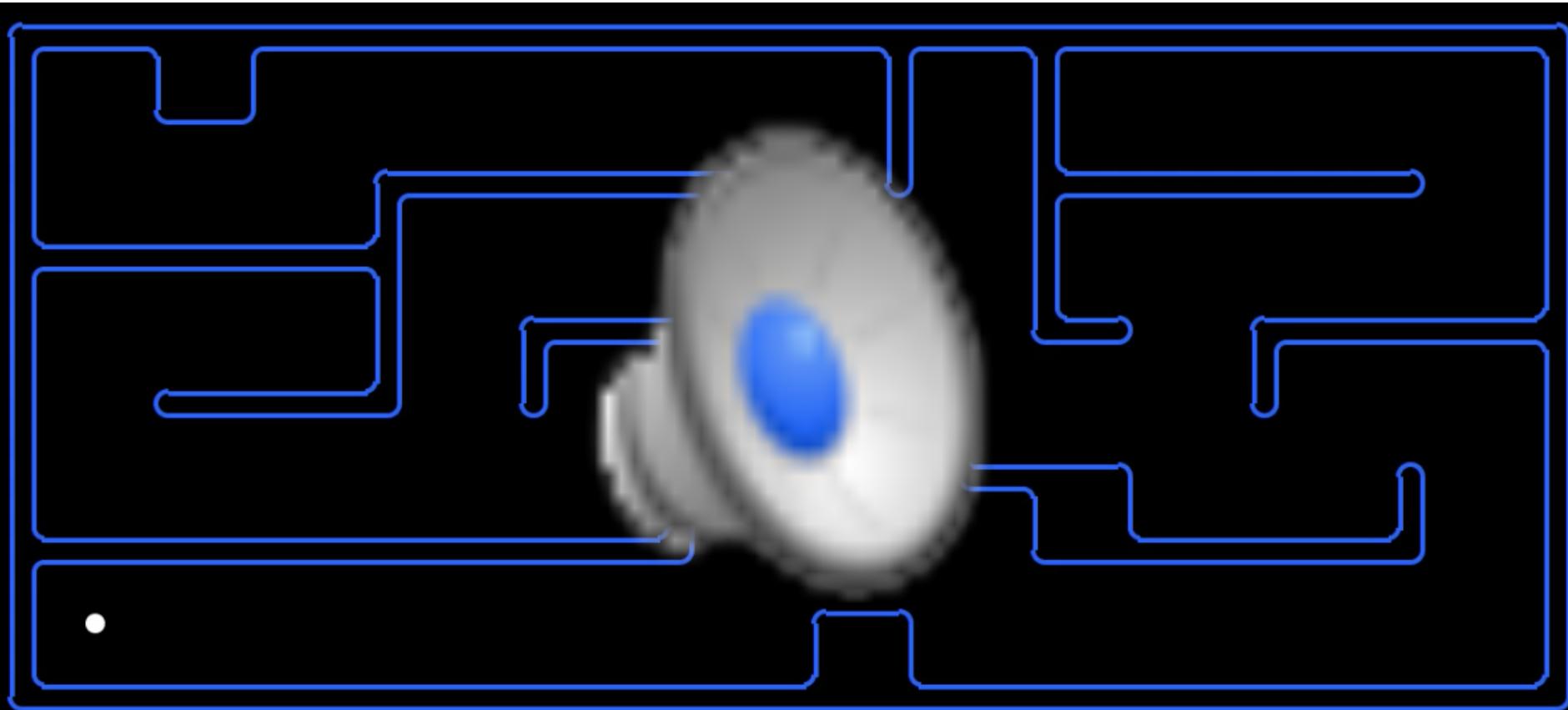
# Explorarea arborelui de căutare – demo Greedy



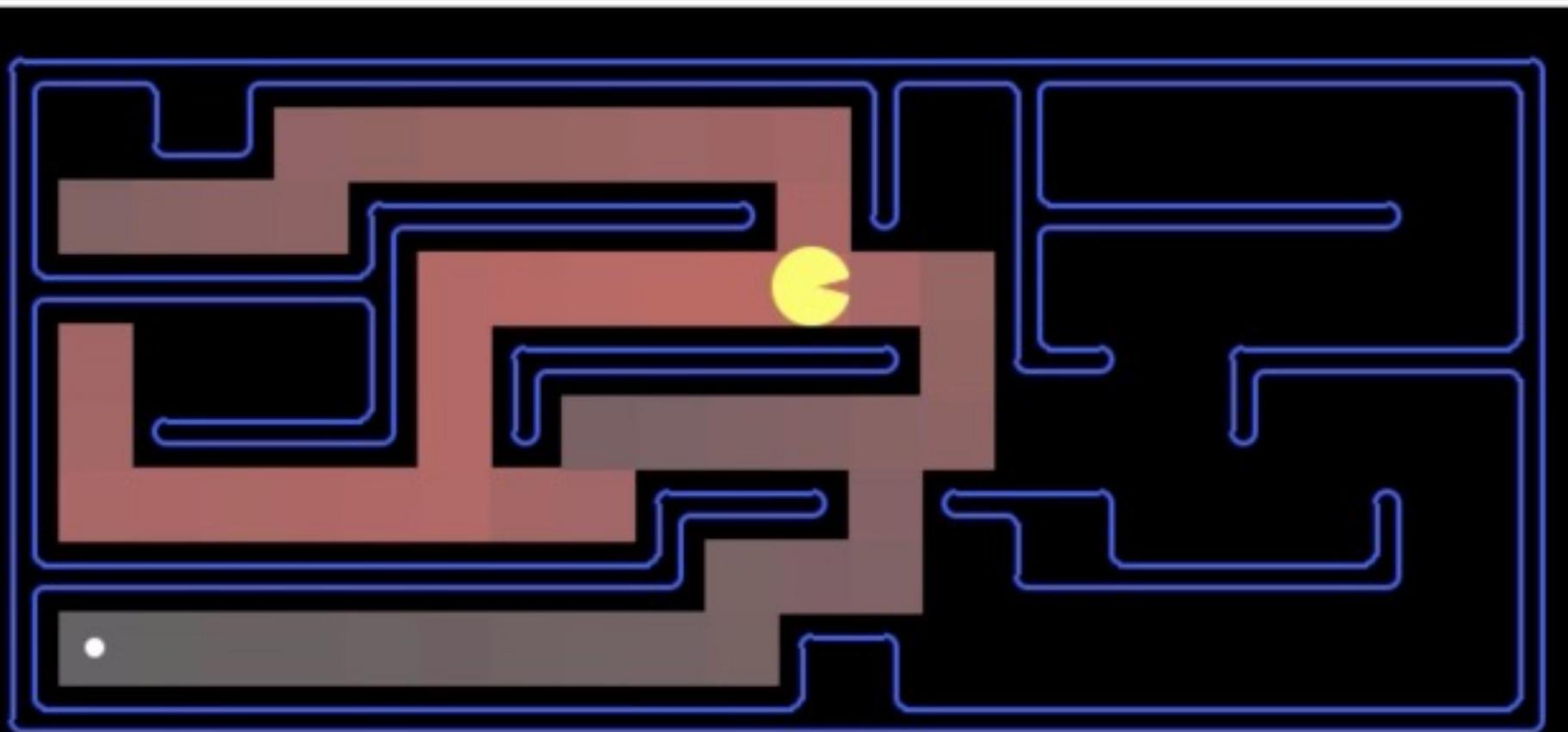
**SCORE:** 0

**Soluția Greedy**

# Explorarea arborelui de căutare – demo A\*



# Explorarea arborelui de căutare – demo A\*



**SCORE:** 0

**Soluția A\***

# Comparare Pac-Man



Greedy

suboptimal

rapid



UCS

optimal

incent



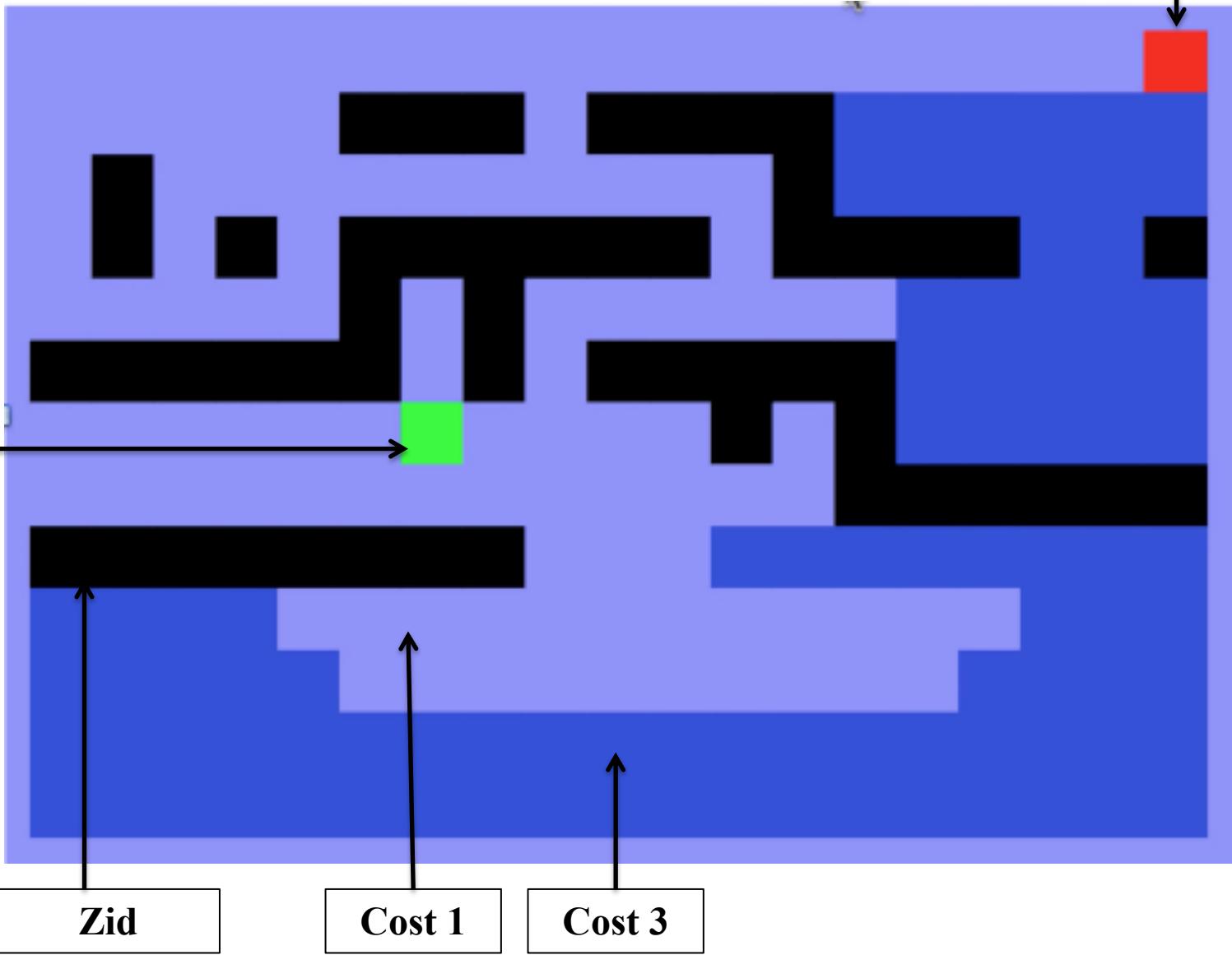
A\*

optimal

rapid

NOD  
SCOP

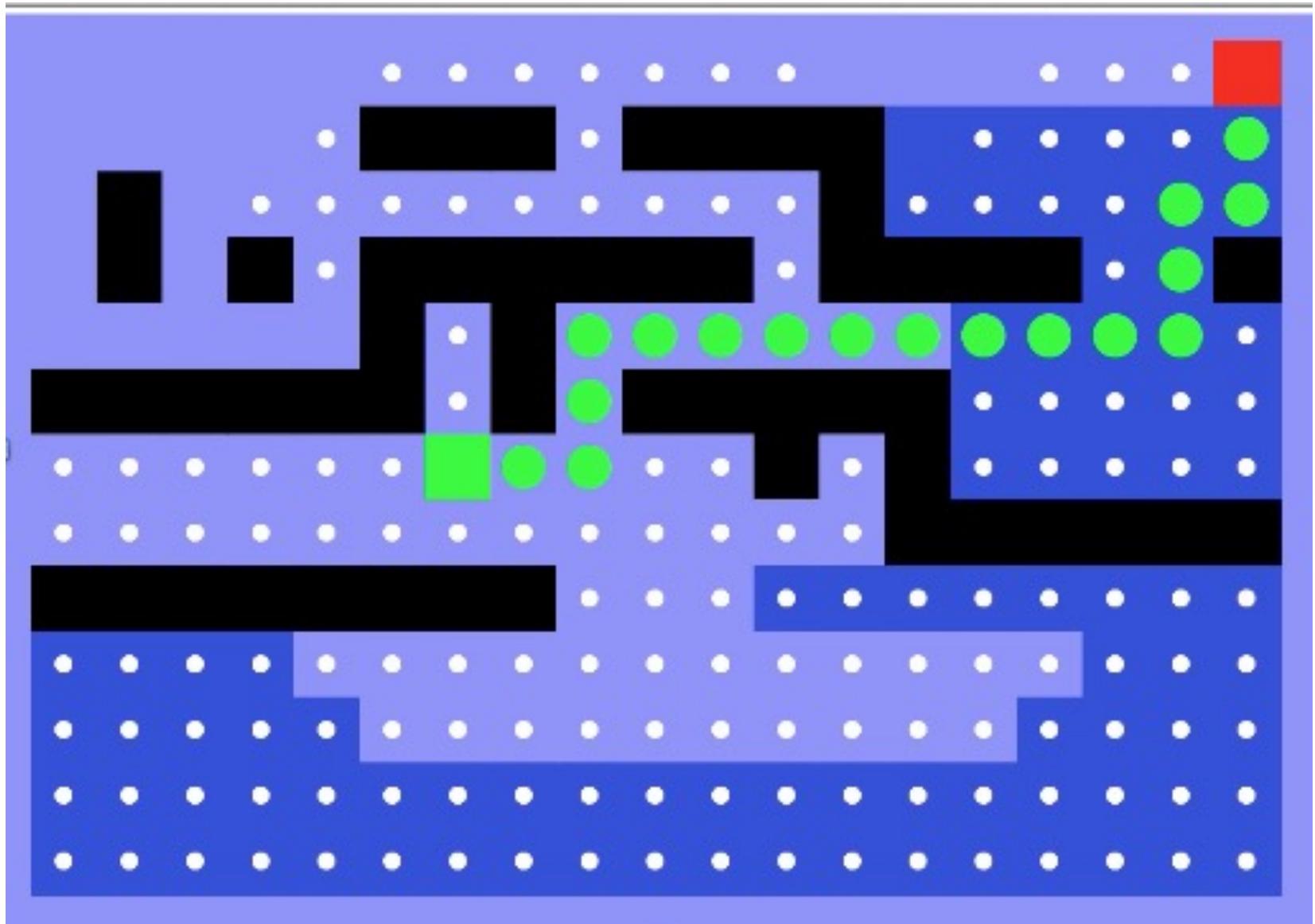
# Labirint - demo



# Labirint – demo Bread First search



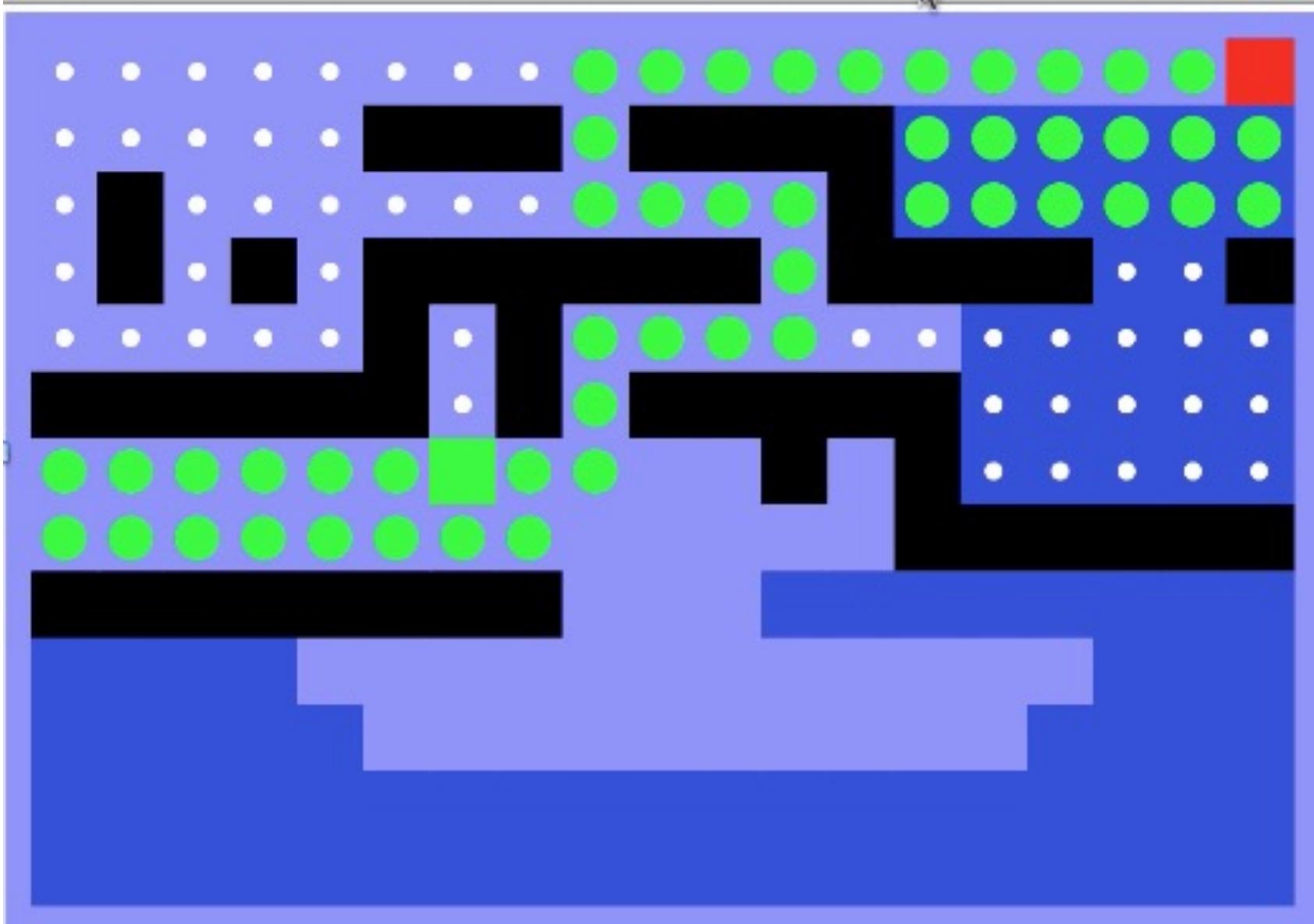
# Labirint – demo Bread First search



# Labirint – demo Depth First search



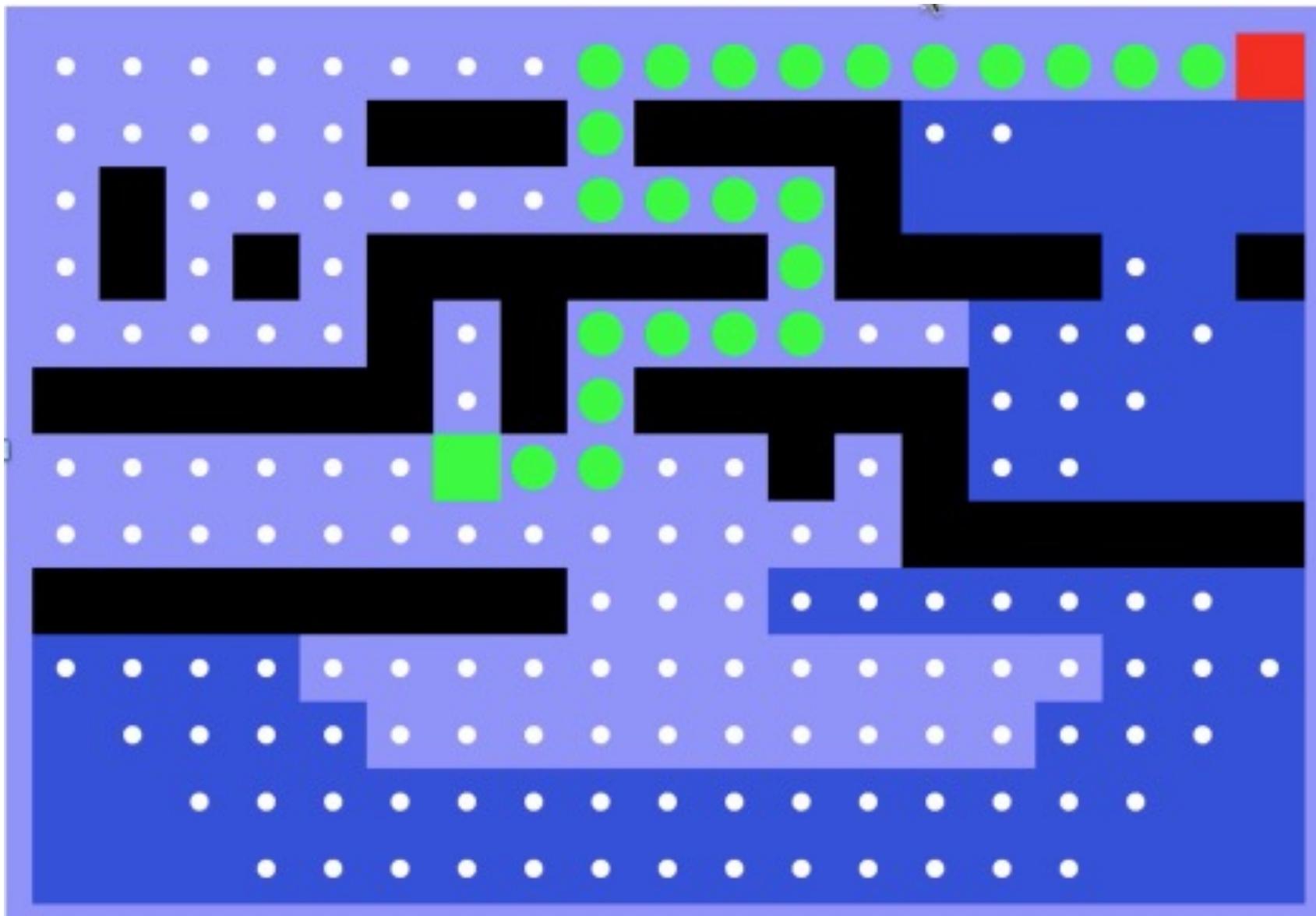
# Labirint – demo Depth First search



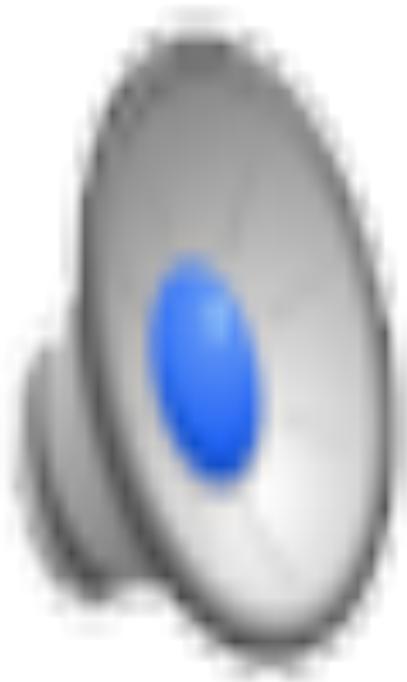
# Labirint – demo Uniform Cost Search



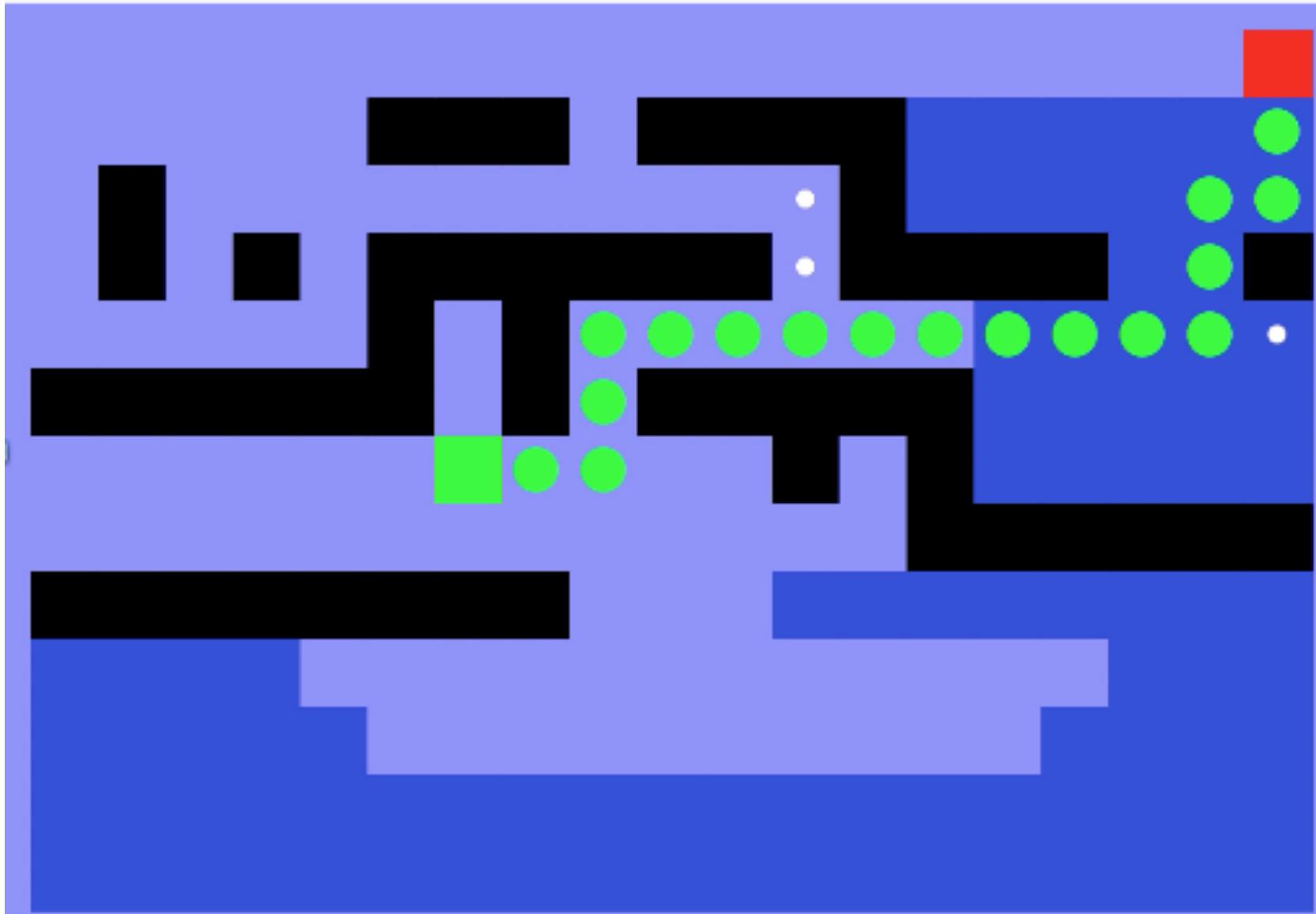
# Labirint – demo Uniform Cost Search



# Labirint – demo Greedy search



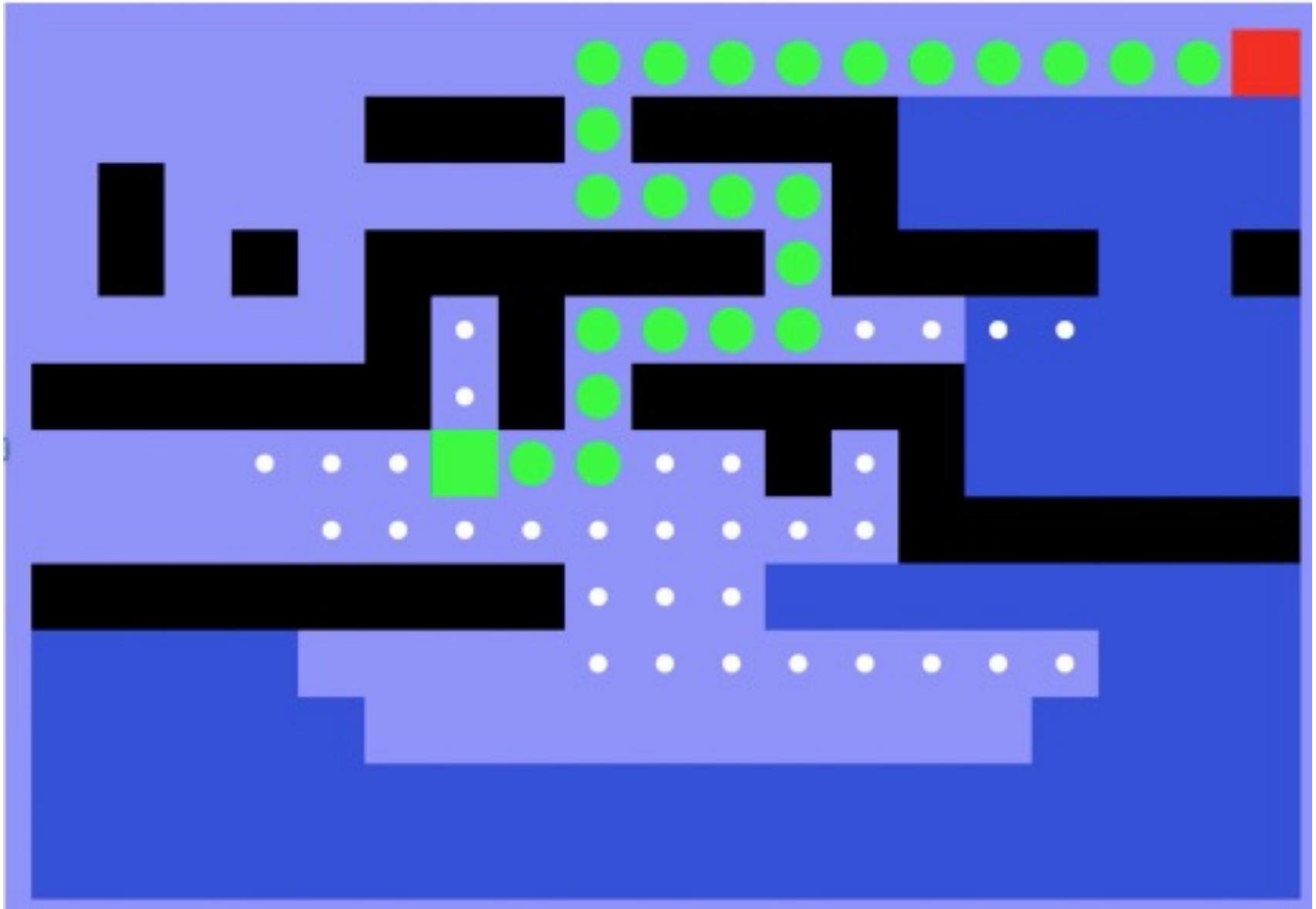
# Labirint – demo Greedy search



# Labirint – demo A\*



# Labirint – demo A\*



# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 11

# Recapitulare – cursul trecut

## 1. Căutare neinformată

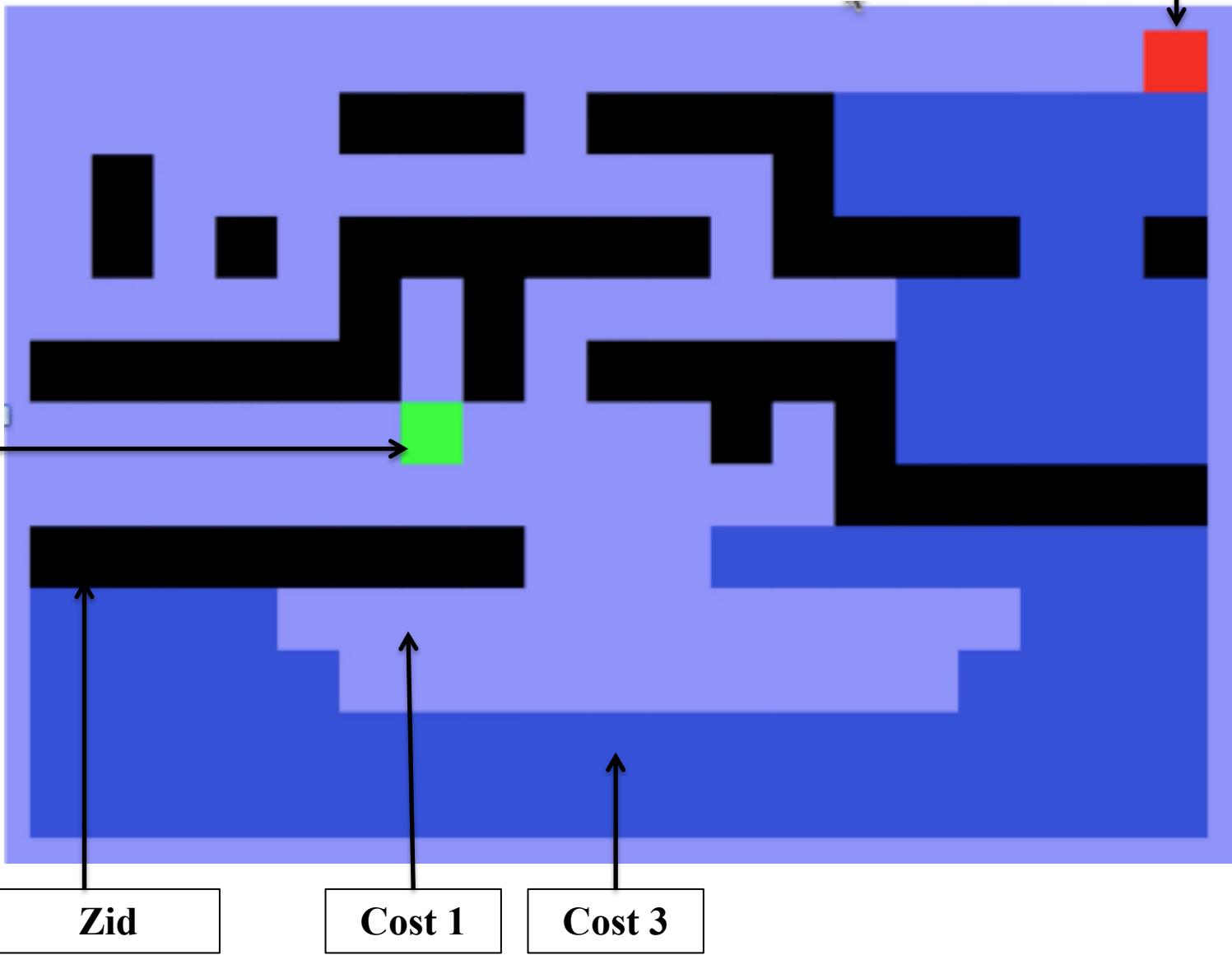
- Căutare în adâncime incrementală (iterative deepening search)
- Căutare uniformă după cost (uniform-cost search)

## 2. Căutare informată

- căutare Greedy
- algoritmul A\*
- euristici admisibile, dominante și banale

NOD  
SCOP

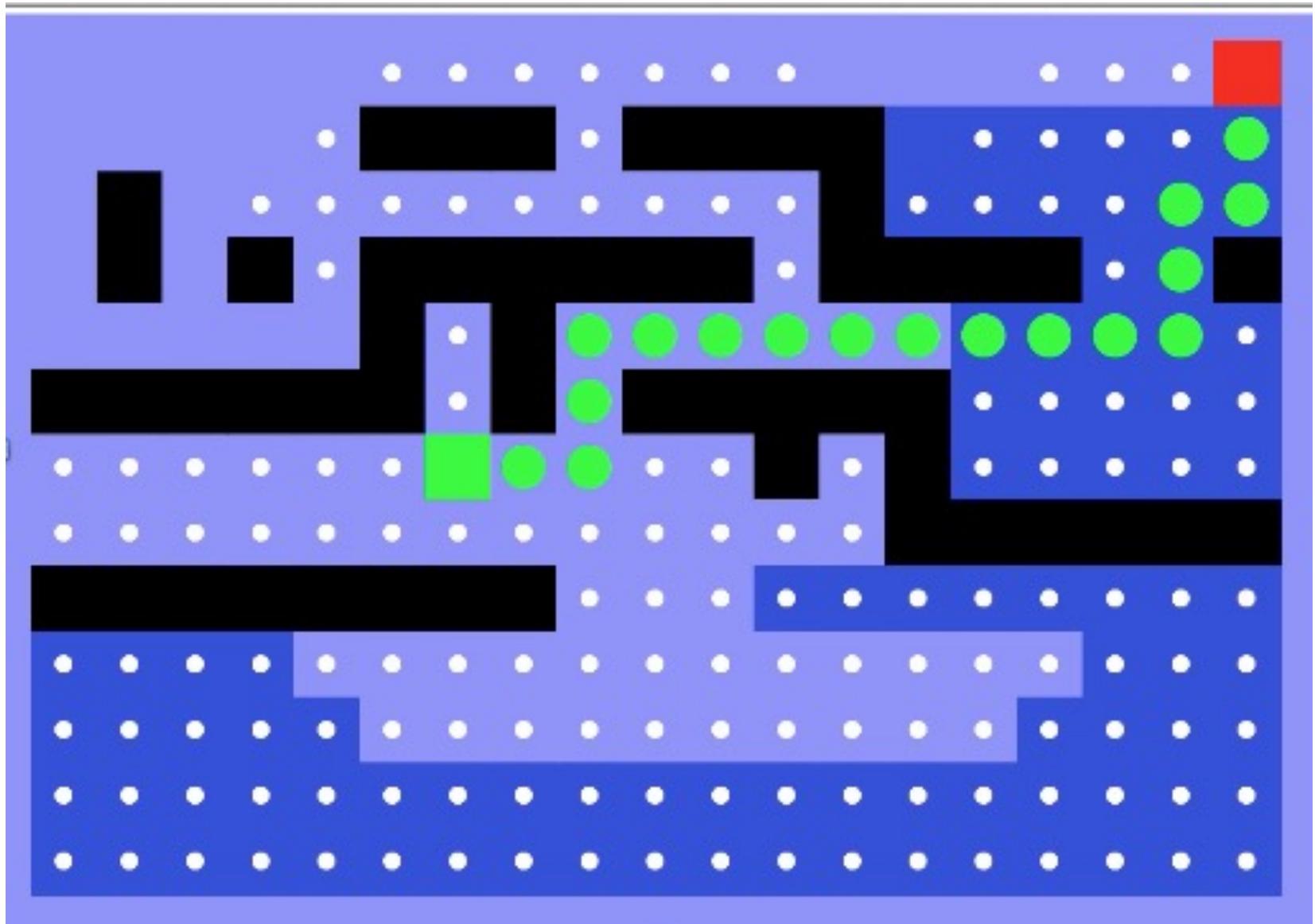
# Labirint - demo



# Labirint – demo Bread First search



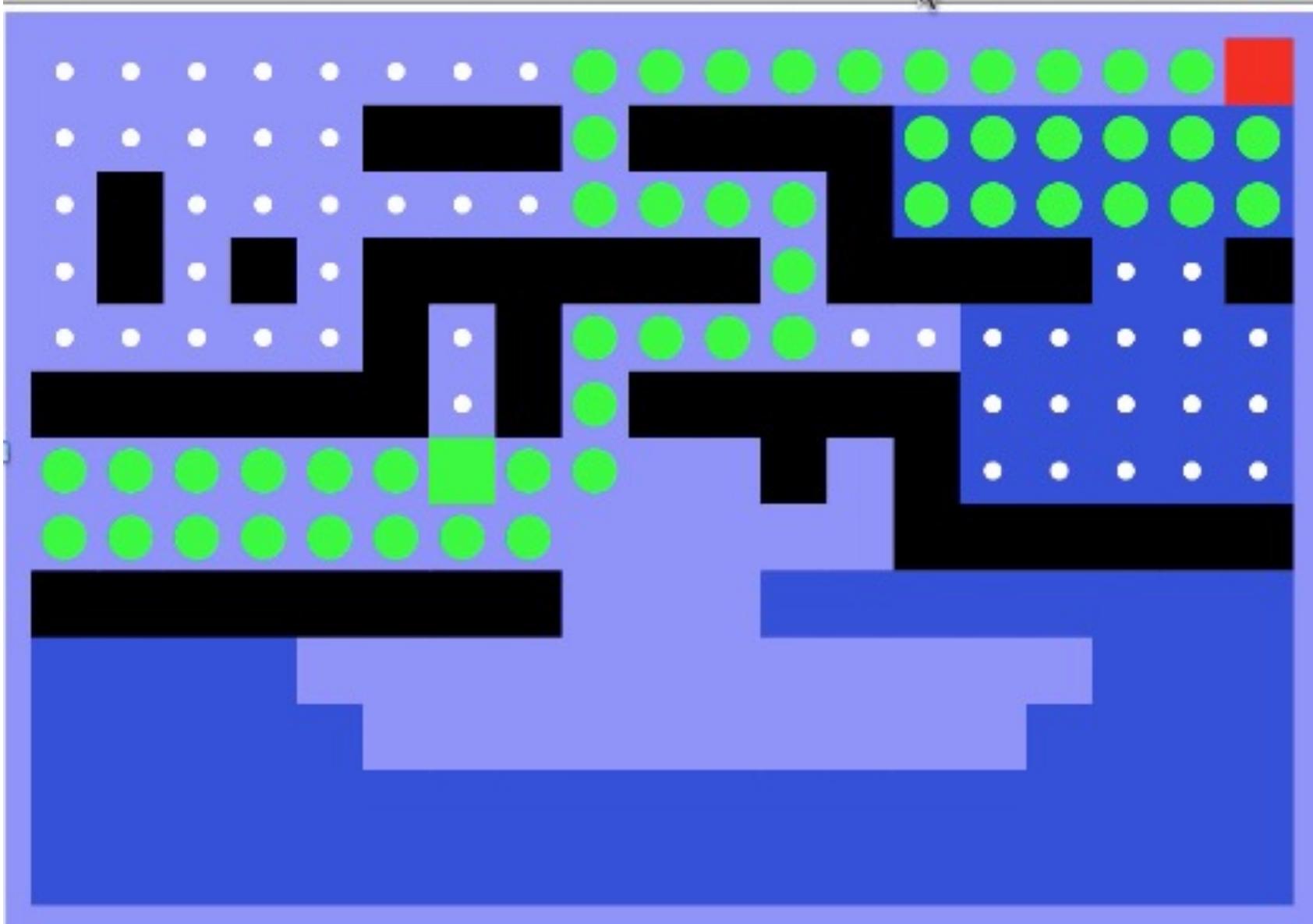
# Labirint – demo Bread First search



# Labirint – demo Depth First search



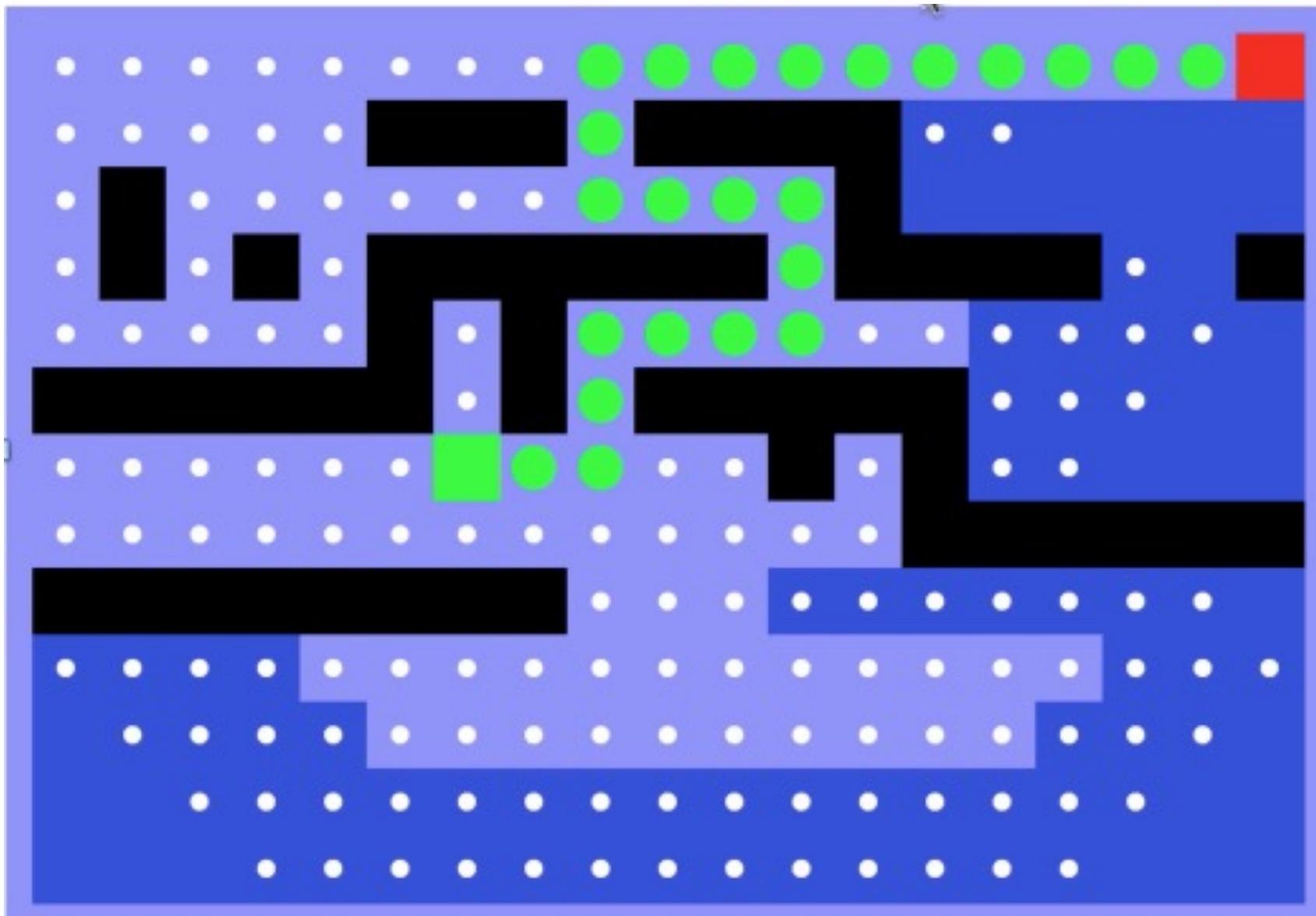
# Labirint – demo Depth First search



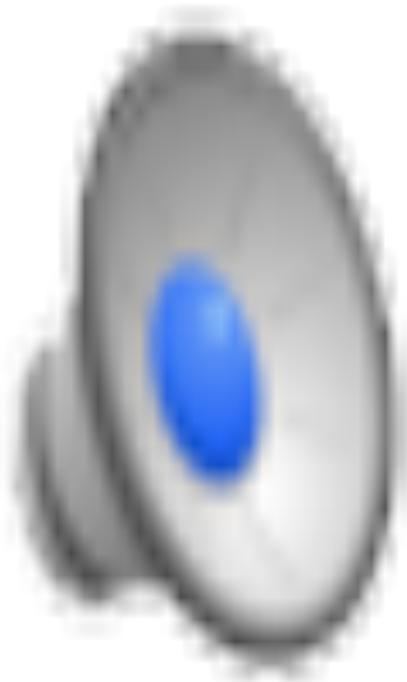
# Labirint – demo Uniform Cost Search



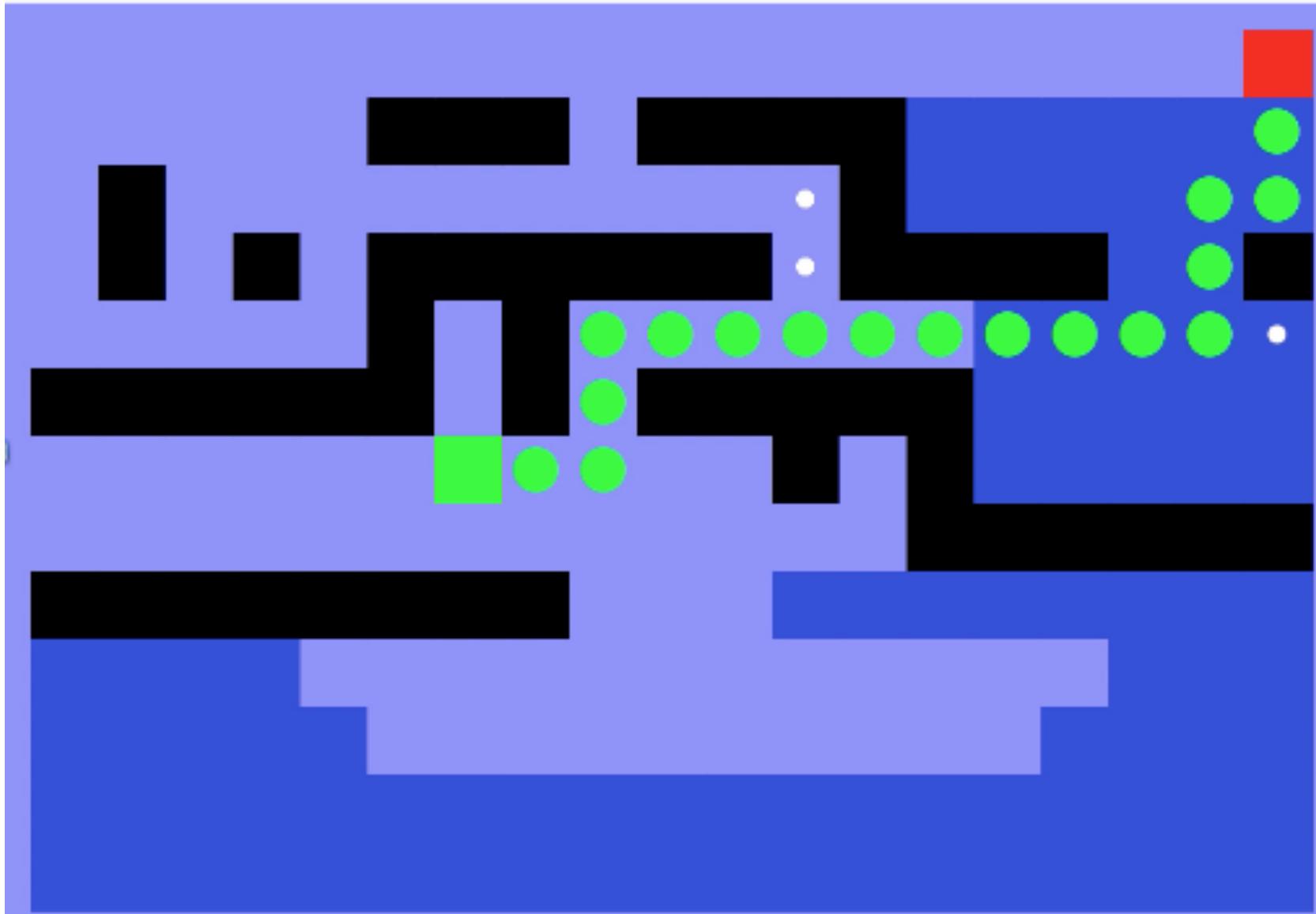
# Labirint – demo Uniform Cost Search



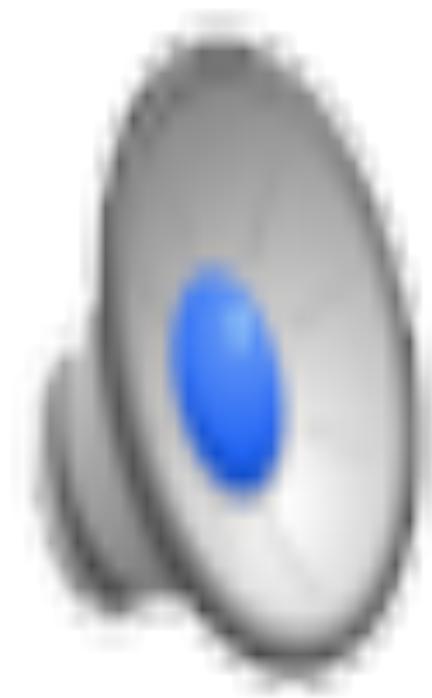
# Labirint – demo Greedy search



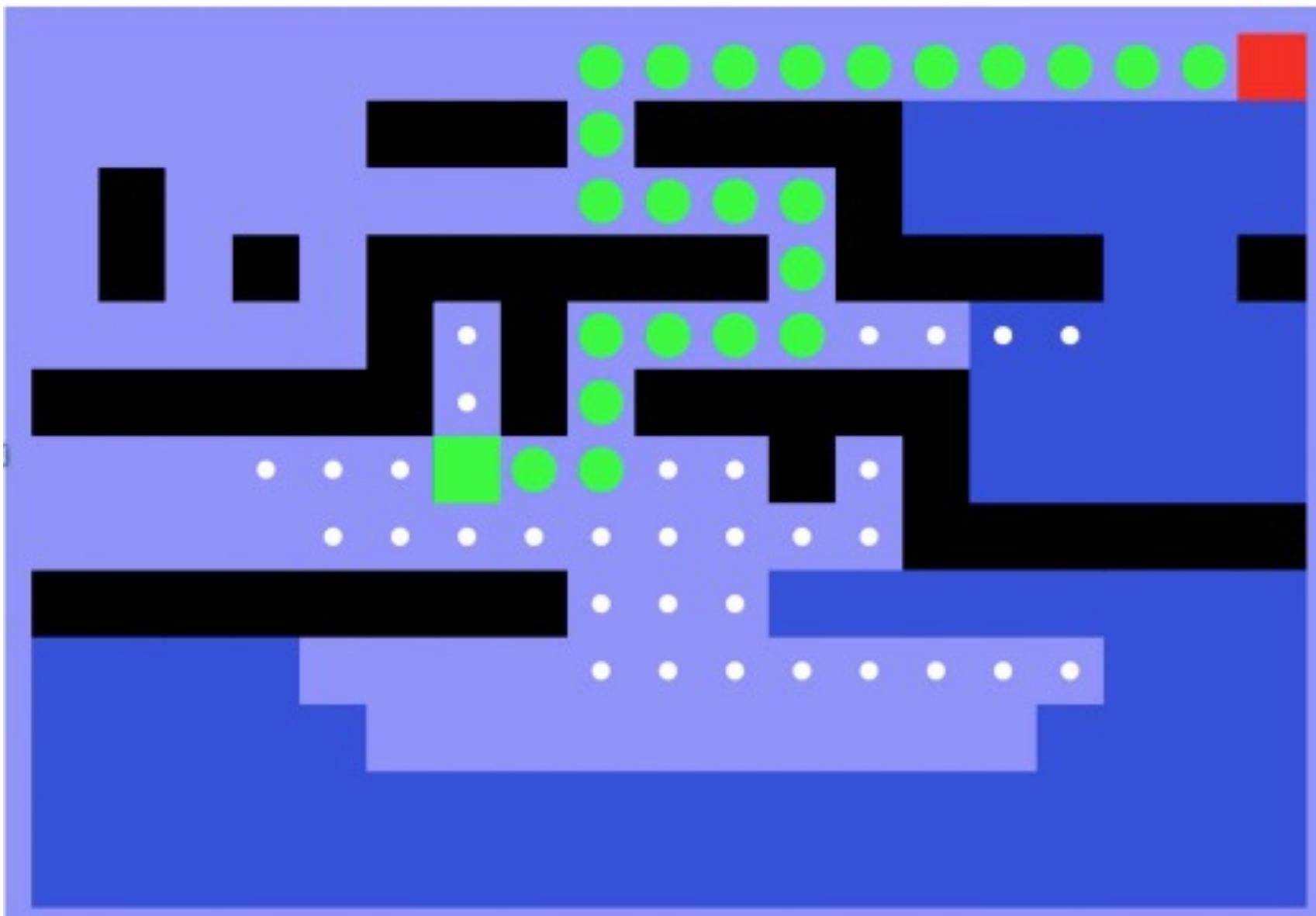
# Labirint – demo Greedy search



# Labirint – demo A\*



# Labirint – demo A\*



# Cuprinsul cursului de azi

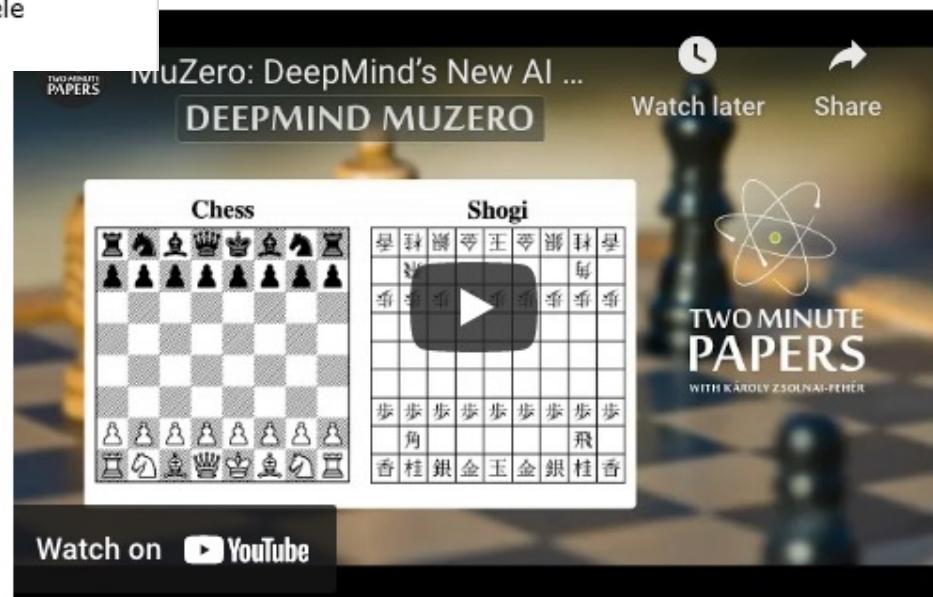
## 1. Căutare adversarială:

- jocuri adversariale
- algoritmul MINIMAX
- algoritmul  $\alpha$ - $\beta$  retezare ( $\alpha$ - $\beta$  pruning)

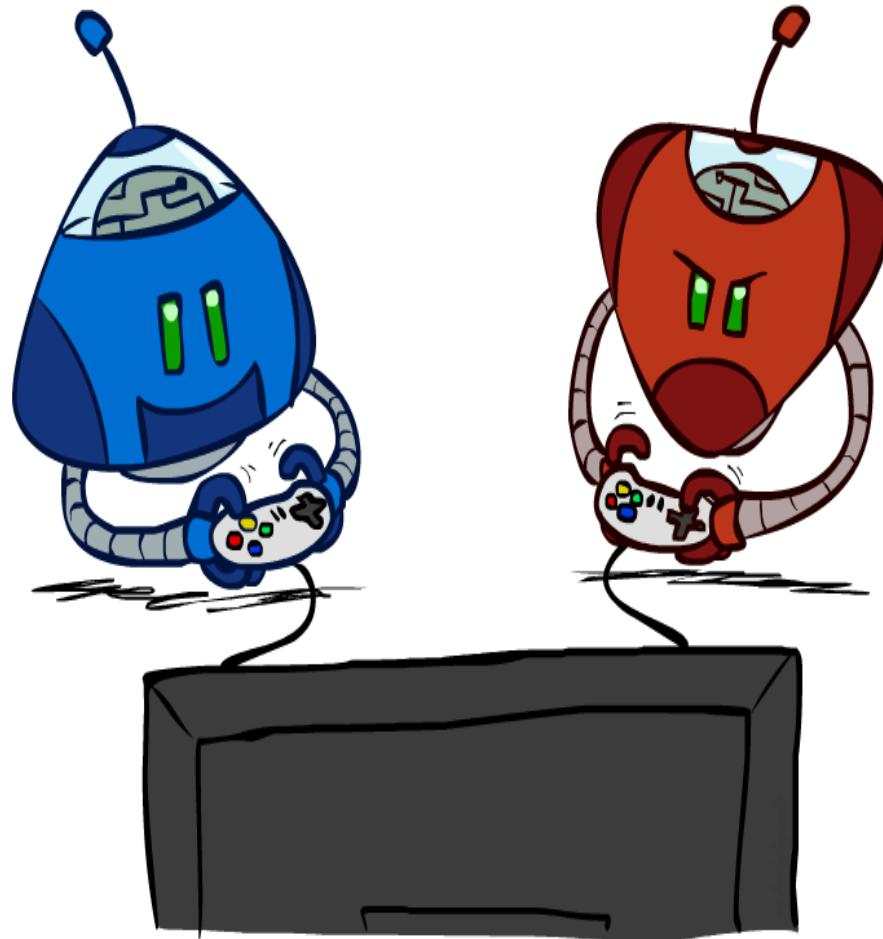
**Un software al DeepMind a depășit o nouă barieră.** DeepMind, compania de cercetare în domeniul inteligenței artificiale deținută de Alphabet (compania-mamă a Google), a dezvoltat un program care poate învăța jocuri complexe fără a cunoaște în prealabil regulile lor.

Cei de la DeepMind s-au remarcat anterior prin progresele revoluționare înregistrate în dezvoltarea unor programe care să învețe jocuri precum Go sau Shogi (un alt joc de strategie chinezesc), precum și în cazul săhului sau a unor jocuri video dezvoltate de compania Atari. Însă în toate cazurile anterioare realizările s-au bazat pe cunoașterea de la început a regulilor jocurilor respective.

Noul software MuZero dezvoltat de DeepMind a reușit însă acest lucru fără a cunoaște de la început regulile, programatorii companiei bazându-se pe un principiu numit „căutare anticipată” (look-ahead search). Aceasta evaluează un număr de mutări potențiale în funcție de maniera în care un adversar ar răspunde la ele. Deși există un număr incredibil de mari de permutări posibile în jocuri complexe precum săhul, MuZero acordă prioritate celor mai relevante și probabile mișcări, învățând din partidele de succes și evitându-le pe cele pierzătoare.



# Căutare adversarială



# Proprietăți ale mediilor în care funcționează agenții

- Complet observabil vs parțial observabil
- Determinist vs stochastic
- Discret vs continuu
- Episodic vs secvențial
- Static vs dinamic
- Agent vs multiagent

# Proprietăți ale mediilor în care funcționează agenții

- **Complet observabil vs parțial observabil:** la orice moment în timp mediul este complet observabil dacă senzorii agentului îi dau acces complet la starea mediului la fiecare moment și îi oferă toate aspectele relevante în luarea deciziei optime.
  - săh vs poker
- **Determinist vs stochastic:** dacă următoarea stare a mediului este în întregime determinată de starea curentă și de acțiunile agentului, mediul este determinist.
  - săh vs table (dau cu zarurile)

# Proprietăți ale mediilor în care funcționează agenții

- **Discret vs continuu:** dacă numărul de percepții și de acțiuni ale agentului sunt finite atunci mediul este discret.
  - șah vs condusul unei mașini
- **Episodic vs sevențial:** într-un mediu episodic, experiența agentului este împărțită în episoade independente. Un episod constă în percepție urmată de o singură acțiune. Episoadele următoare nu depind de acțiunile episoadelor anterioare.
  - corectarea unui test grilă vs șah

# Proprietăți ale mediilor în care funcționează agenții

- **Static vs dinamic:** dacă mediul se schimbă cât timp agentul decide ce acțiuni să facă mediul este dinamic.
  - rezolvarea unui rebus vs condusul unei mașini
- **Agent vs multiagent:** dacă sunt mai mulți agenți mediul este multiagent (sisteme multiagent competitive vs sisteme multiagent cooperative).
  - corectarea unui test grilă vs șah

# Jocul de dame



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs sevențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Determinist

Discret

Sevențial

Static

Multiagent competitiv

# Jocul de poker



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs sevențial

Static vs dinamic

Agent vs multiagent

Parțial observabil

Stochastic

Discret

Sevențial

Static

Multiagent competitiv

# 8 - puzzle

1	2	3
8		4
7	6	5

Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs sevențial

Static vs dinamic

Agent vs multiagent

Complet observabil

Determinist

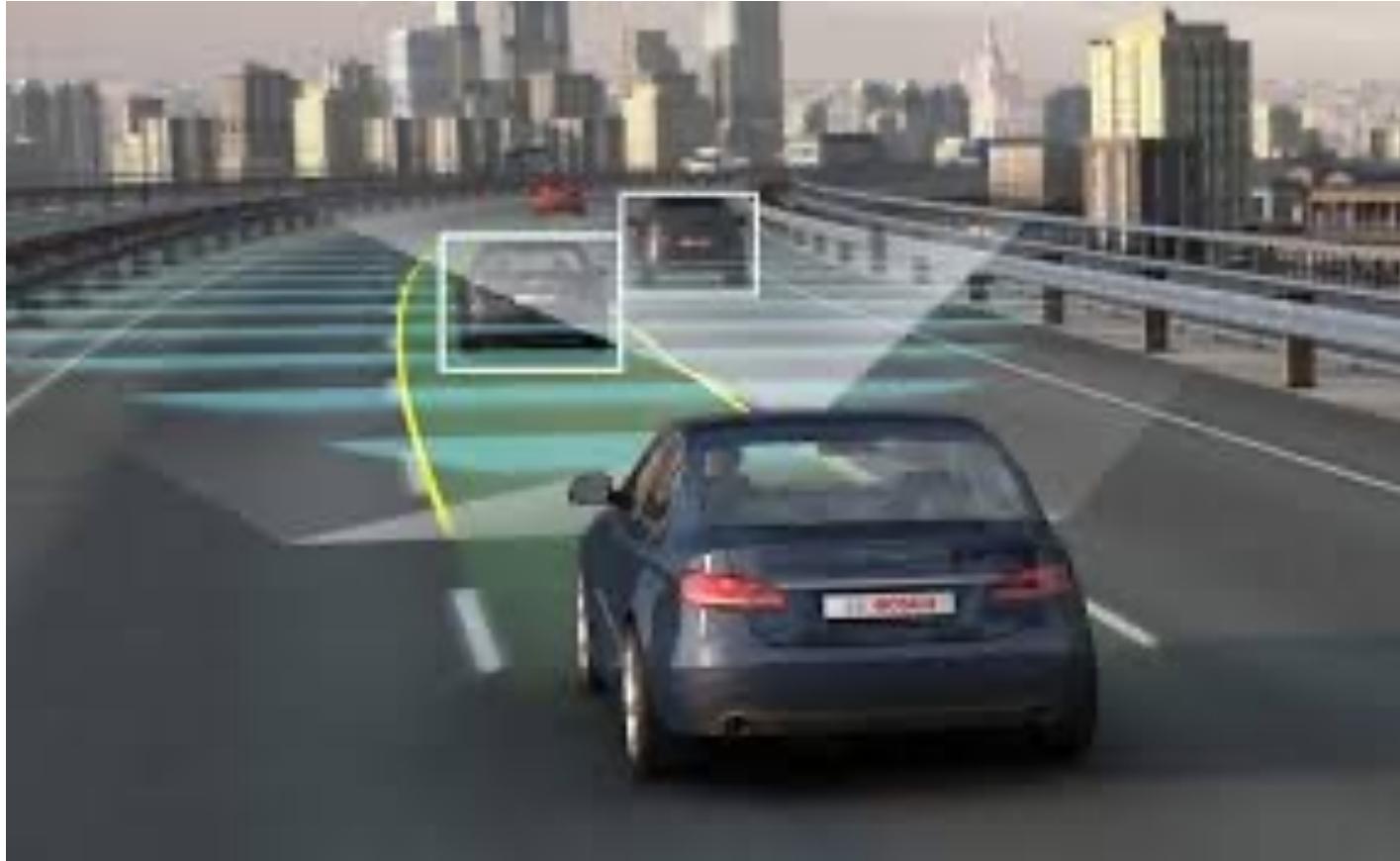
Discret

Sevențial

Static

Agent

# Mașină autonomă



Complet observabil vs parțial observabil

Determinist vs stochastic

Discret vs continuu

Episodic vs secvențial

Static vs dinamic

Agent vs multiagent

Parțial observabil

Stochastic

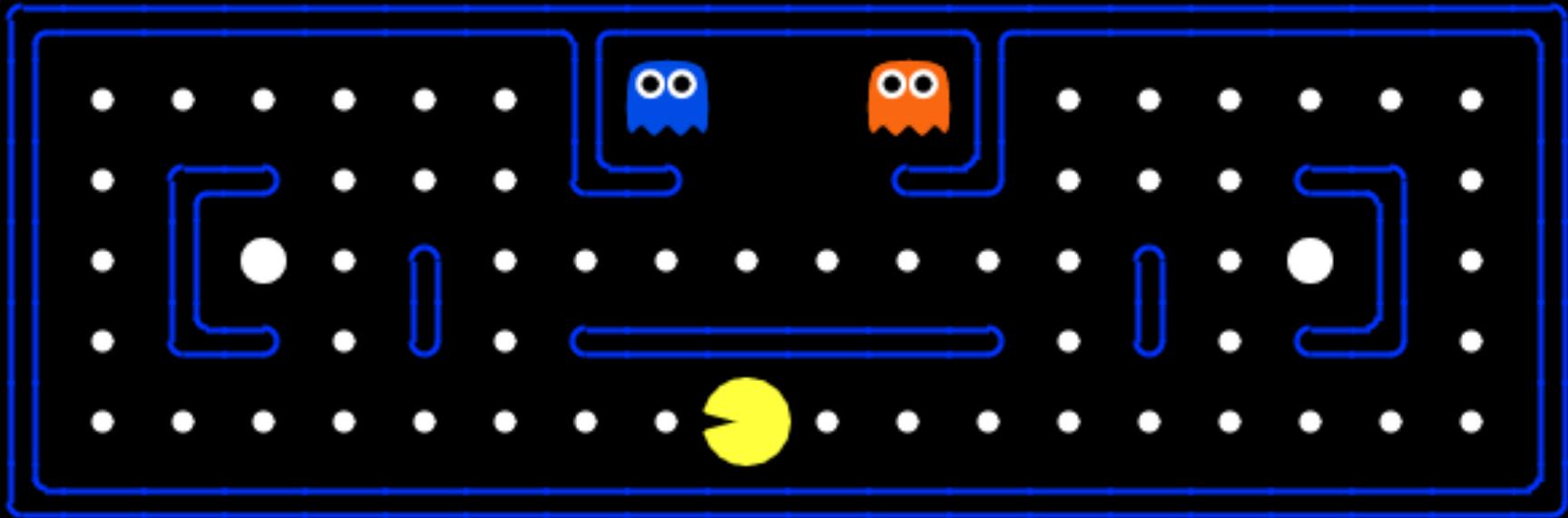
Continuu

Secvențial

Dinamic

Multiagent cooperativ

# PAC MAN

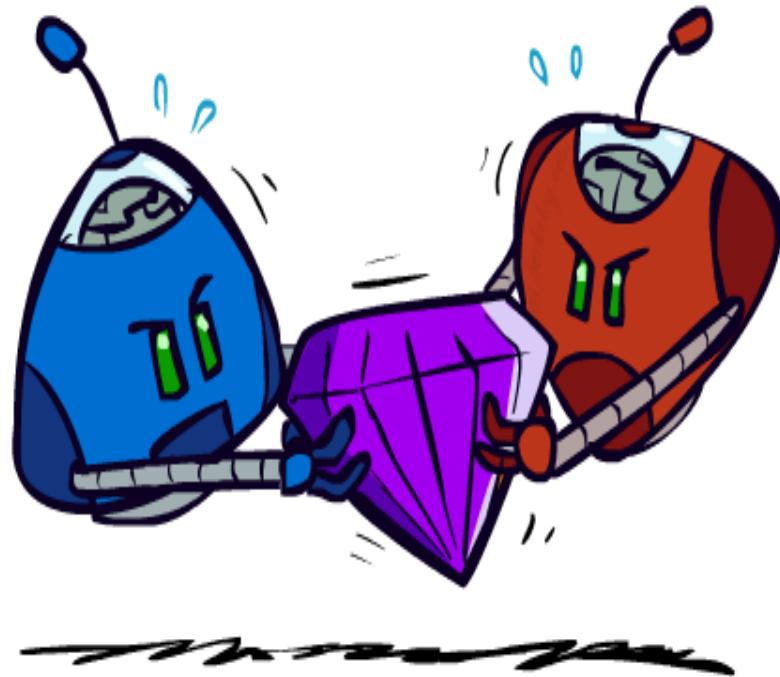


**SCORE:** 0

Complet observabil vs parțial observabil  
Determinist vs stochastic  
Discret vs continuu  
Episodic vs sevențial  
Static vs dinamic  
Agent vs multiagent

Complet observabil  
Stochastic  
Discret  
Sevențial  
Dinamic  
Multiagent competitiv

# Jocuri adversariale



# Jocuri

Medii multi-agent:

- fiecare agent trebuie să considere acțiunile celorlalți agenți
- adversar imprevizibil, incontrolabil
- mediu competitiv

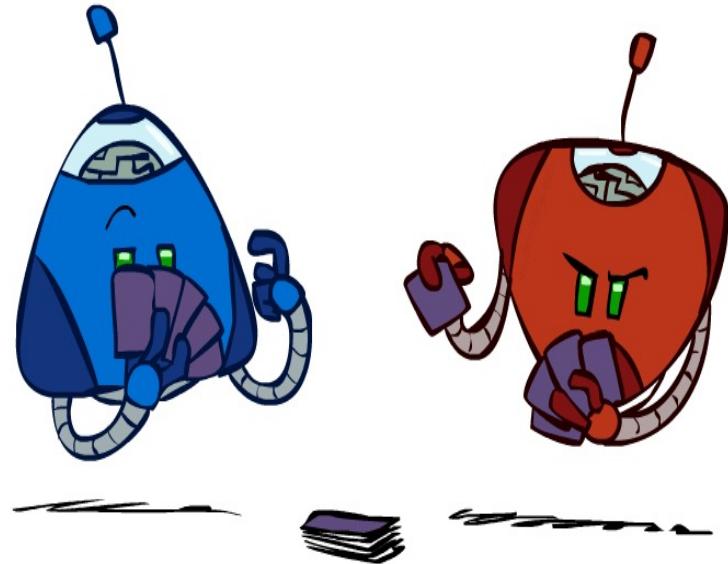
→ Căutare adversarială = joc

Soluția este o strategie (policy) care specifică mutarea pentru fiecare mutare posibilă a adversarului

Probleme de căutare adversarială sunt diferite de probleme de căutare, unde soluția este o secvență de acțiuni care garantează obținerea scopului.

# Tipuri de jocuri

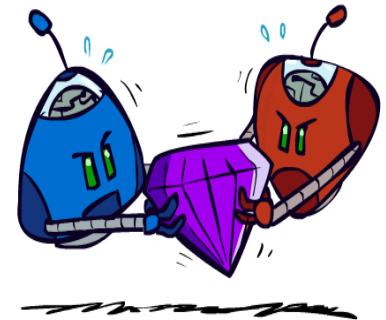
- multe tipuri de jocuri!
- dimensiuni:
  - deterministic sau stocastic?
  - unul, doi, sau mai mulți jucători?
  - joc de sumă zero?
  - complet sau parțial observabil? (se cunoaște întreaga stare)
- vrem să găsim algoritmi pentru a calcula o strategie ce recomandă o mutare din fiecare stare



# Tipuri de jocuri

	determinist	stochastic
Complet observabil	sah, dame, go	table, monopoly
Parțial observabil	avioane	poker, catan

- dimensiuni:
  - deterministic sau stochastic?
  - unul, doi, sau mai mulți jucători?
  - joc de sumă zero?
  - complet sau parțial observabil? (se cunoaște întreaga stare)
- vrem să găsim algoritmi pentru a calcula o strategie ce recomandă o mutare din fiecare stare



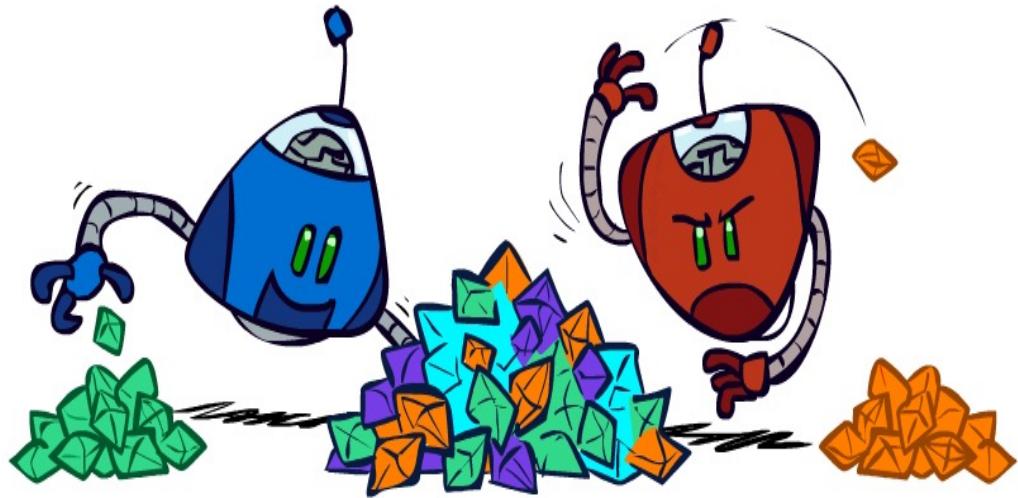
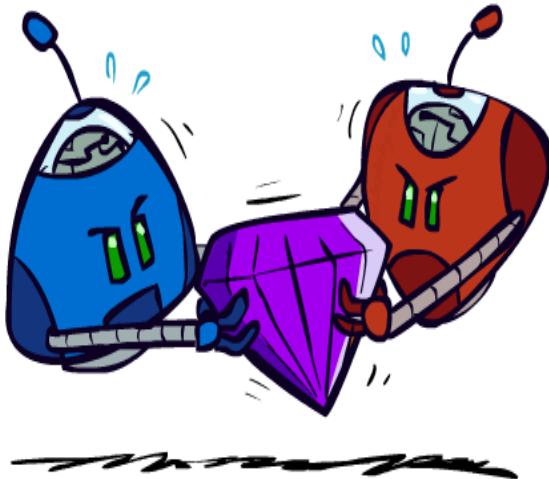
# Tipuri de jocuri

	determinist	stochastic
Complet observabil	sah, dame, go	table, monopoly
Partial observabil	avioane	poker, catan

- dimensiuni:
  - deterministic sau stochastic?
  - unul, doi, sau mai mulți jucători?
  - joc de sumă zero?
  - complet sau parțial observabil? (se cunoaște întreaga stare)
- vrem să găsim algoritmi pentru a calcula o strategie ce recomandă o mutare din fiecare stare



# Jocuri cu sumă zero



- Jocuri cu sumă zero
  - agenții au utilități (evaluari ale rezultatelor) opuse
  - un agent dorește maximizarea utilității (agentul MAX), celălalt minimizarea (agentul MIN)
  - mediu adversarial pur competitiv
- Jocuri generale
  - agenții au utilități independente
  - cooperare, indiferență, competiție

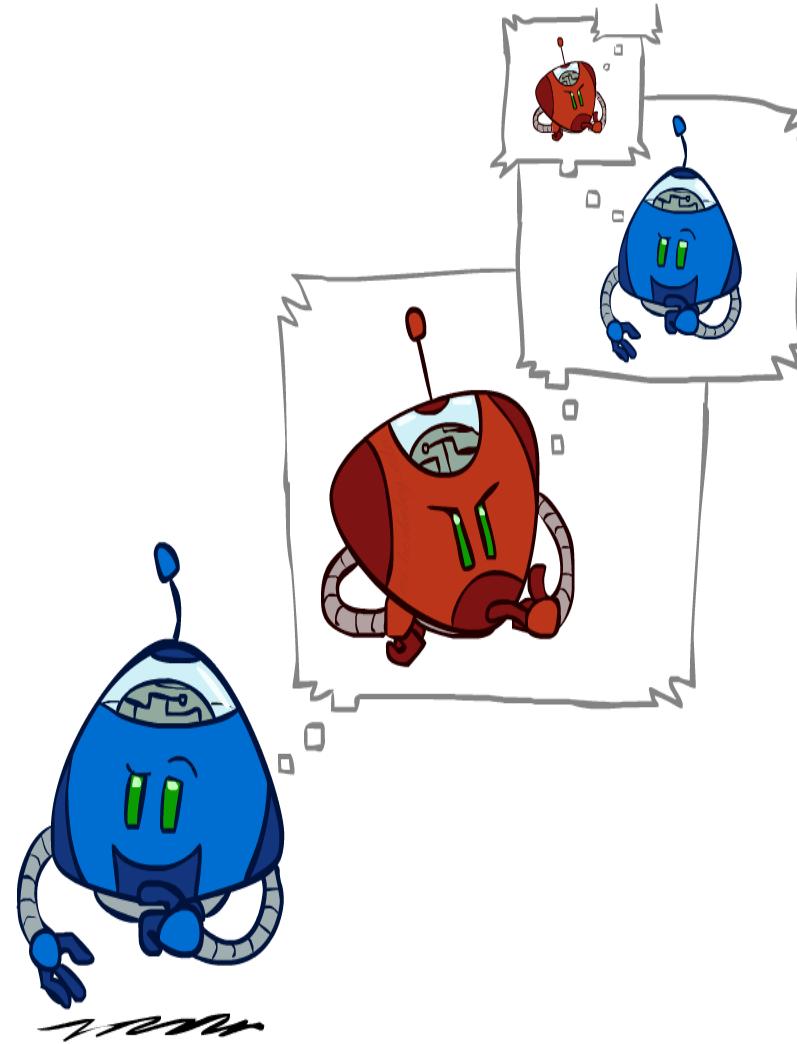
# O posibilă formulare a problemei

Jocuri cu doi jucători (MAX și MIN); la finalul jocului punctele sunt ale câștigătorului, penalitățile sunt ale pierzătorului (joc de sumă zero)

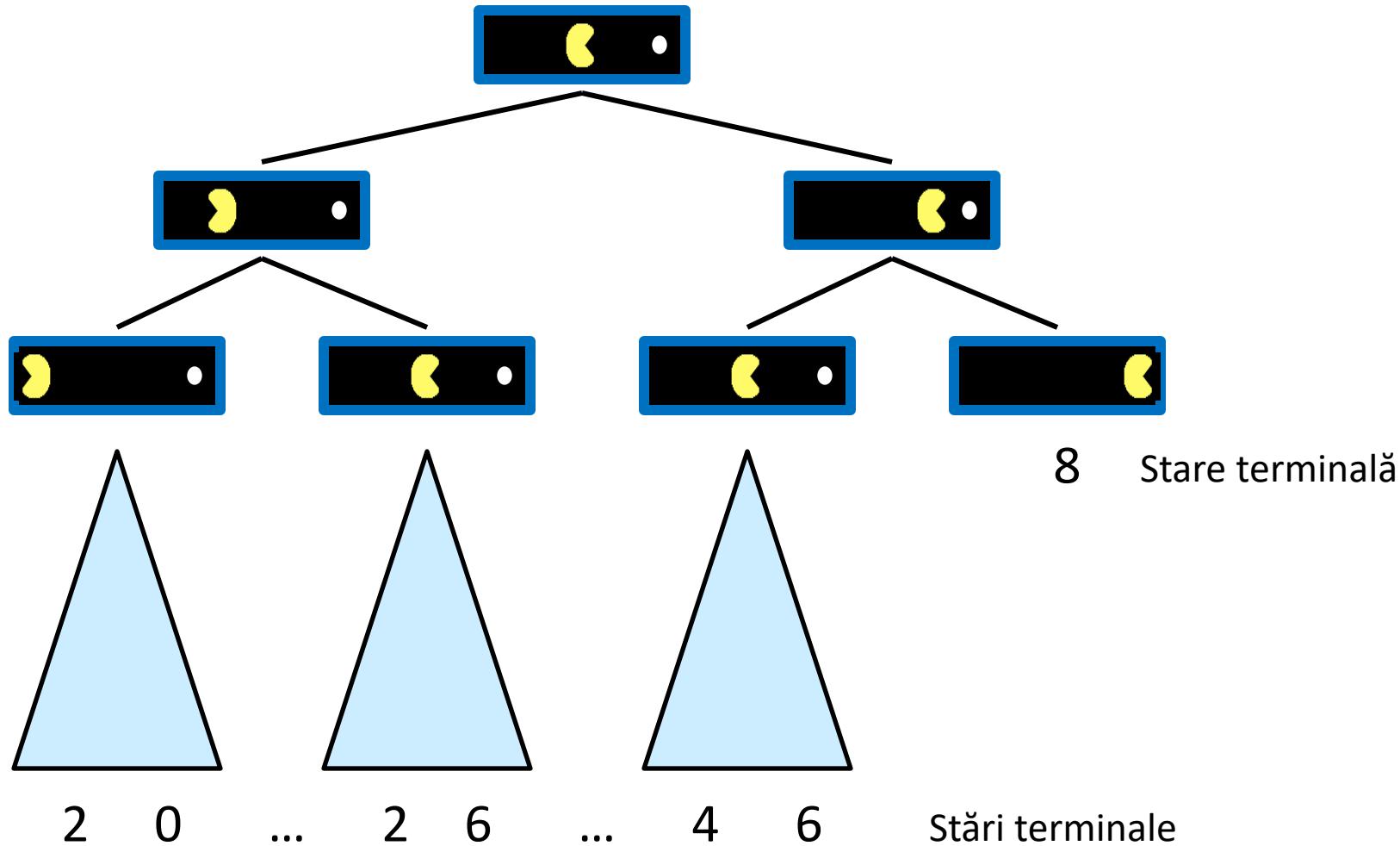
- stare initială:  $s_0$
- jucătorul care trebuie să mute într-o stare:  $\text{PLAYER}(s)$
- acțiuni (mutări) legale într-o stare:  $\text{ACTIONS}(s)$
- modelul de tranziție (funcția succesor):  $\text{RESULT}(s,a)$
- test terminal :  $\text{TERMINAL-TEST}(s)$
- o funcție de utilitate – asociază o valoare numerică stărilor terminale:  $\text{UTILITY}(s)$

Şah: funcția de utilitate: 0 – remiză, +1 câștig, -1 pierdere.  
Există și jocuri cu funcție de utilitate diferită.

# Găsirea unei strategii optime



# Arbore de căutare într-un mediu cu un singur agent

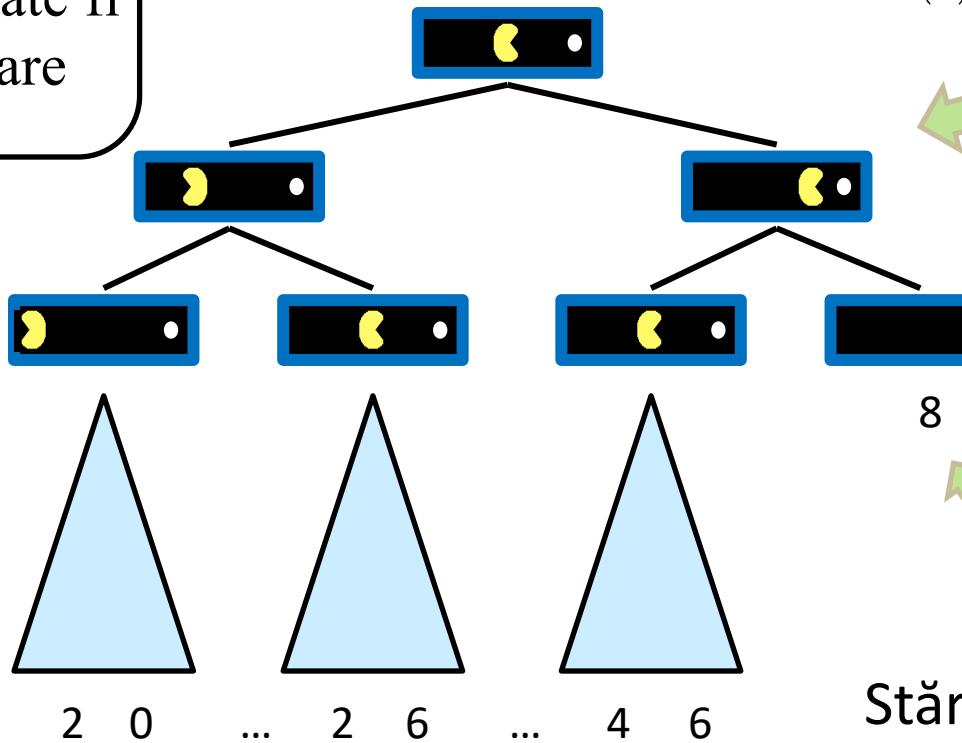


Fiecare mutare: -1 punct

Mânânc punctul alb: + 10 puncte

# Valoarea unei stări

Valoarea unei stări:  
cea mai bună utilitate  
(rezultat) care poate fi  
atinsă din acea stare



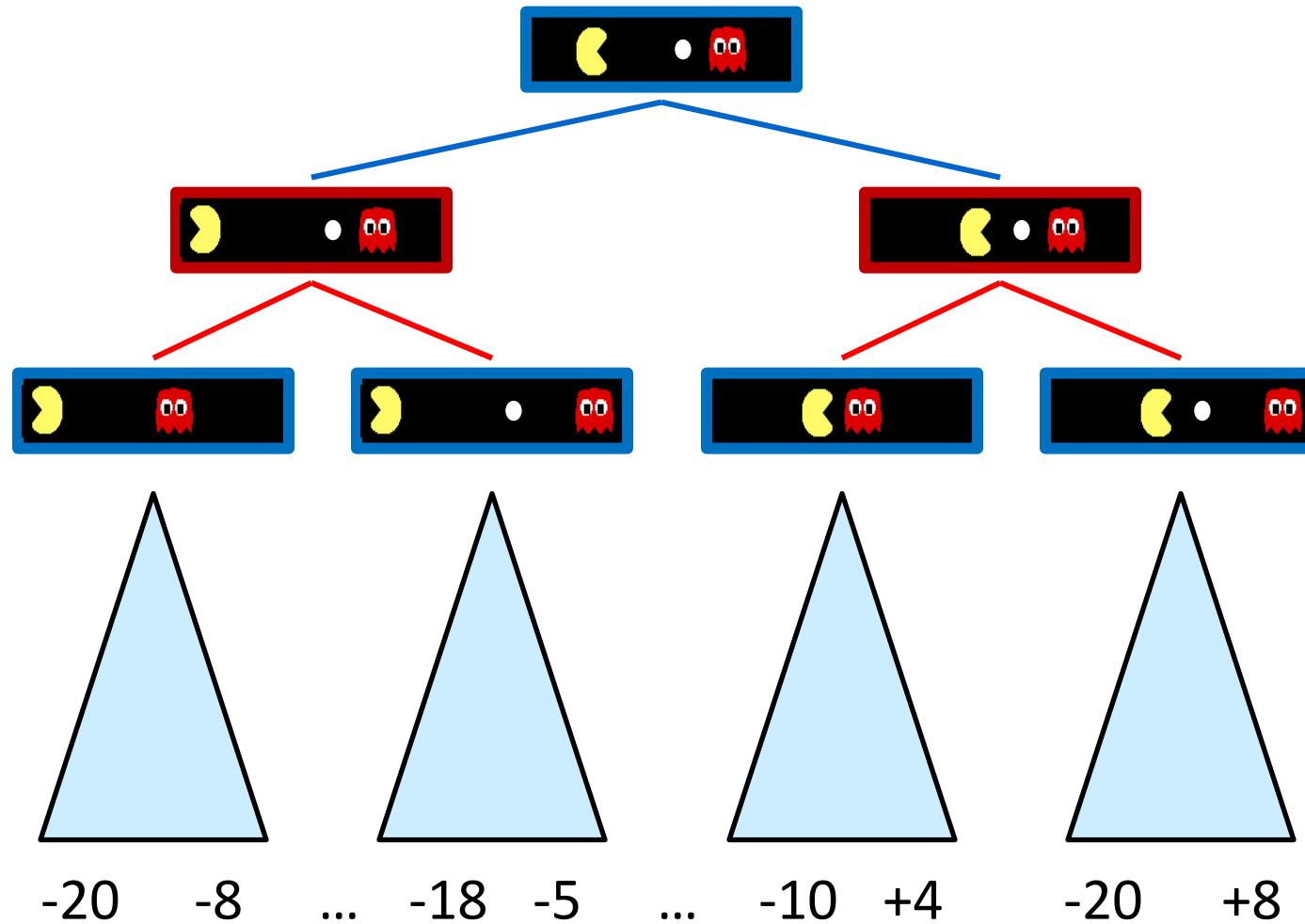
Stări neterminale:

$$V(s) = \max_{s' \in \text{children}(s)} V(s')$$

Stări terminale:

$V(s)$  = cunoscută, dată  
de regulile jocului

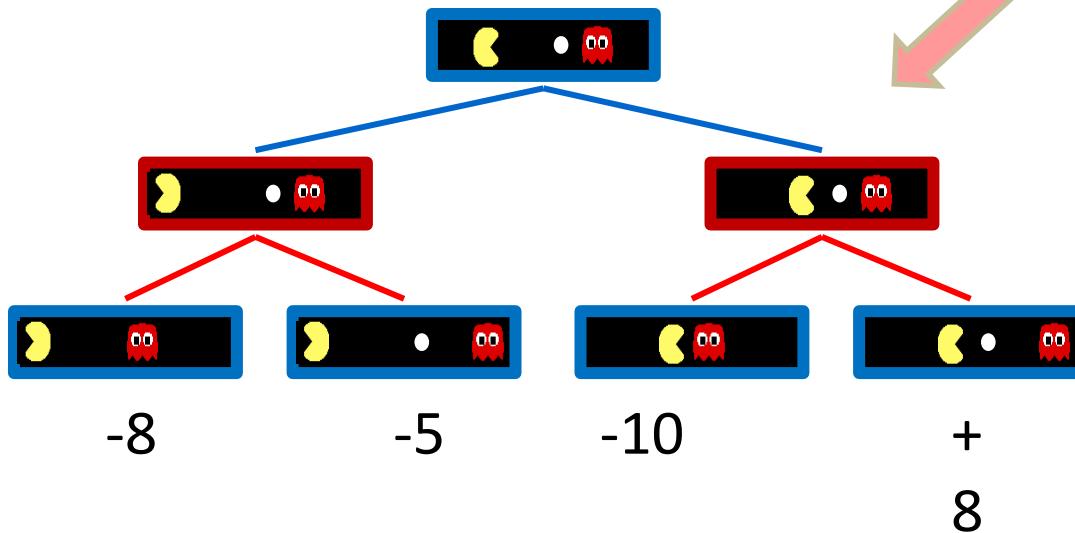
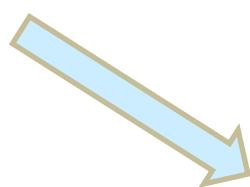
# Arbore de joc în medii cu doi agenti, deterministic, mutări pe rând



# Valoare MiniMax a unei stări

Stări controlate de MAX

$$V(s) = \max_{s' \in \text{successors}(s)} V(s')$$



Stări controlate de MIN:

$$V(s') = \min_{s \in \text{successors}(s')} V(s)$$

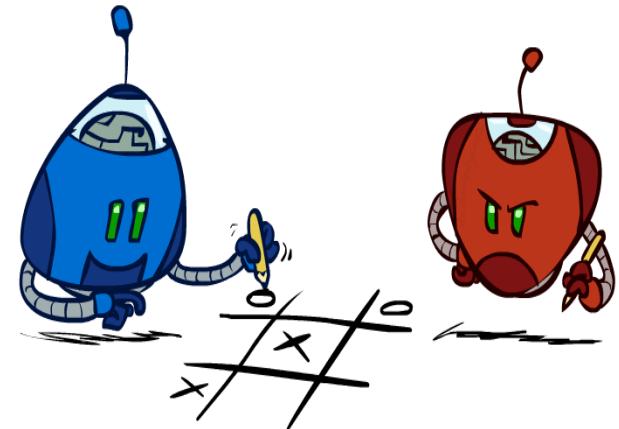
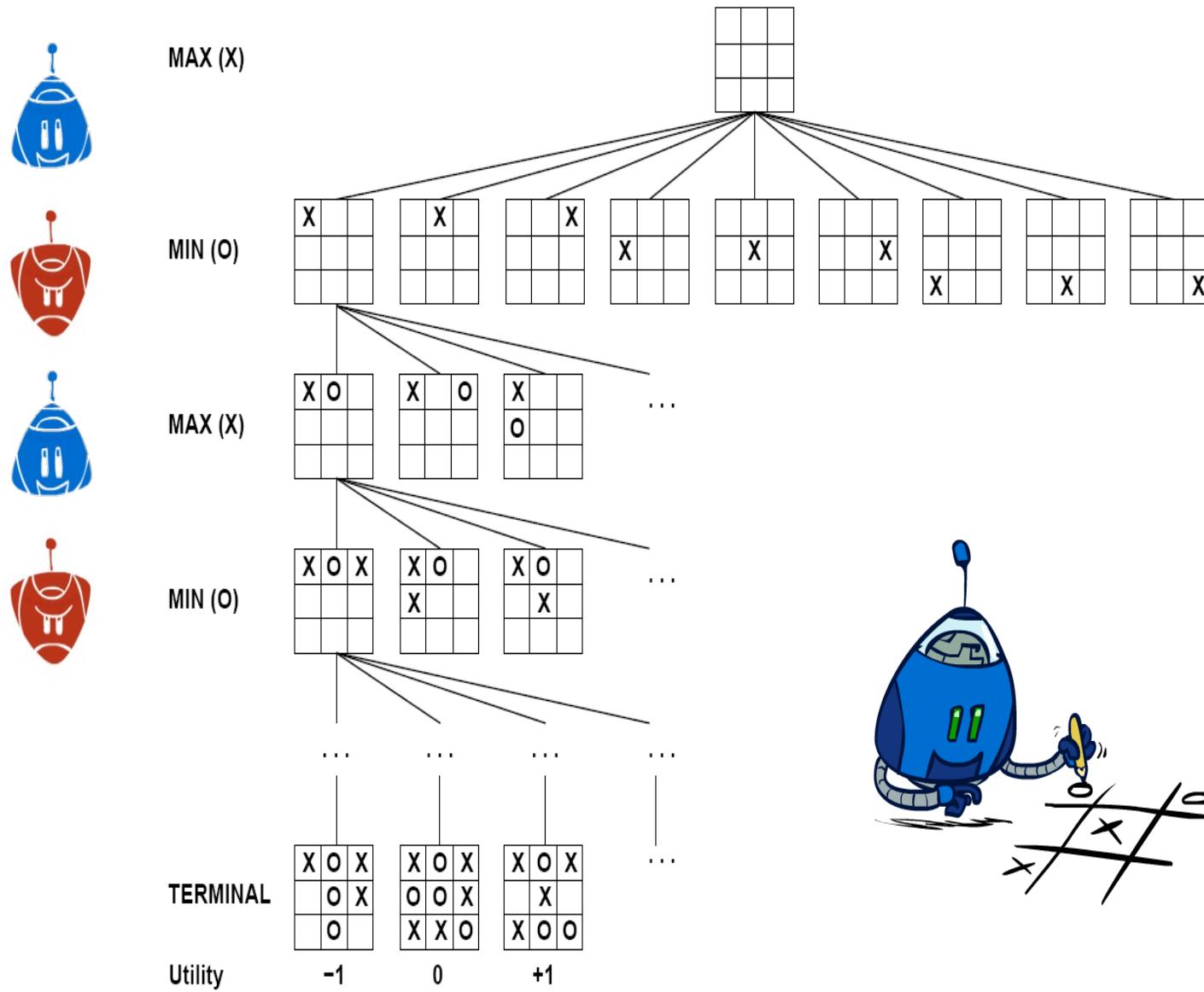


Stări terminale:  $V(s) = \text{cunoscută}$ ,  
dată de regulile jocului

# Arbore de joc

- nodurile corespund situațiilor (stărilor)
- situația inițială a jocului dată de nodul rădăcină
- arcele corespund mutărilor
- frunzele corespund situațiilor terminale (s-a terminat jocul)
- 2 jucători: MAX (de obicei este cel care face prima mutare) și MIN

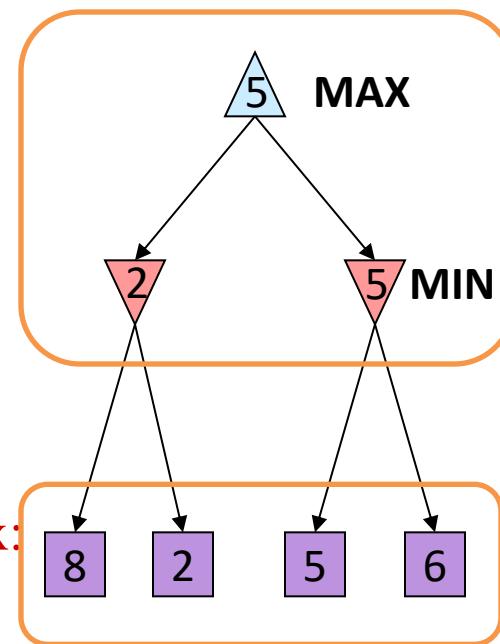
# Arbore de joc pentru X și 0



# Algoritmul MiniMax pentru căutare adversarială

- Jocuri de sumă zero cu doi jucători, deterministe, cu informație perfectă
  - X și 0, săh, dame
  - un jucător (MAX) maximizează utilitatea
  - celălalt jucător (MIN) minimizează utilitatea
- Căutare MiniMax:
  - arbore de căutare în spațiul stărilor
  - jucătorii mută pe rând
  - calculăm pentru fiecare nod valoarea **minimax**: cea mai bună utilitate care poate fi obținută împotriva unui adversar care joacă perfect
  - determină strategia optimă corespunzătoare lui MAX, care este cea mai bună primă mutare

**Valori minimax:**  
calculate recursiv



**Valori terminale:**  
date de regulile jocului

# Algoritmul MiniMax - descriere

1. Generează întreg arborele de joc până la stările terminale
2. Aplică funcția de utilitate fiecărei stării terminale - obține valoarea stării
3. Deplasează-te înapoi în arbore, de la nodurile-frunze spre nodul-rădăcină, determinând, corespunzător fiecărui nivel al arborelui, valorile care reprezintă utilitatea nodurilor aflate la acel nivel. Propagarea acestor valori la niveluri anterioare se face prin intermediul nodurilor-părinte succesive, conform următoarei reguli:
  - dacă starea-părinte este un nod de tip MAX, atribuie-i maximul dintre valorile avute de fii săi;
  - dacă starea-părinte este un nod de tip MIN, atribuie-i minimul dintre valorile avute de fii săi;
4. Ajuns în nodul-rădăcină, alege pentru MAX acea mutare care conduce la valoarea maximă. Mutarea se numește ***decizia minimax*** - maximizează utilitatea, în ipoteza că oponentul joacă perfect cu scopul de a o minimiza.

# Algoritmul MiniMax - formalizare

- jucătorul care trebuie să mute într-o stare:  $\text{PLAYER}(s)$
- acțiuni (mutări) legale într-o stare:  $\text{ACTIONS}(s)$
- modelul de tranziție (funcția succesor):  $\text{RESULT}(s, a)$
- test terminal :  $\text{TERMINAL-TEST}(s)$
- o funcție de utilitate – asociază o valoare numerică stărilor terminale:  $\text{UTILITY}(s)$

$\text{MINIMAX}(s) =$

$$\begin{cases} \text{UTILITY}(s) & \text{if } \text{TERMINAL-TEST}(s) \\ \max_{a \in \text{Actions}(s)} \text{MINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{PLAYER}(s) = \text{MAX} \\ \min_{a \in \text{Actions}(s)} \text{MINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{PLAYER}(s) = \text{MIN} \end{cases}$$

# Implementare MiniMax

```
def value(stare):
```

dacă stare este o stare terminală: returnează utilitatea stării

dacă următorul agent care mută este **MAX**:  
returnează **max-value(stare)**

dacă următorul agent care mută este **MIN**: returnează **min-value(stare)**

```
def max-value(stare):
```

initializează  $v = -\infty$

pentru fiecare successor al stării:

$v = \max(v, \text{value(successor)})$

returnează  $v$

```
def min-value(state):
```

initializează  $v = +\infty$

pentru fiecare successor al stării:

$v = \min(v, \text{value(successor)})$

returnează  $v$

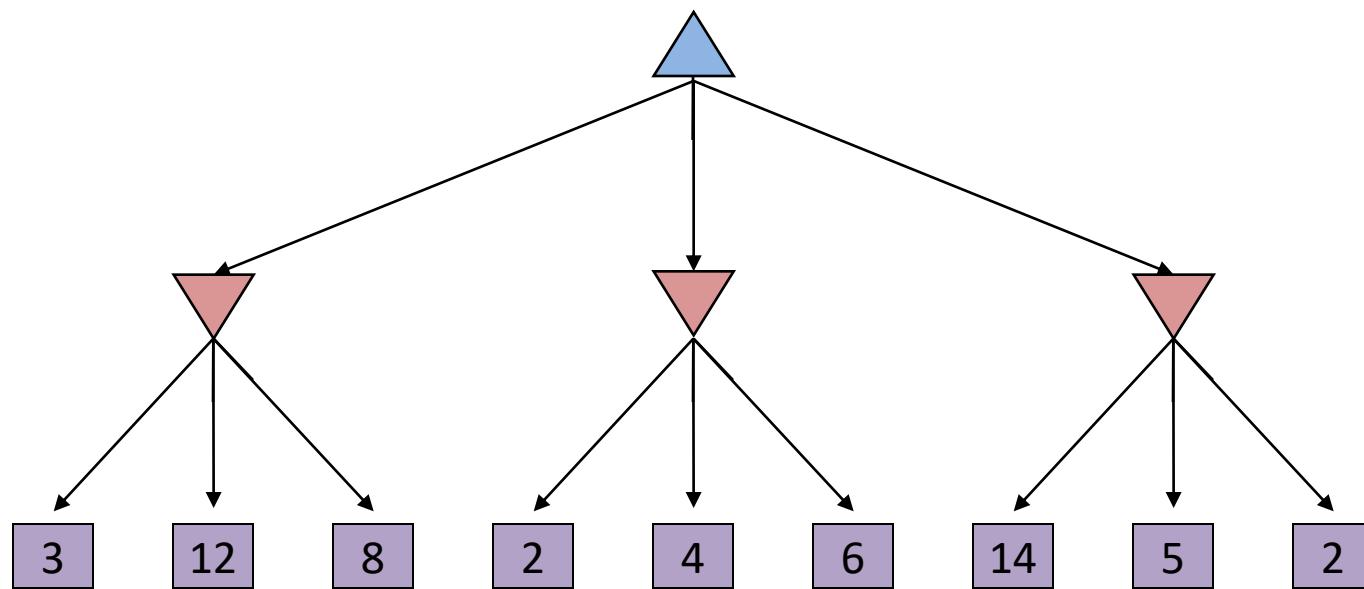
$$V(s) = \max_{s' \in \text{successors}(s)} V(s')$$

$$V(s') = \min_{s \in \text{successors}(s')} V(s)$$

# Exemplul MiniMax

**MAX**

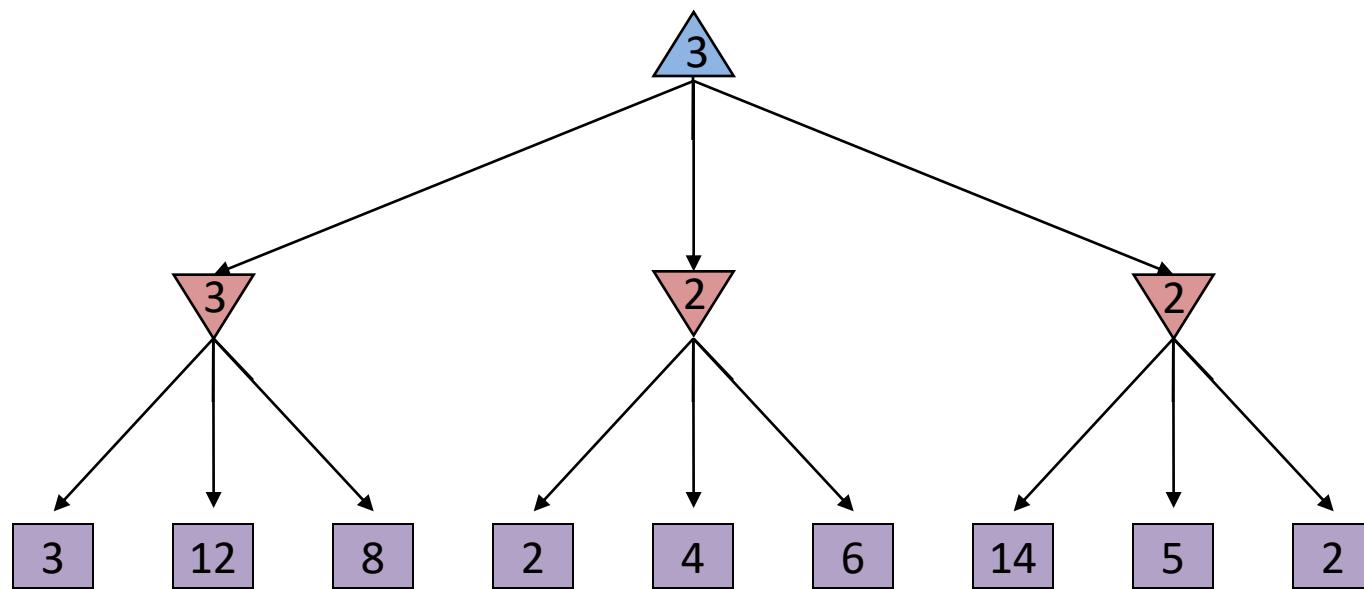
**MIN**



# Exemplul MiniMax

**MAX**

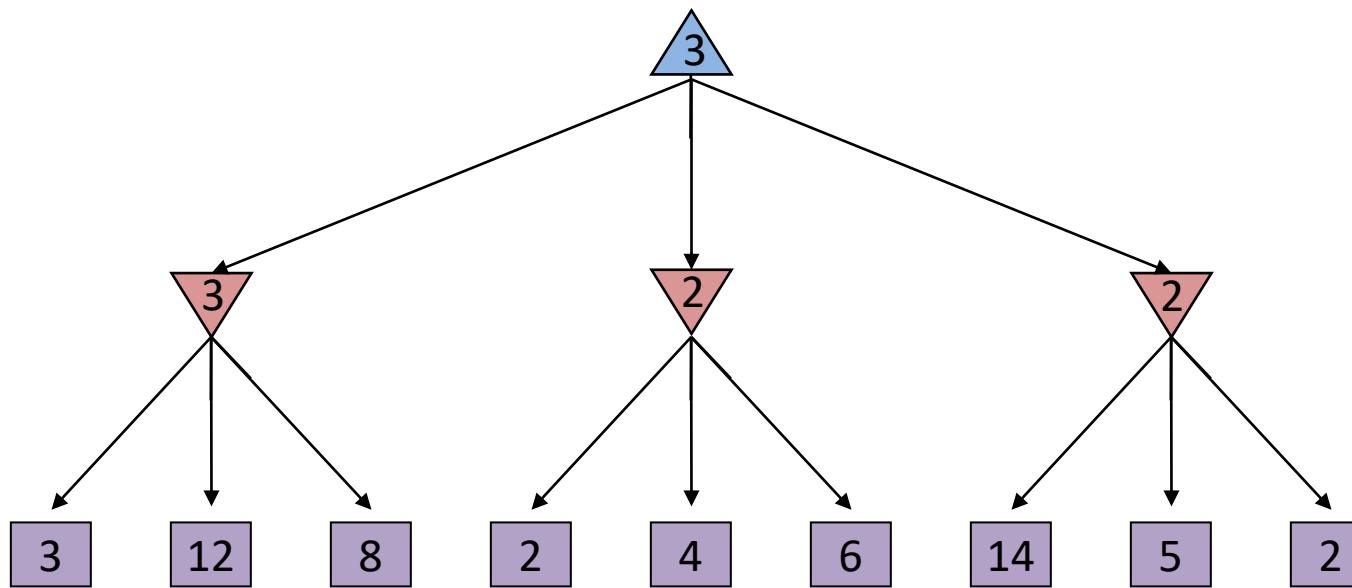
**MIN**



# Exemplul MiniMax

MAX

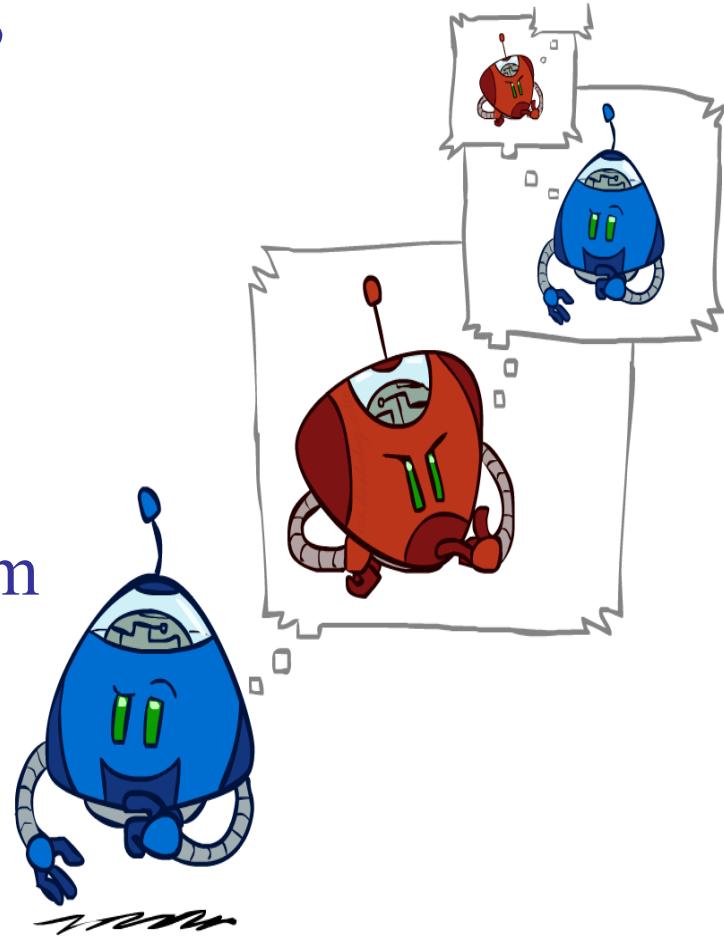
MIN



- valorile pozițiilor de la ultimul nivel sunt determinate de către funcția de utilitate și se numesc valori statice (se realizează evaluări statice).
- valorile minimax ale nodurilor interne sunt calculate în mod dinamic, în manieră bottom-up, nivel cu nivel, până când este atins nodul-rădăcină.
- valoarea rezultată este 3 și prin urmare cea mai bună mutare a lui MAX din poziția curentă este mutarea la stânga. Cel mai bun răspuns al lui MIN este mutarea la stânga. Această secvență a jocului poartă denumirea de variație principală. Ea definește jocul optim de tip minimax pentru ambele părți.
- se observă ca valoarea pozițiilor de-a lungul variației principale nu variază. Prin urmare, **mutările corecte sunt cele care conservă valoarea jocului**.

# Proprietăți ale algoritmului MiniMax

- Cât de eficient este MiniMax?
  - Căutare exhaustivă în manieră depth-first
  - timp:  $O(b^m)$
  - spațiu:  $O(bm)$
- Exemple: pentru șah,  $b \approx 35$ ,  $m \approx 100$ 
  - o soluție exactă este imposibilă
  - dar, oare chiar trebuie să explorăm întreg arborele?

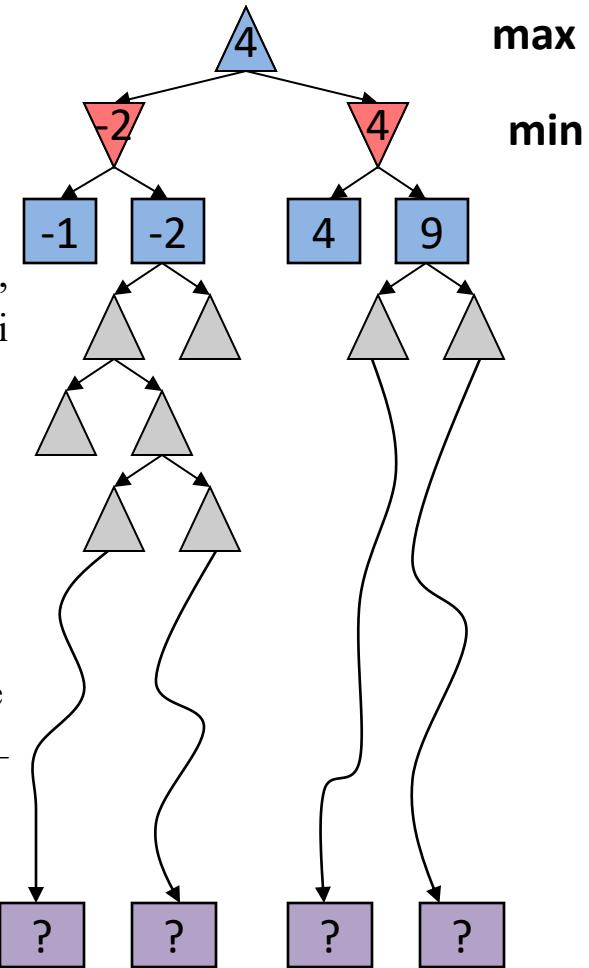


# Limitări date de resurse ale algoritmului MiniMax

1. Generează întreg arborele de joc până la stările terminale
2. Aplică funcția de utilitate fiecărei stării terminale - obține valoarea stării
3. Deplasează-te înapoi în arbore, de la nodurile-frunze spre nodul-rădăcină, determinând, corespunzător fiecărui nivel al arborelui, valorile care reprezintă utilitatea nodurilor aflate la acel nivel. Propagarea acestor valori la niveluri anterioare se face prin intermediul nodurilor-părinte succesive, conform următoarei reguli:
  - dacă starea-părinte este un nod de tip MAX, atribuie-i maximul dintre valorile avute de fii săi;
  - dacă starea-părinte este un nod de tip MIN, atribuie-i minimul dintre valorile avute de fii săi;
4. Ajuns în nodul-rădăcină, alege pentru MAX acea mutare care conduce la valoarea maximă. Mutarea se numește ***decizia minimax*** - maximizează utilitatea, în ipoteza că oponentul joacă perfect cu scopul de a o minimiza.

# Limitări date de resurse

- problemă: în jocuri reale, nu putem genera tot arborele până la frunze din cauza resurselor limitate (timp + spațiu)
- soluție posibilă: căutare în adâncime limitată
  - căută până la o anumită adâncime în arborele de joc
  - înlocuiește utilitățile stărilor terminale (care sunt calculate exact, după regulile jocului) cu o funcție de evaluare a utilității unor stări neterminale (această funcție aproximează utilitatea)
- exemplu:
  - avem la dispoziție 100 de secunde pentru a realiza mutarea;
  - putem explora 10000 noduri / secundă;
  - în 100 de secunde putem explora 1 Milion de noduri pentru mutare
  - algoritmul  $\alpha$ - $\beta$  retezare (îl prezentăm azi) ajunge la adâncime 8 – performanță bună în șah;
- garanția de optimalitate dispare;
- cu cât pot analiza mai multe noduri cu atât mutarea este mai bună;
- uneori pot folosi și un algoritm de căutare incrementală în adâncime



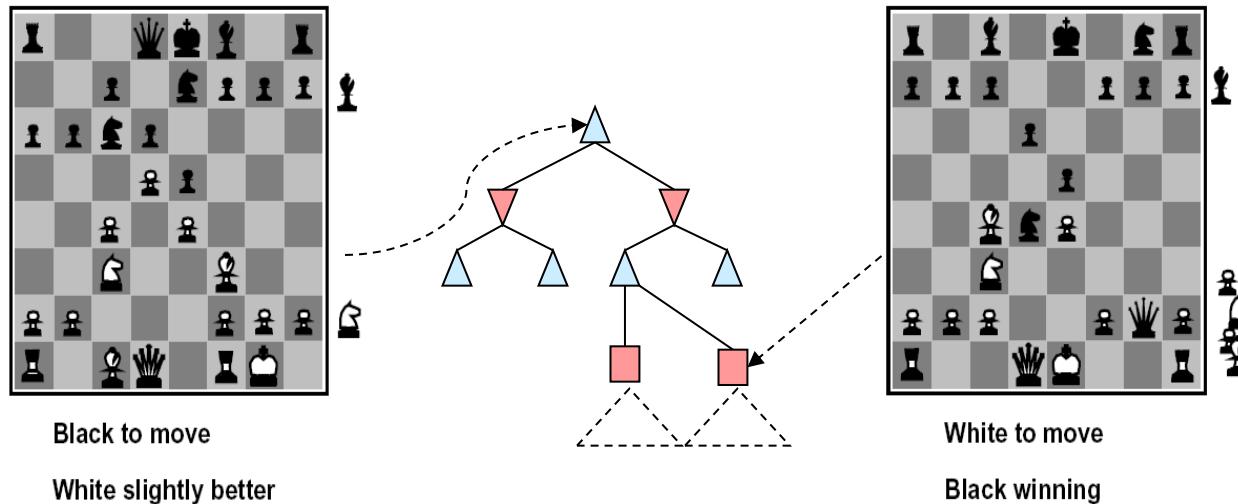
# Influența adâncimii pentru funcția de evaluare

- Funcțiile de evaluare sunt imperfecte
- Cu cât merg mai adâncime în arbore și apelez funcția de evaluare a unei stări neterminale mai târziu cu atât am sănse să greșesc mai puțin.



# Funcții de evaluare - șah

- Funcțiile de evaluare asociază un scor stărilor neterminale (este folosită de căutarea în adâncime limitată, căutarea în adâncime iterativă)



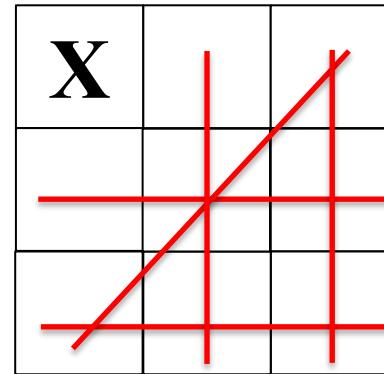
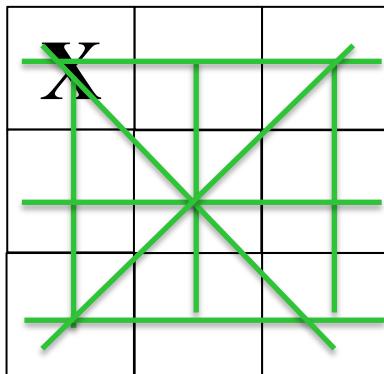
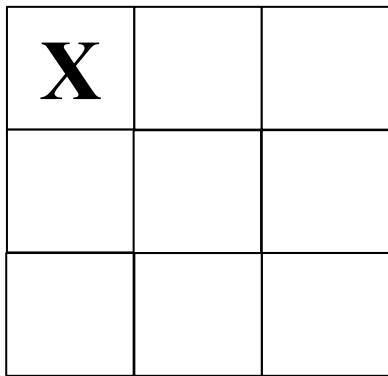
- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc: pentru șah se consideră funcții liniare în care ponderăm piesele de pe tablă

$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s)$$

- unde,  $f_1(s) = (\#\text{regine\_albe} - \#\text{regine\_negre})$ ,  $w_1 = 100$ , etc.

# Functii de evaluare – X și 0

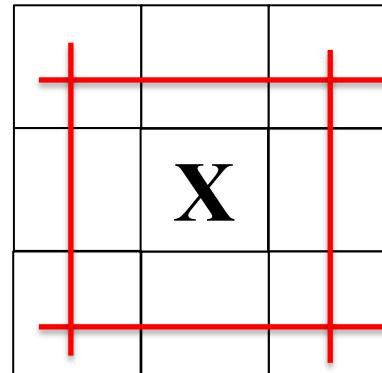
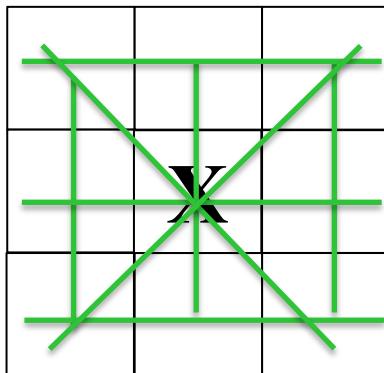
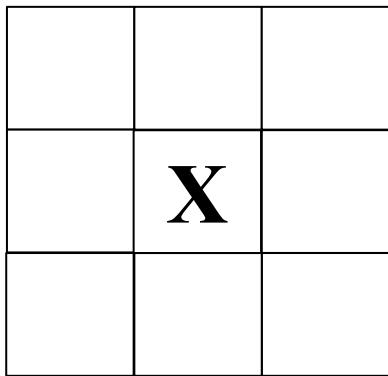
- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:



- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- în exemplul de mai sus:  $8 - 5 = 3$

# Functii de evaluare – X și 0

- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:

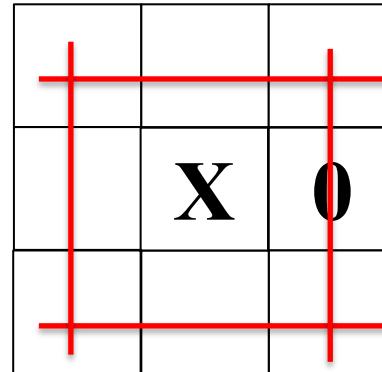
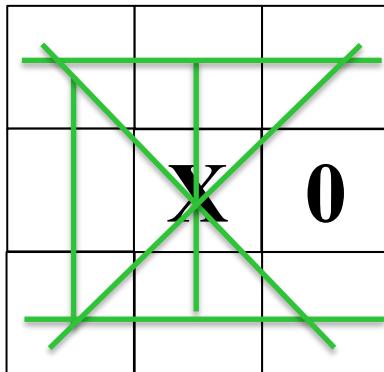


- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- În exemplul de mai sus:  $8-4 = 4$

# Functii de evaluare – X și 0

- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:

	X	0

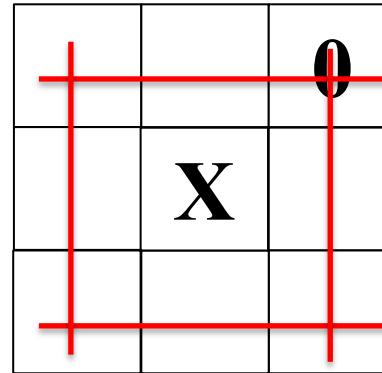
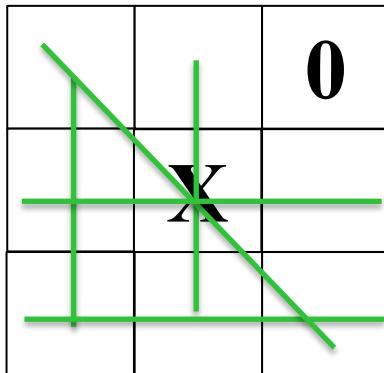


- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- În exemplul de mai sus:  $6-4 = 2$

# Functii de evaluare – X și 0

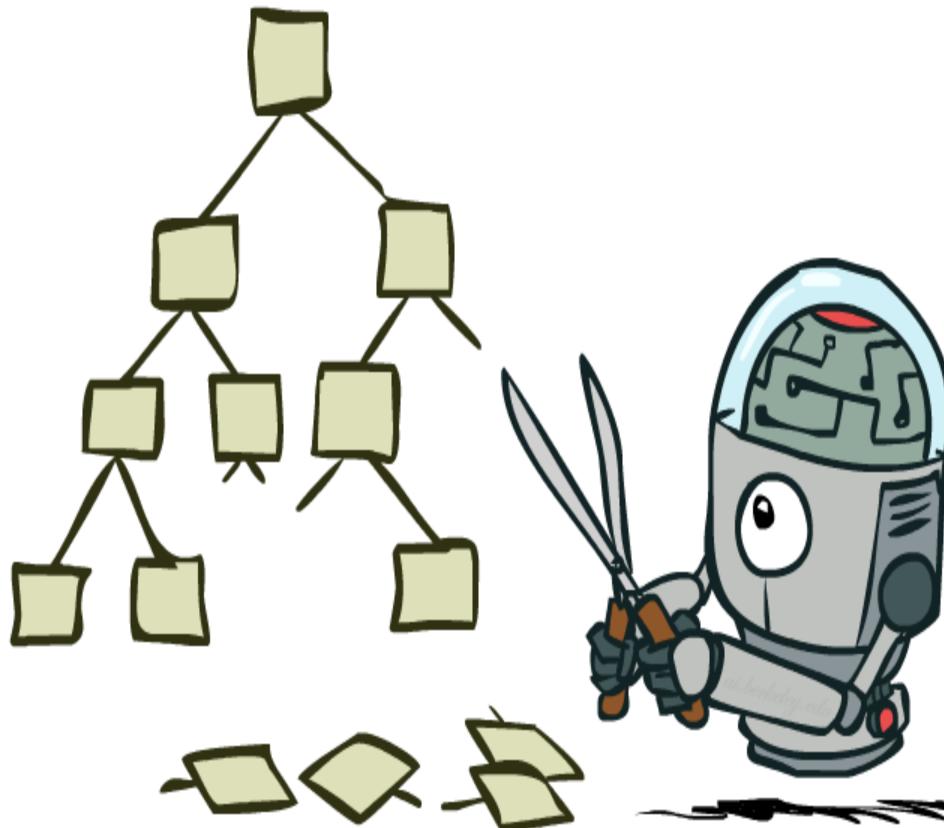
- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:

		0
	X	



- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- În exemplul de mai sus:  $5 - 4 = 1$

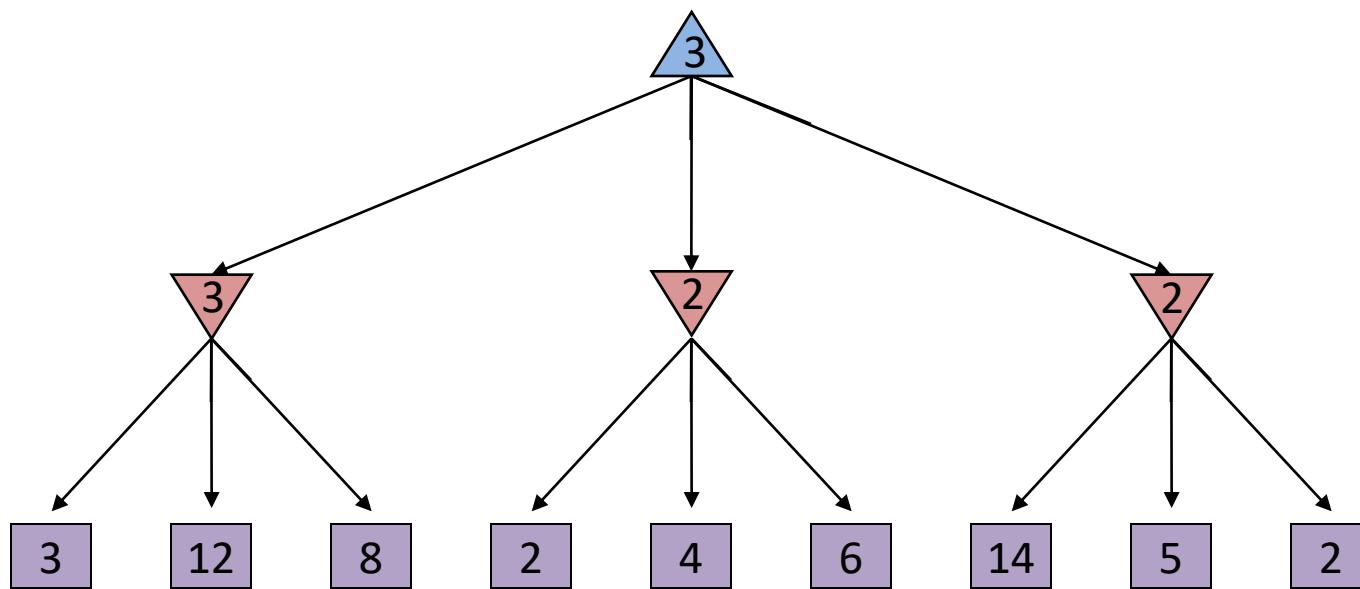
# Retezări (Pruning) în arborele de joc



# Exemplu MiniMax

**MAX**

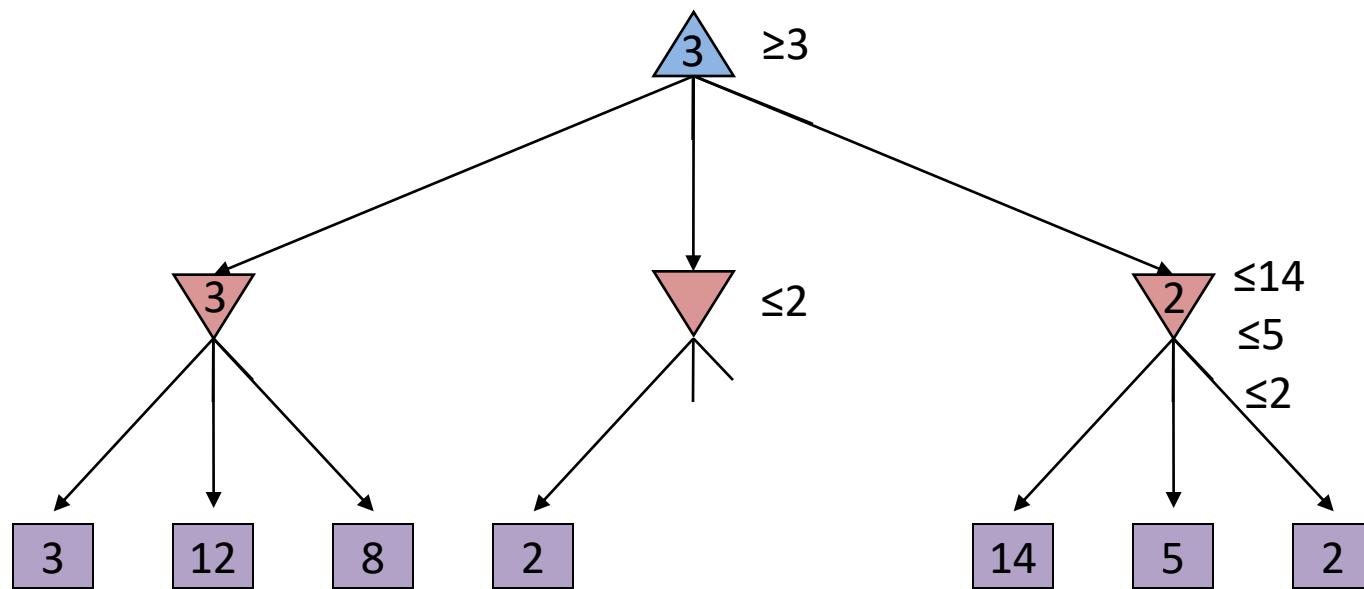
**MIN**



# Accelerarea algoritmului MiniMax

**MAX**

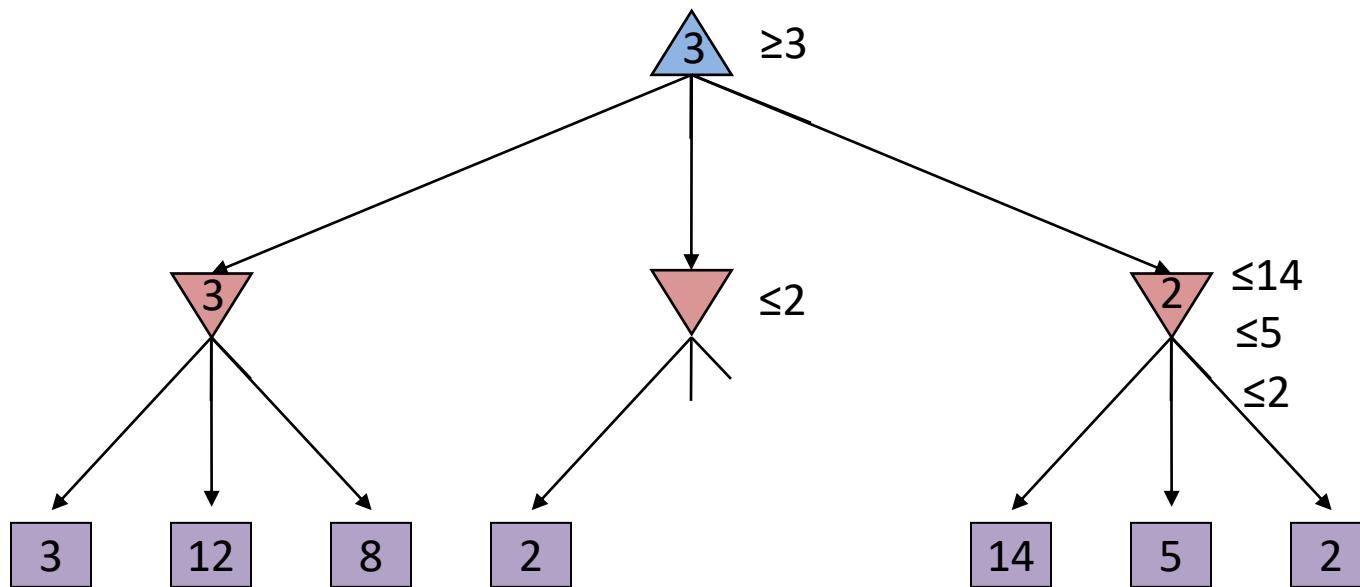
**MIN**



# Retezare alfa-beta

MAX

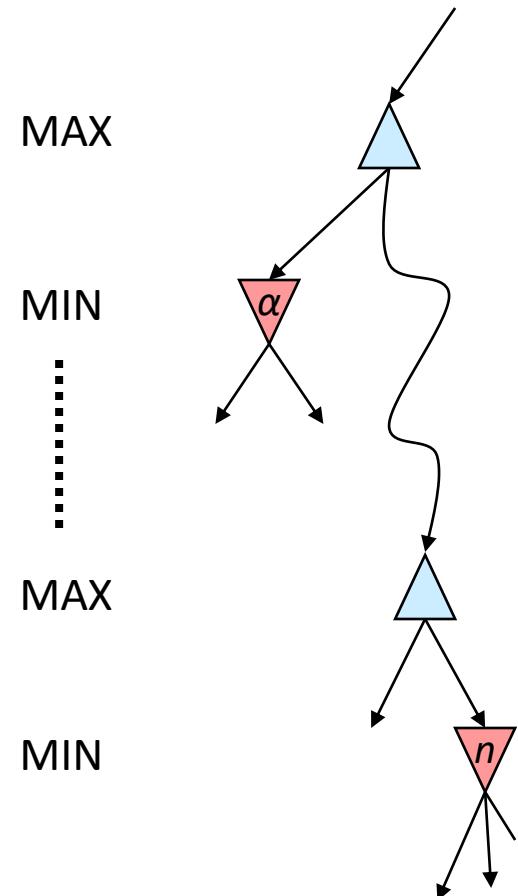
MIN



Tehnica de alfa-beta retezare, când este aplicată unui arbore de tip Minimax standard, va întoarce aceeași mutare pe care ar furniza-o și Algoritmul MiniMax, dar într-un timp mai scurt, întrucât realizează o retezare a unor ramuri (subarbore) ale arborelui care nu pot influența decizia finală și care nu mai sunt vizitate.

# Retezare alfa-beta (alfa-beta pruning)

- Cazurile când se aplică (pentru noduri de tip MIN)
  - calculăm pentru jucătorul MIN valoarea nodului  $n$
  - iterăm după toți succesorii nodului  $n$  de la stânga la dreapta
  - valoarea estimată a nodului  $n$  va scădea pe măsură ce îi vizităm toți succesorii
  - cine este interesat de valoarea nodului  $n$ ? MAX
  - fie  $\alpha$  cea mai bună valoare curentă (garantată) pe care MAX o poate obține după ce a vizitat subarborele din stânga subarborelui actual vizitat pornind de la rădăcină
  - dacă valoarea curentă estimată a lui  $n$  devine mai mică decât  $\alpha$ , MAX nu va alege acest drum, deci putem să renunțăm (să retezăm) să evaluăm valorile pentru celelalte noduri succesor ale lui  $n$ .
- Cazurile când se aplică (pentru noduri de tip MAX)
  - simetric, se definește în mod similar valoarea  $\beta$  ca fiind cea mai bună valoare (garantată) pe care MIN o poate obține.
  - MIN nu mai trebuie să mai ia în considerare nicio valoare internă mai mare sau egală cu  $\beta$  care este asociată oricărui nod intern  $n$  de tip MAX. Putem tăia (reteza) acea ramură a arborelui de joc.



# Implementare Alfa-Beta retezare

$\alpha$ : opțiunea curentă cea mai bună a lui MAX (MAX vrea să maximizeze  $\alpha$ )

$\beta$ : opțiunea curentă cea mai bună a lui MIN (MIN vrea să minimizeze  $\beta$ )

Întotdeauna vom avea  $\alpha \leq \beta$

```
def max-value(stare, α, β): //nodul n e de tip MAX, MAX are α, MIN are β
    initializeaza v = -∞
    pentru fiecare succesor al stării: //pentru fiecare succesor al lui n
        v = max(v, value(successor, α, β)) //calculăm valoarea v și actualizăm max
        if v ≥ β return v //dacă v depășește limita β a lui MIN nu mai continuăm
                           //întoarcem în acest caz v = o estimare a lui n
        α = max(α, v) //actualizăm valoarea α
    return v
```

# Implementare Alfa-Beta retezare

$\alpha$ : opțiunea curentă cea mai bună a lui MAX (MAX vrea să maximizeze  $\alpha$ )

$\beta$ : opțiunea curentă cea mai bună a lui MIN (MIN vrea să minimizeze  $\beta$ )

Întotdeauna vom avea  $\alpha \leq \beta$

```
def max-value(stare,  $\alpha$ ,  $\beta$ ):
```

    initializeaza  $v = -\infty$

    pentru fiecare succesor al stării:

$v = \max(v, \text{value}(\text{successor}, \alpha, \beta))$

        if  $v \geq \beta$  return  $v$

$\alpha = \max(\alpha, v)$

    return  $v$

```
def min-value(stare,  $\alpha$ ,  $\beta$ ):
```

    initializeaza  $v = +\infty$

    pentru fiecare succesor al stării:

$v = \min(v, \text{value}(\text{successor}, \alpha, \beta))$

        if  $v \leq \alpha$  return  $v$

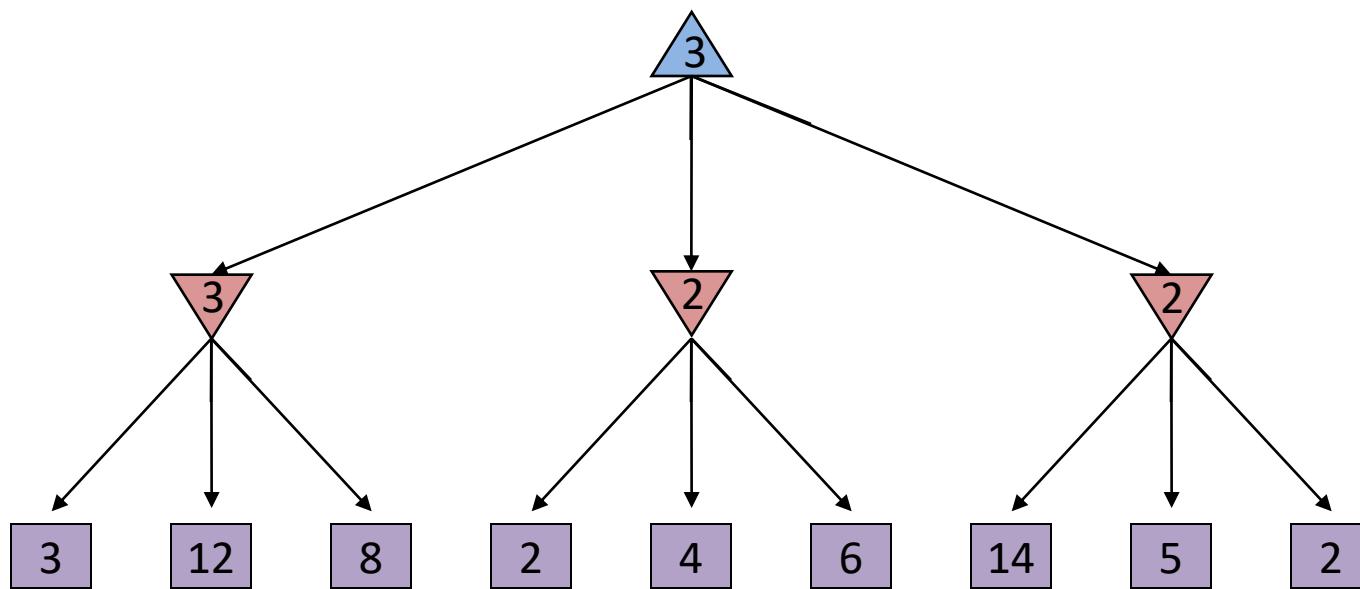
$\beta = \min(\beta, v)$

    return  $v$

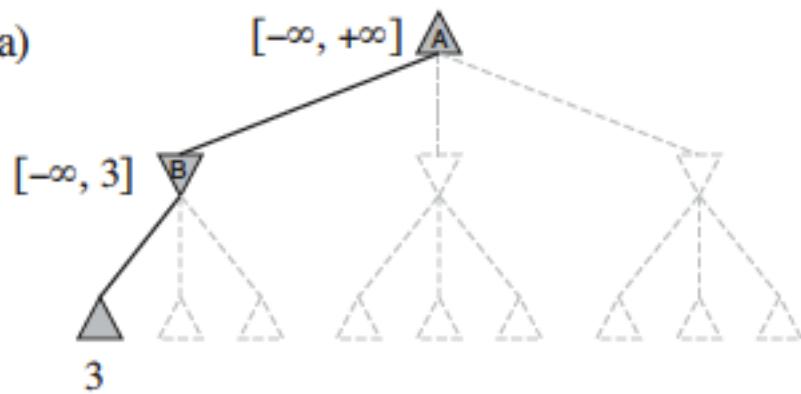
# Exemplu MiniMax

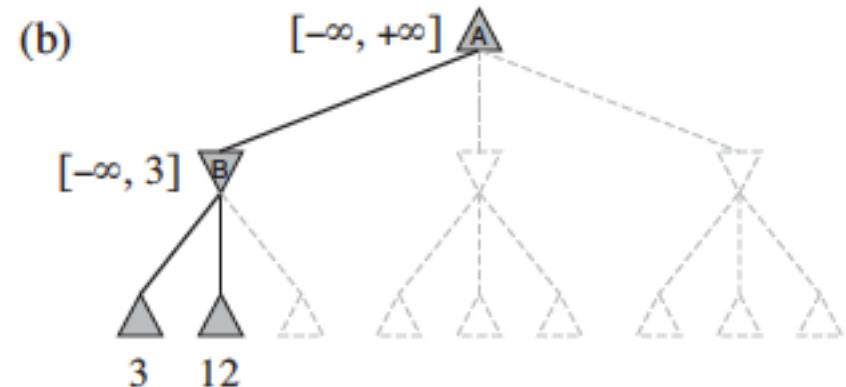
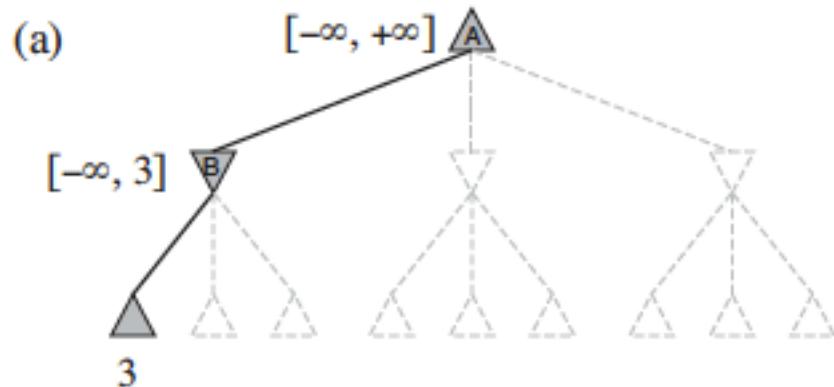
**MAX**

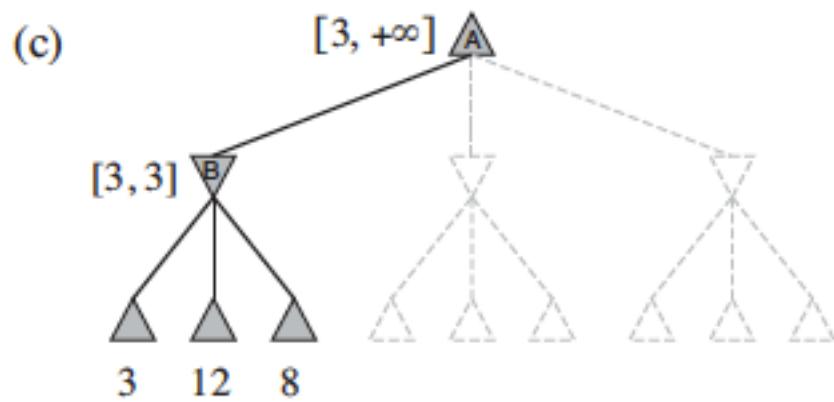
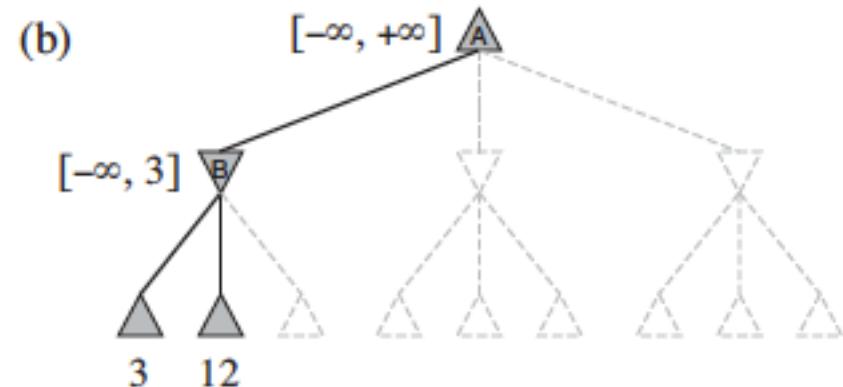
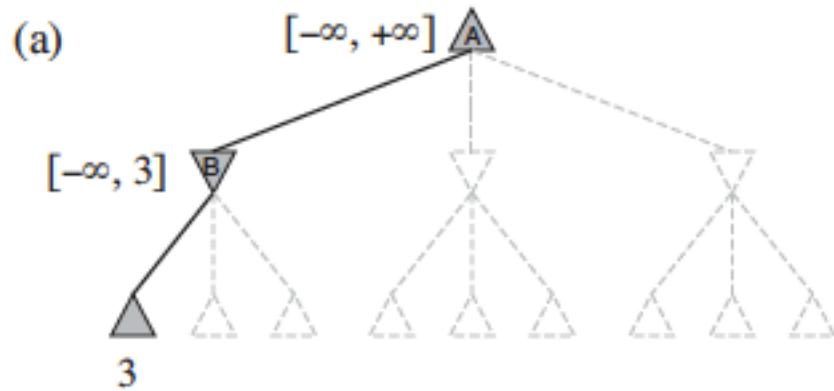
**MIN**

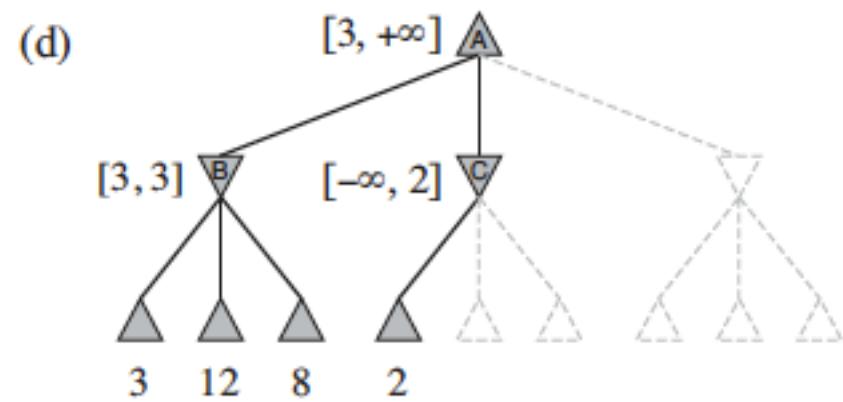
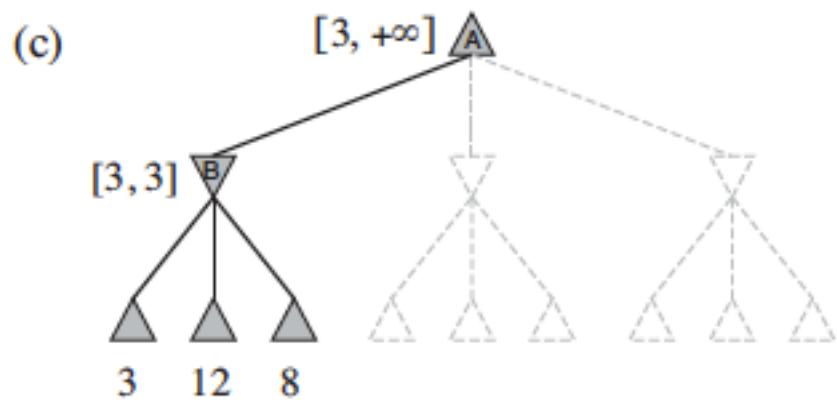
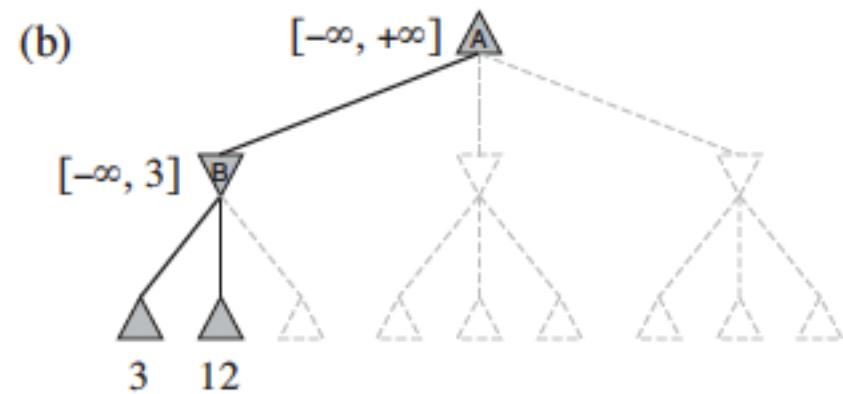
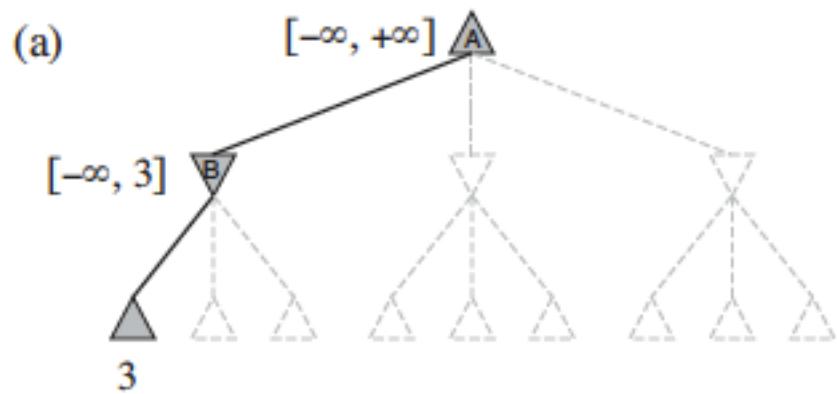


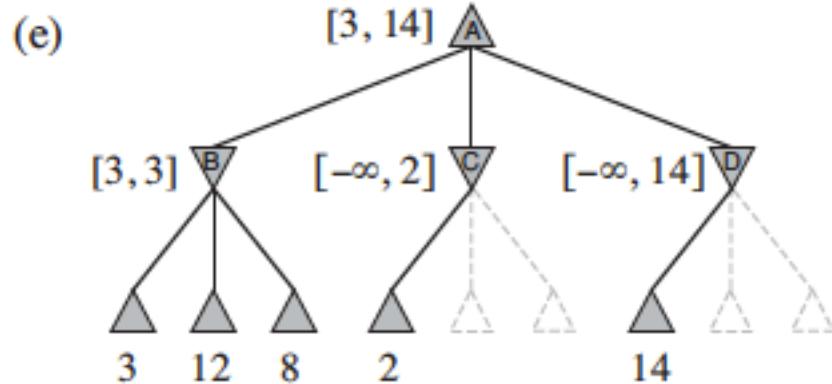
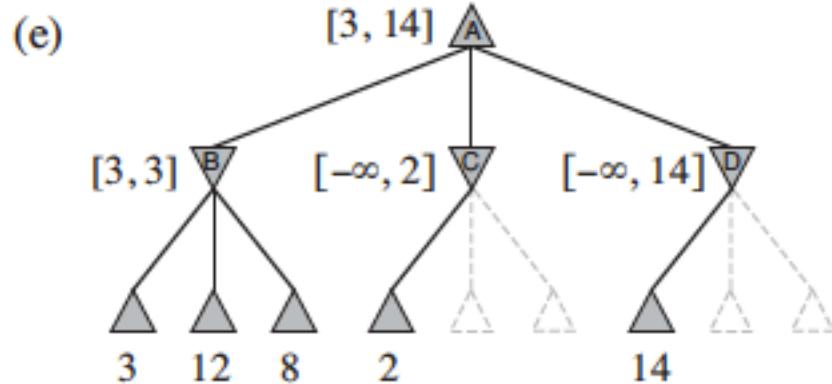
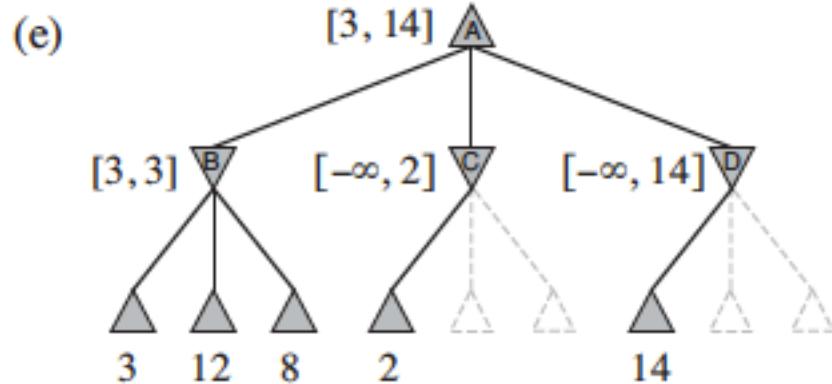
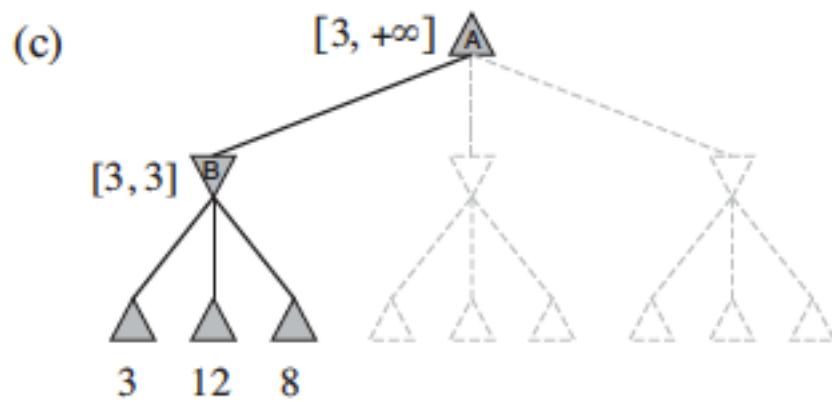
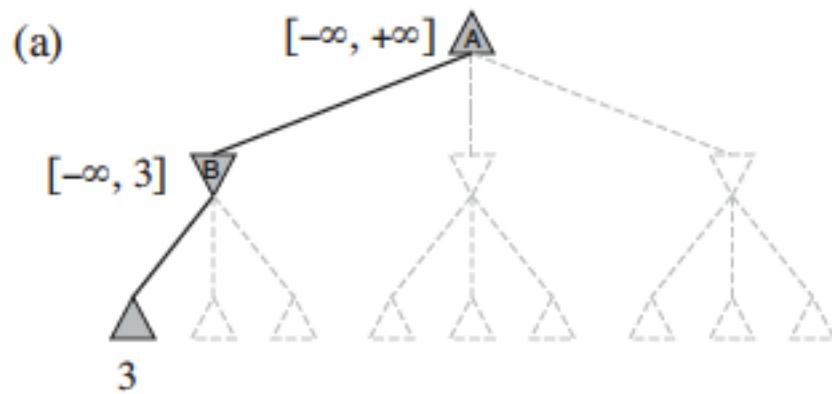
(a)

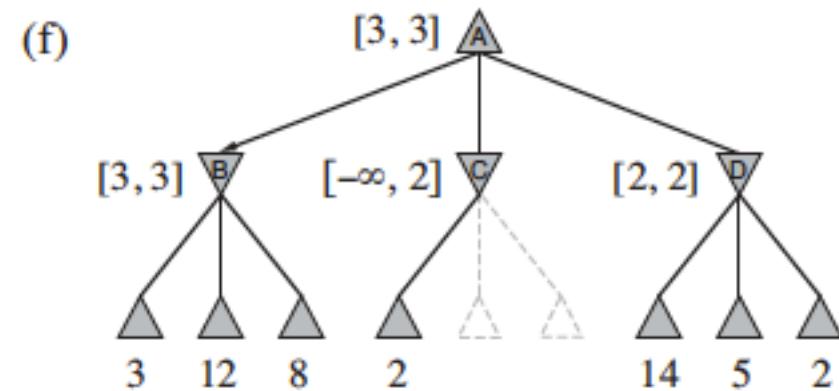
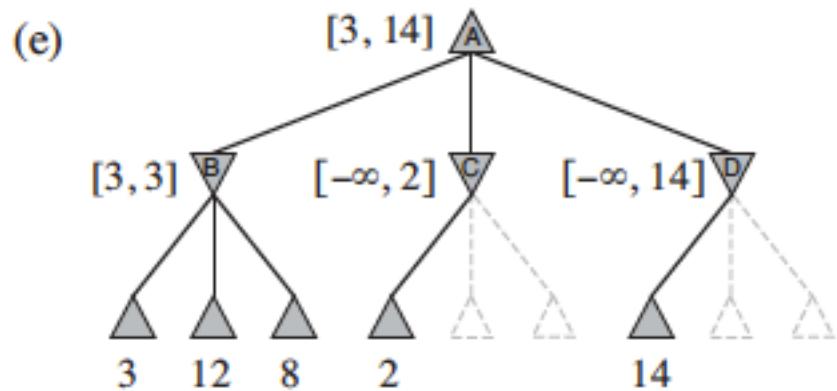
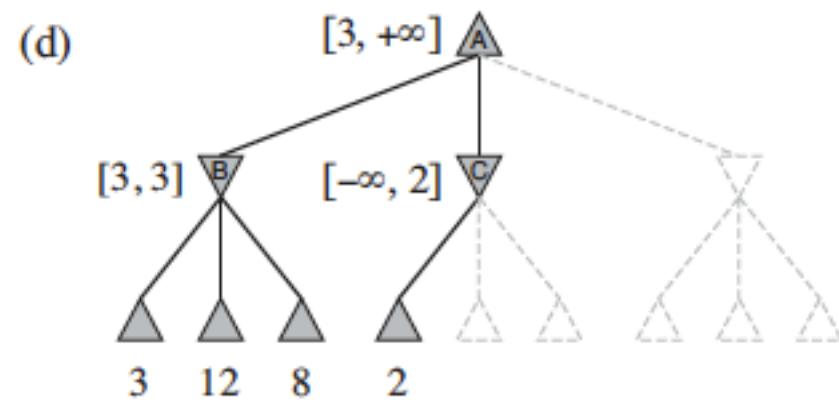
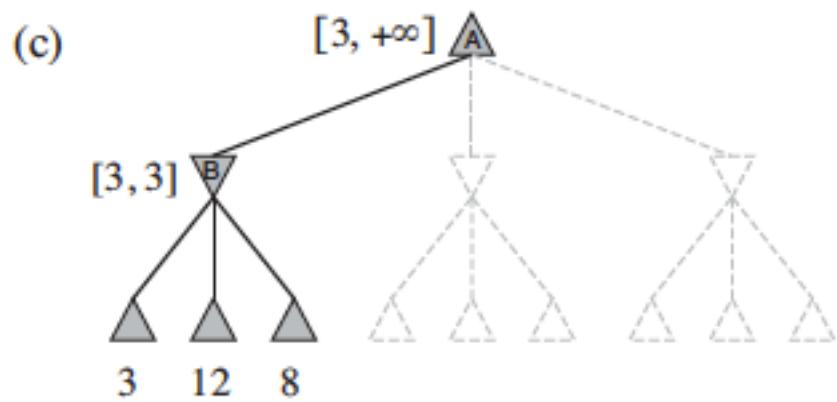
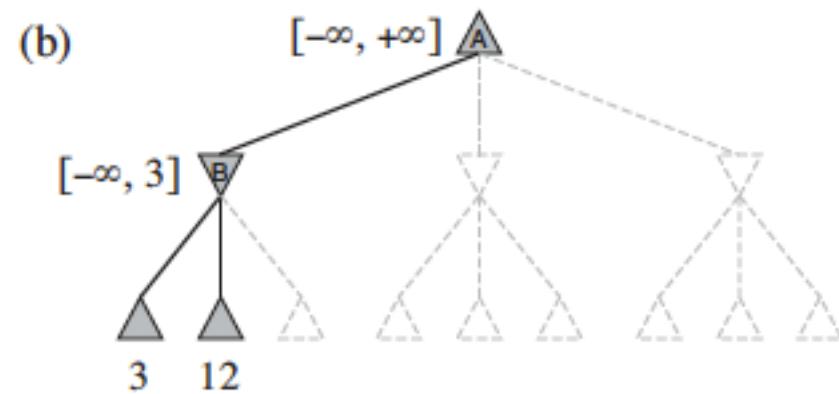
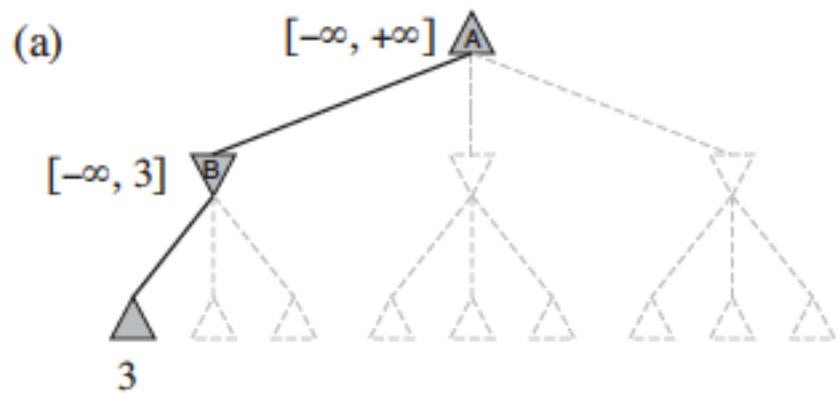




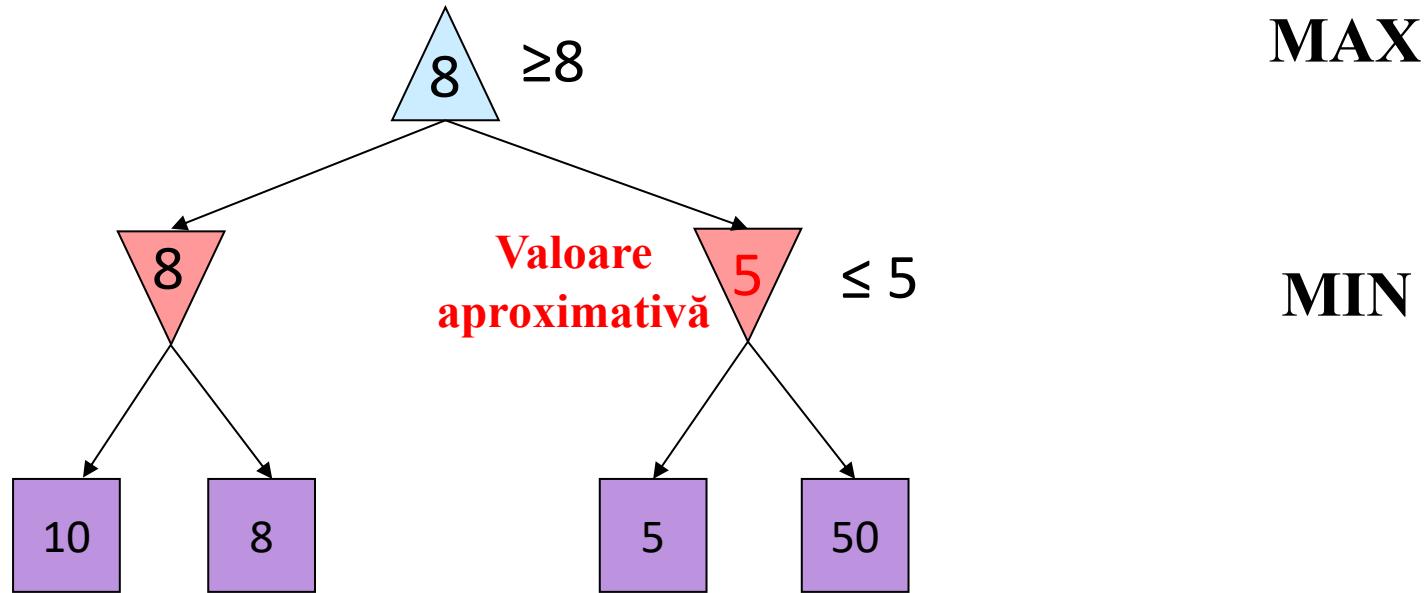




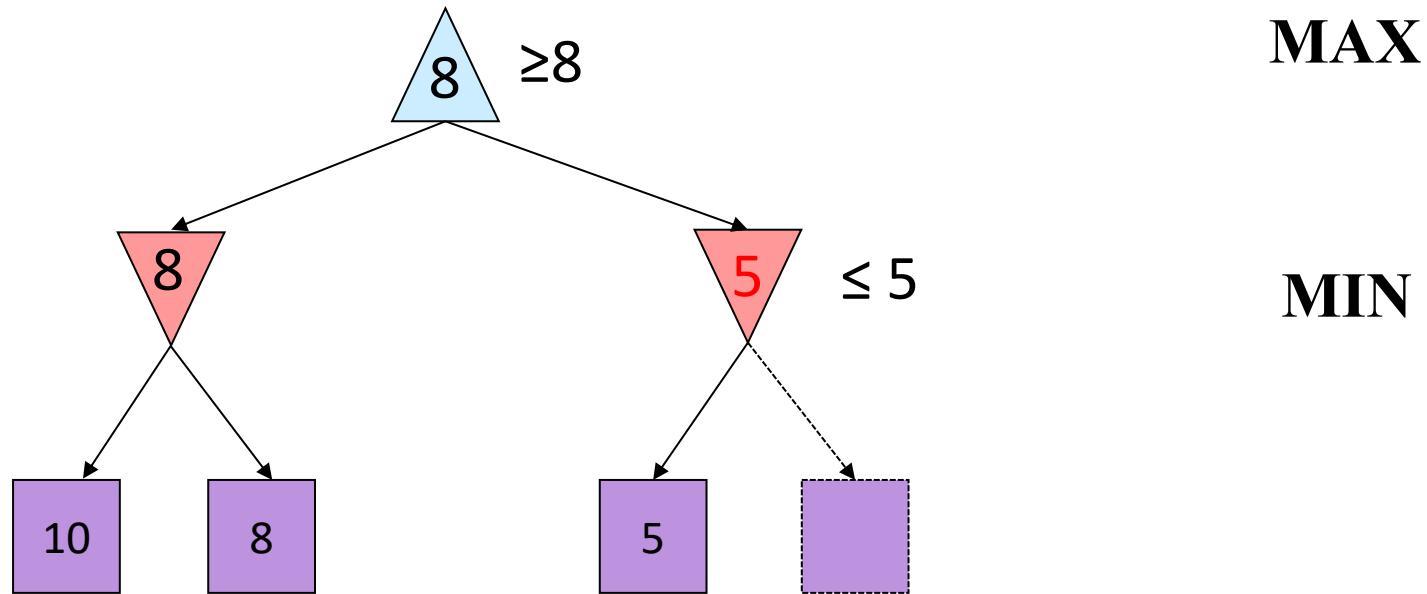




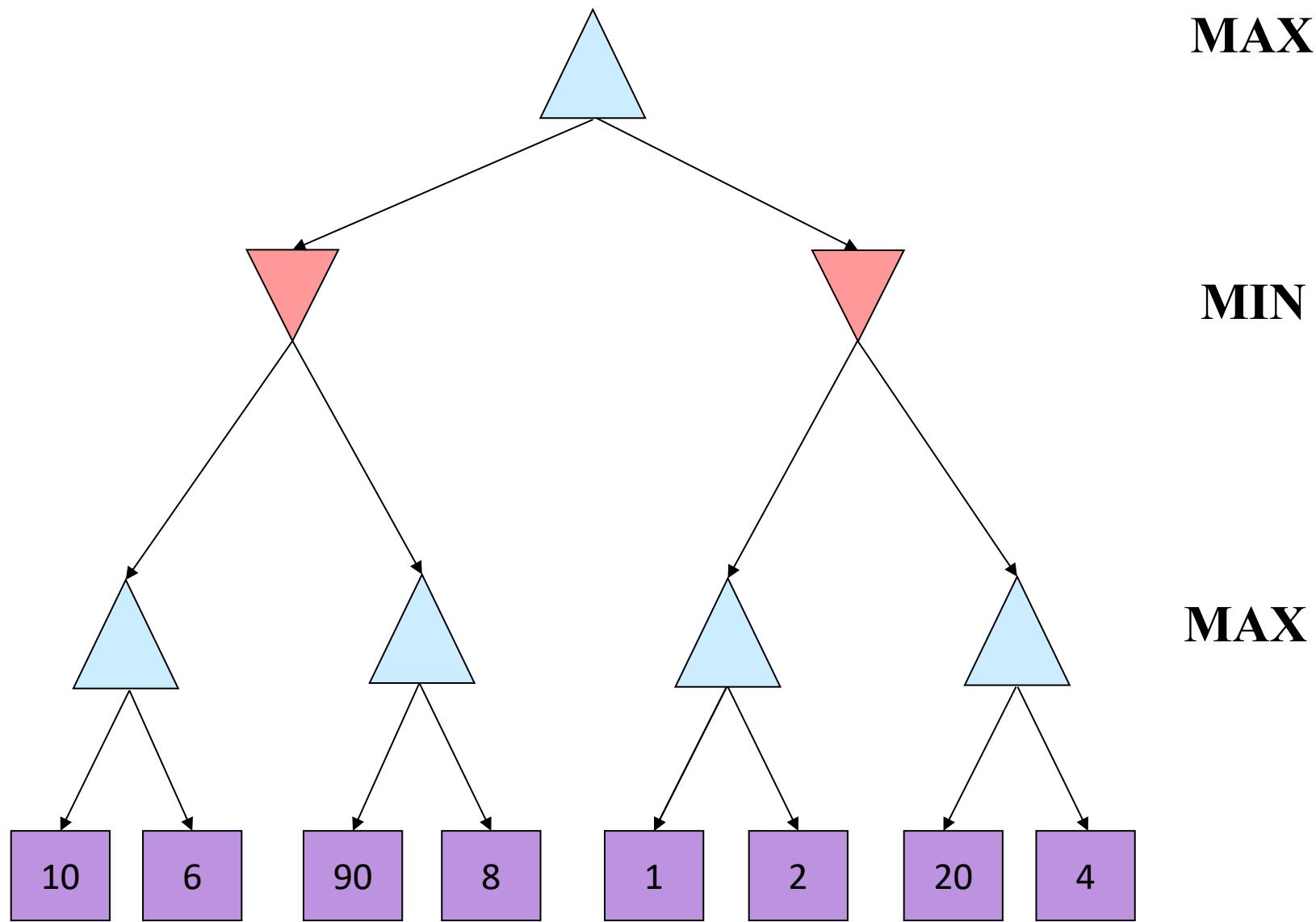
# Exemplu



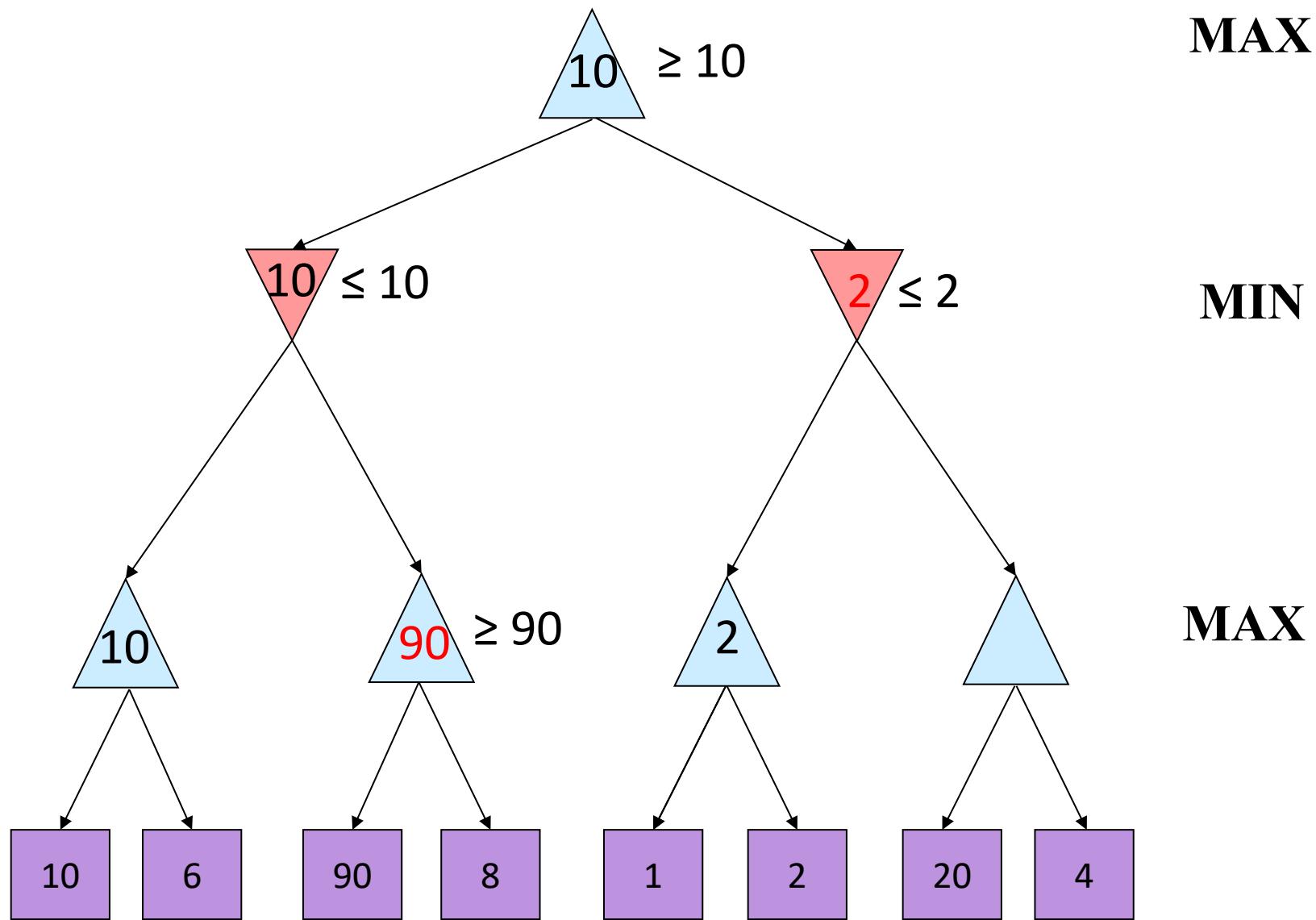
# Exemplu



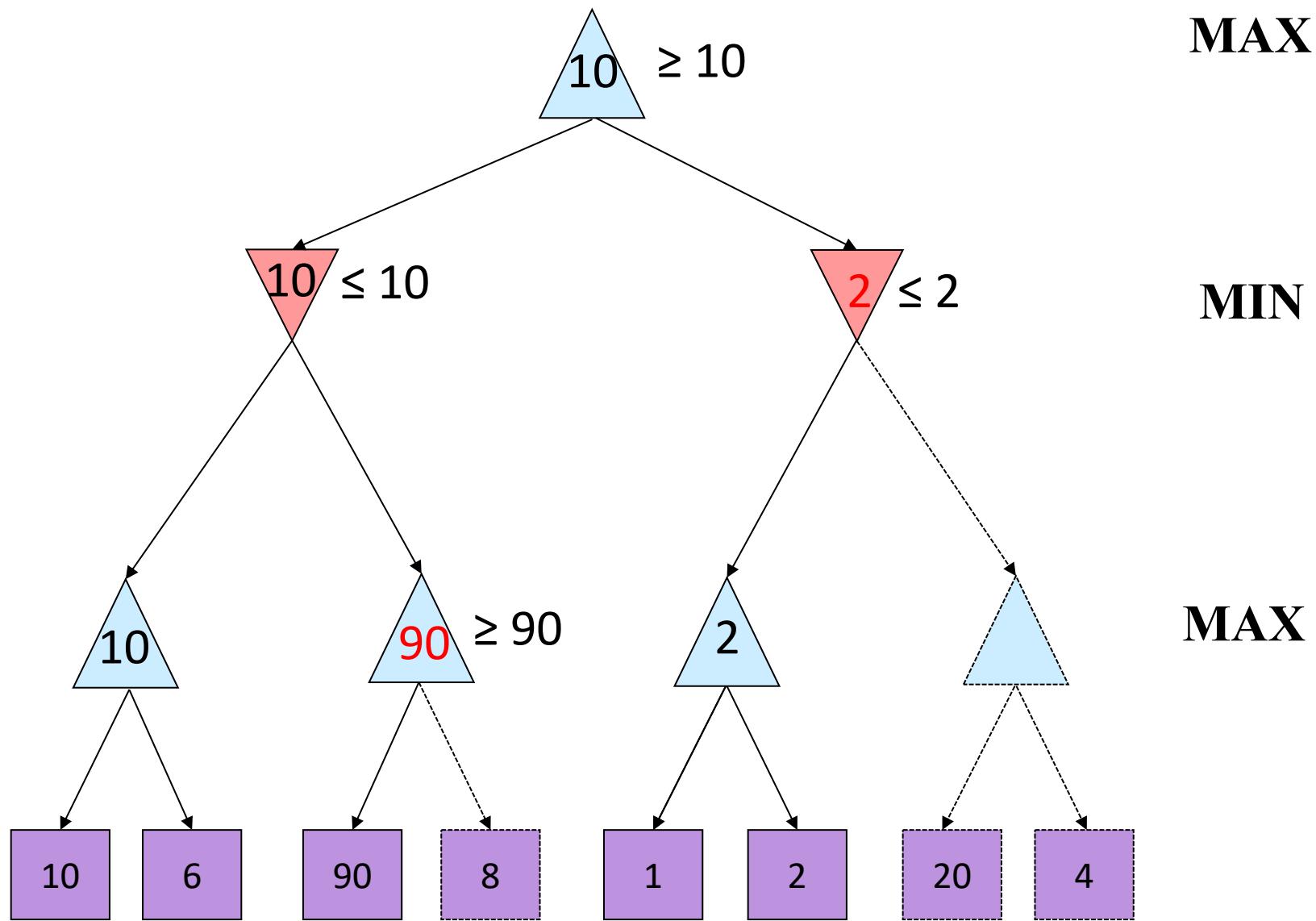
# Exemplu



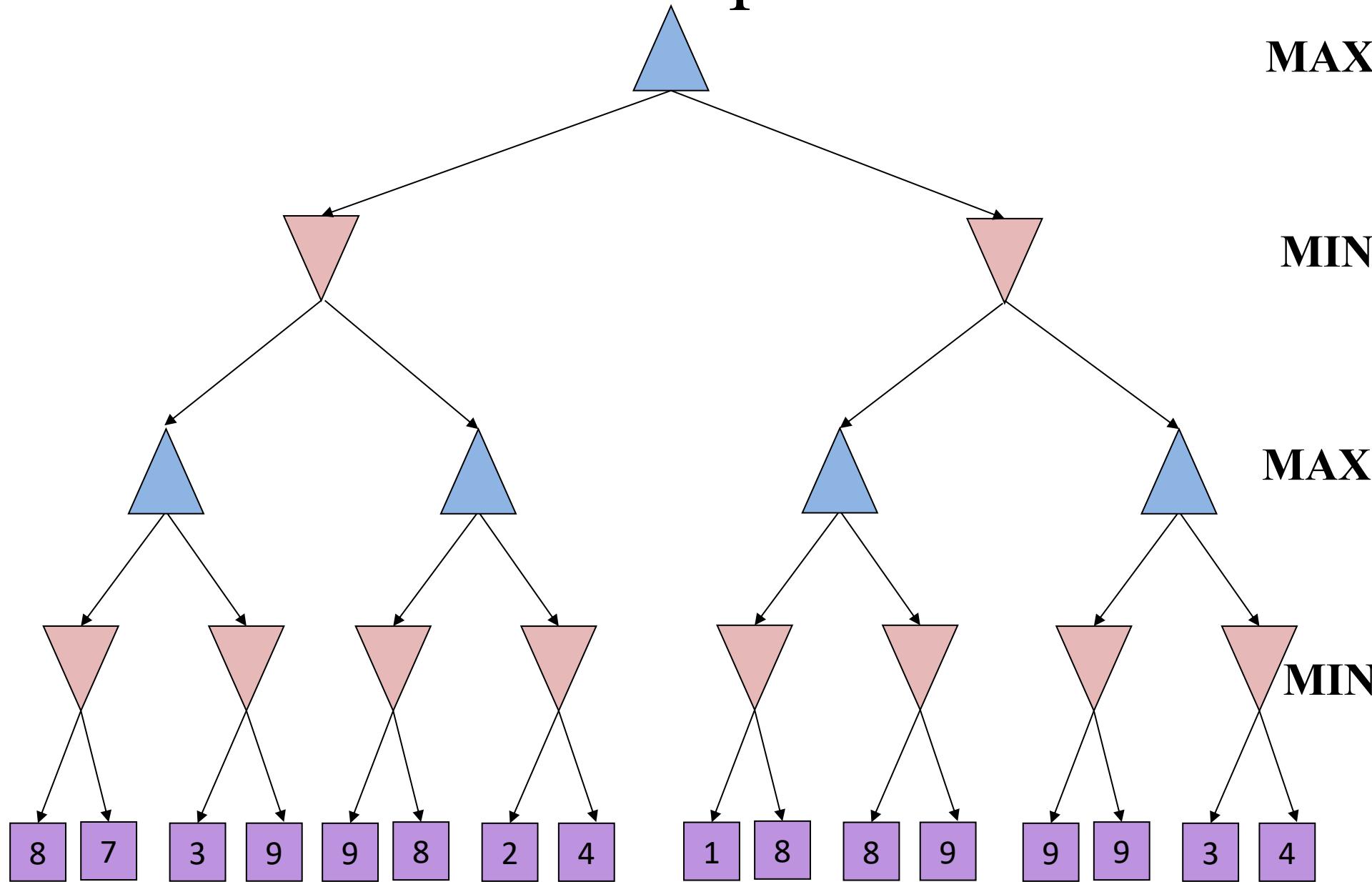
# Exemplu



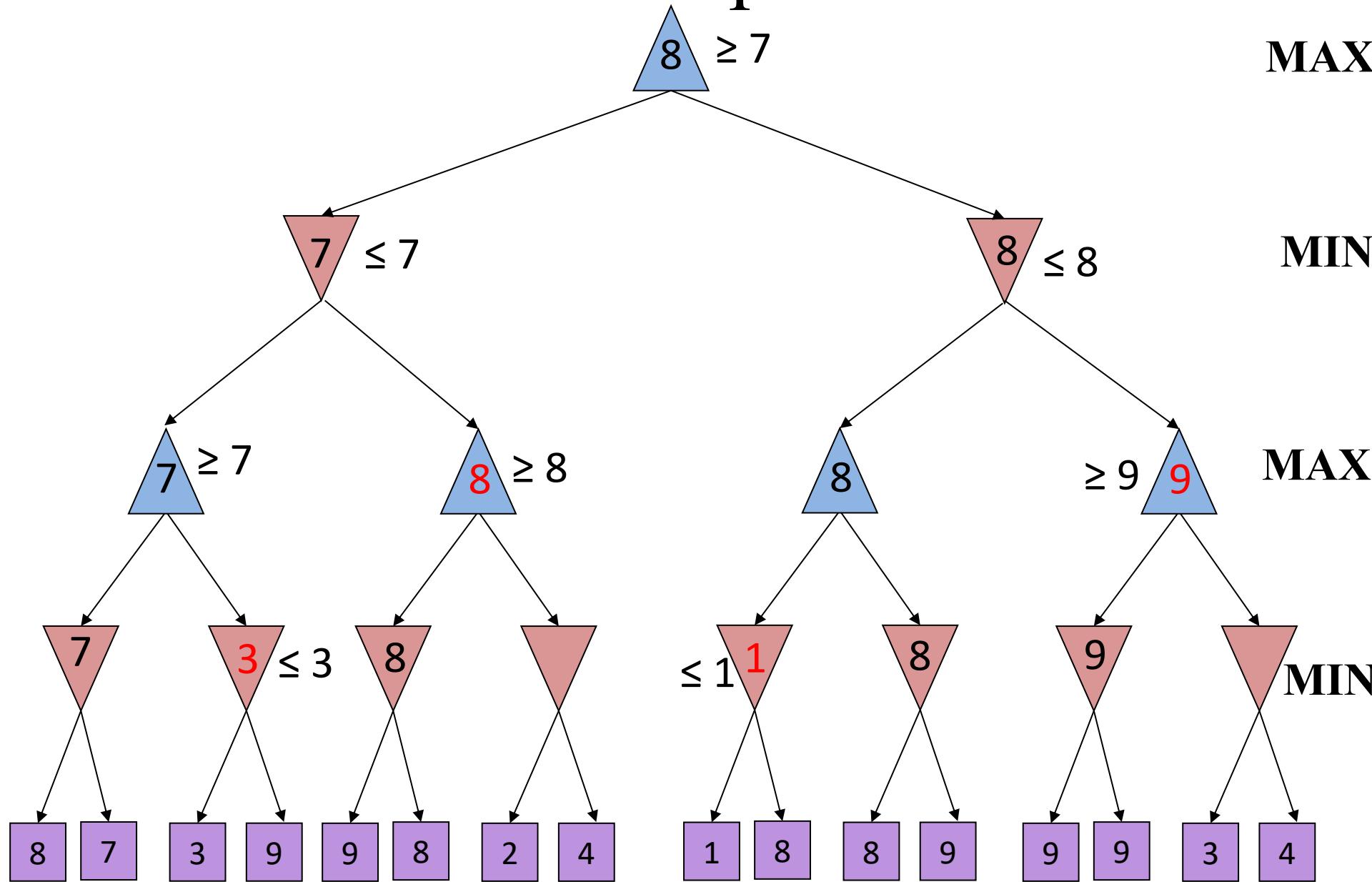
# Exemplu



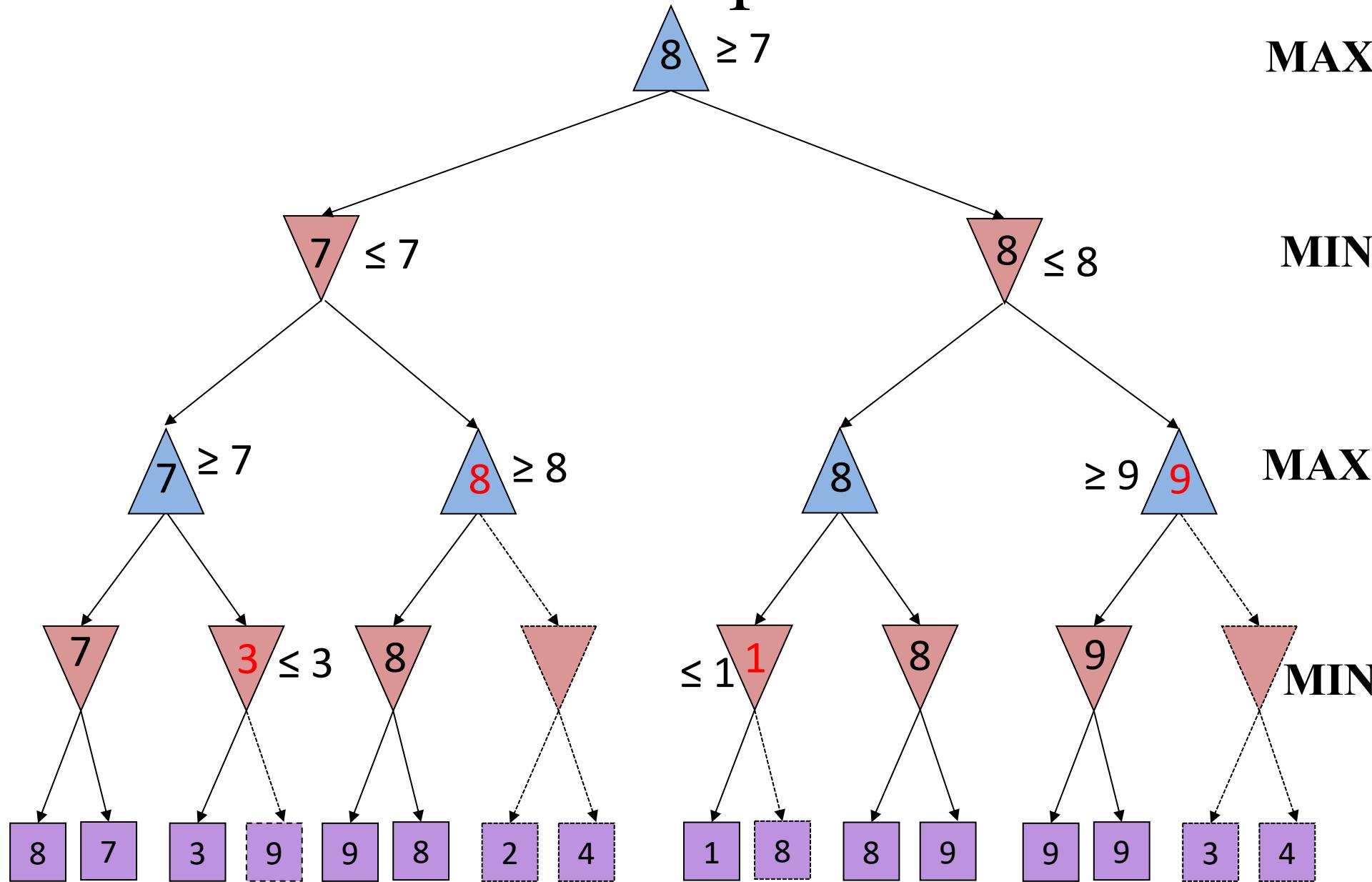
# Exemplu



# Exemplu

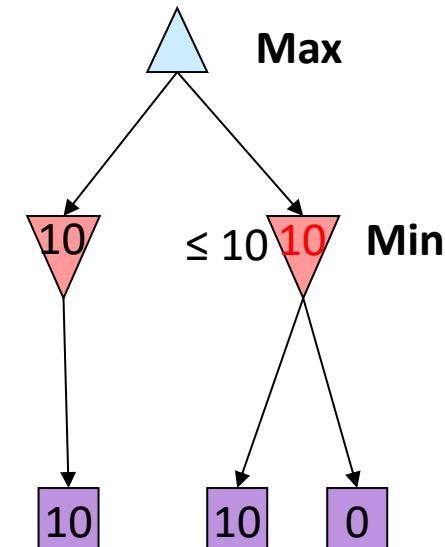


# Exemplu



# Proprietățile algoritmului Alpha-Beta Pruning

- procesul de retezare nu are niciun efect asupra valorii minimax calculate pentru nodul rădăcină (aceasta nu se schimbă)
- valorile nodurilor intermediare (aproximate) pot fi greșite:
  - important: nodurile fii ale rădăcinii pot avea valori greșite  
Implementarea naivă poate conduce la erori
- vizitarea în ordinea bună a nodurilor fii conduce la retezări mai rapide
- cu “ordonare perfectă”:
  - complexitatea timp poate scădea la  $O(b^{m/2})$
  - algoritmul poate rula pe o adâncime dublă
  - căutarea exhaustivă nu este totuși posibilă...



# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 12

# Recapitulare – cursul trecut

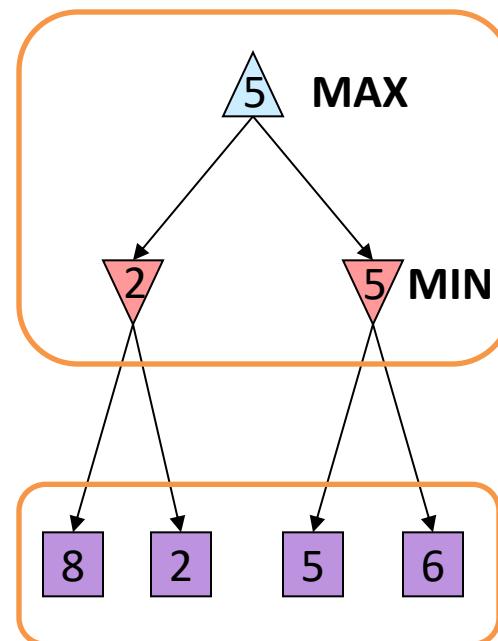
## 1. Căutare adversarială:

- jocuri adversariale
- algoritmul MINIMAX
- algoritmul  $\alpha$ - $\beta$  retezare ( $\alpha$ - $\beta$  pruning)

# Algoritmul MiniMax pentru căutare adversarială

- Jocuri de sumă zero cu doi jucători, deterministe, cu informație perfectă
  - X și 0, săh, dame
  - un jucător (MAX) maximizează utilitatea
  - celălalt jucător (MIN) minimizează utilitatea
- Căutare MiniMax:
  - arbore de căutare în spațiul stărilor
  - jucătorii mută pe rând
  - calculăm pentru fiecare nod valoarea **MiniMax**: cea mai bună utilitate care poate fi obținută împotriva unui adversar care joacă perfect
  - determină strategia optimă corespunzătoare lui MAX, care este cea mai bună primă mutare

**Valori minimax:**  
calculate recursiv

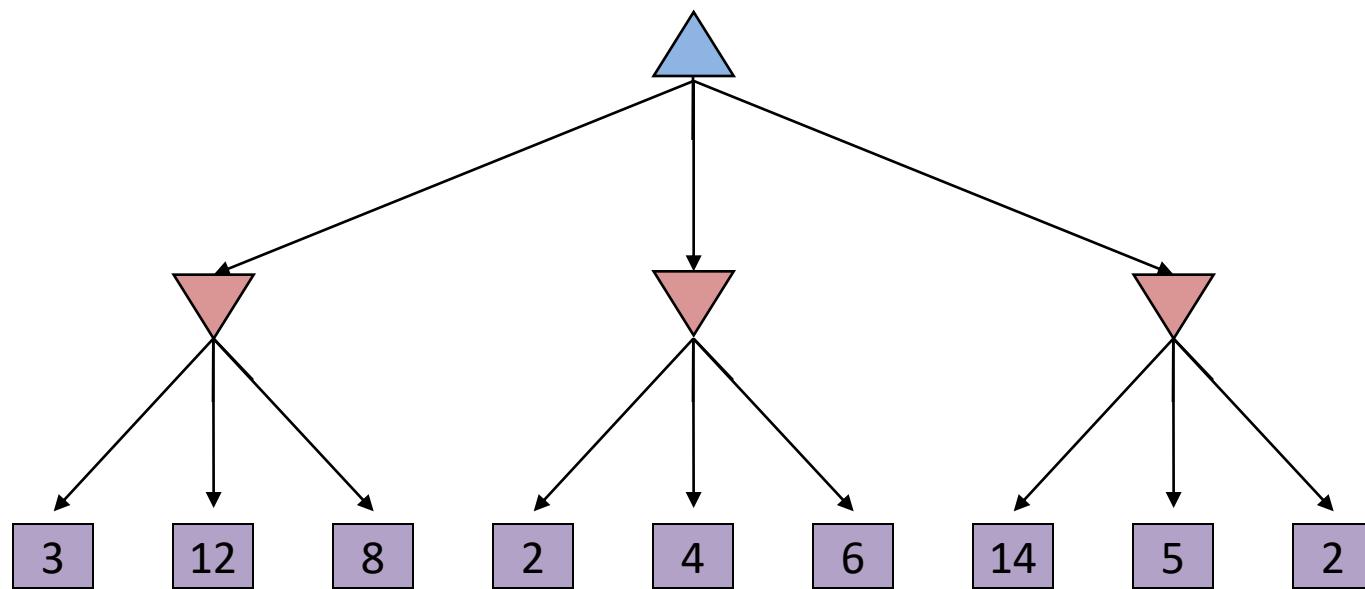


**Valori terminale:**  
date de regulile jocului

# Exemplul MiniMax

**MAX**

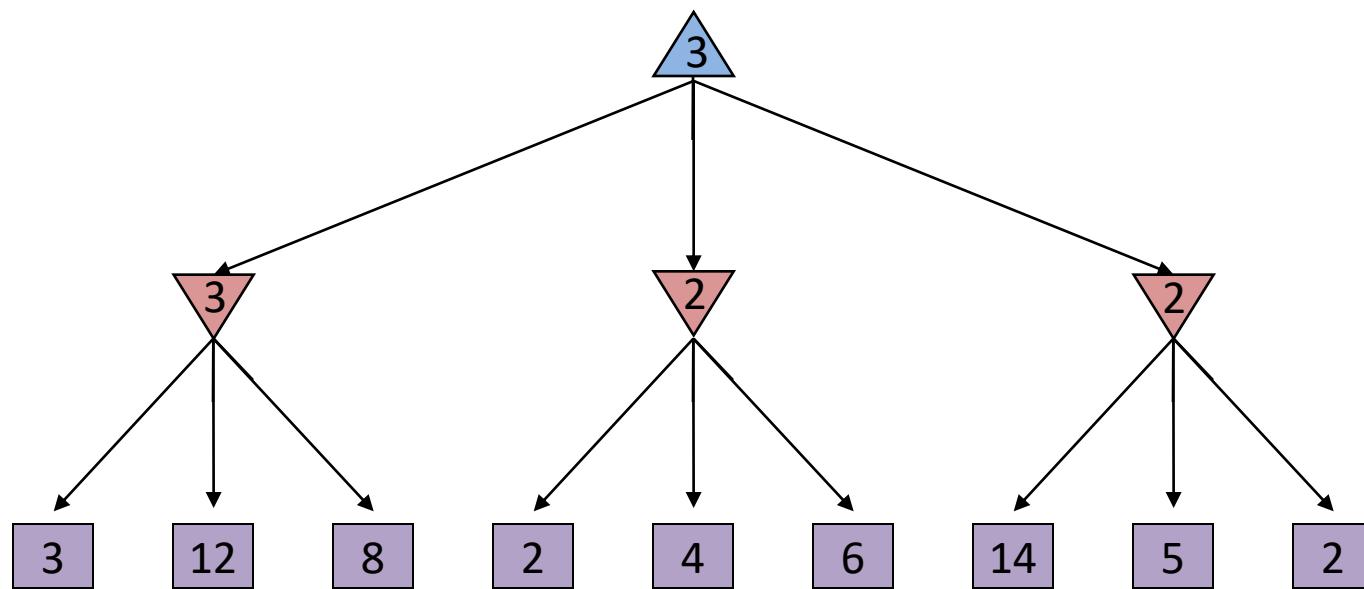
**MIN**



# Exemplul MiniMax

**MAX**

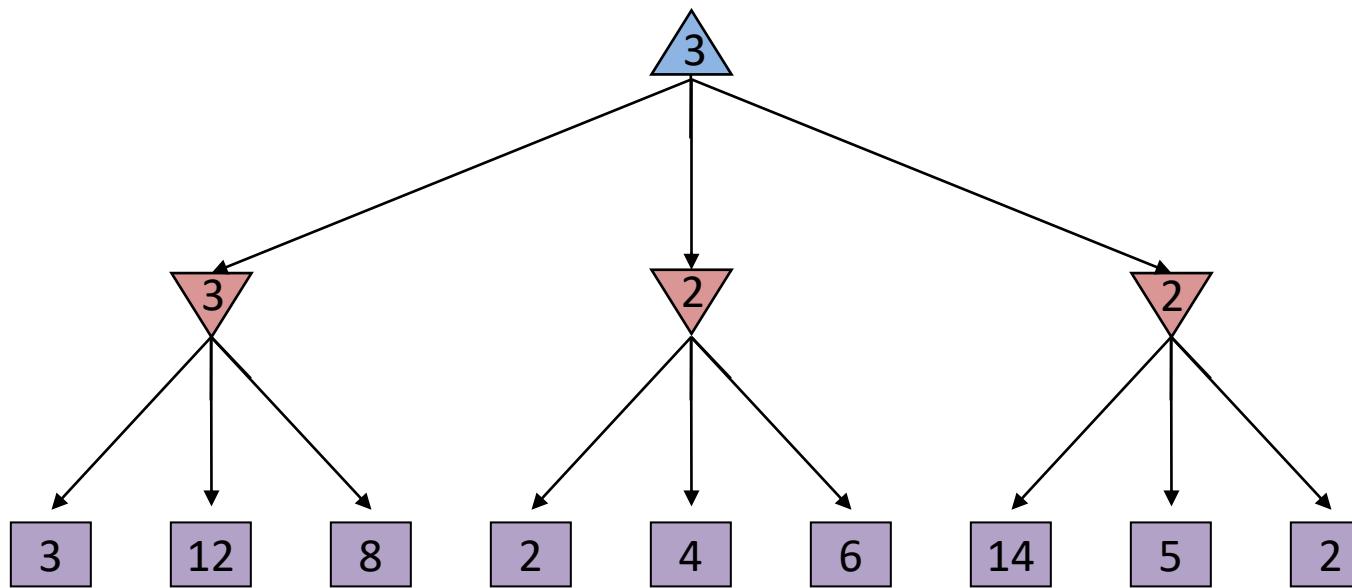
**MIN**



# Exemplul MiniMax

MAX

MIN



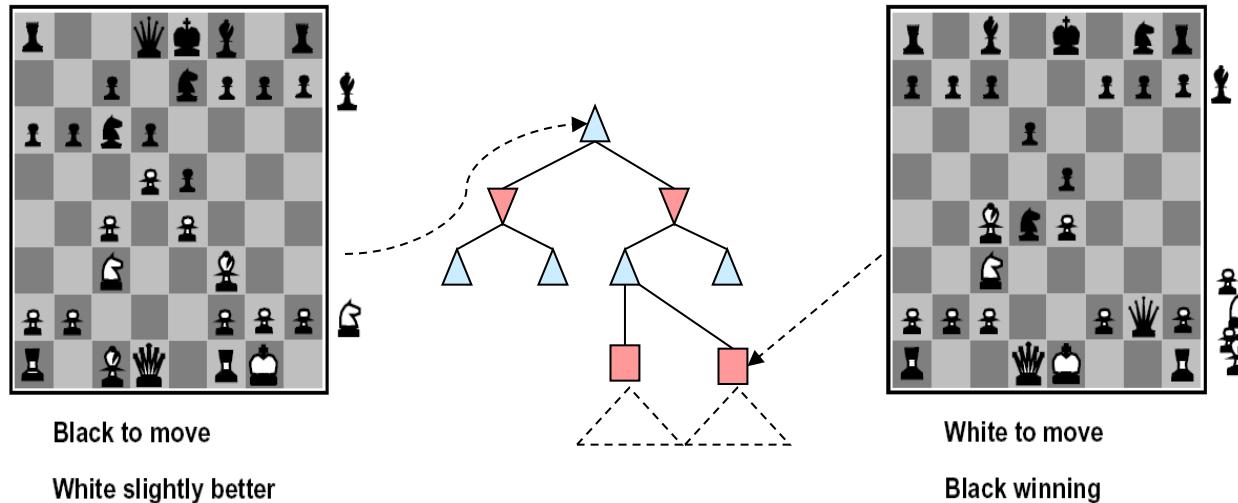
- valorile pozițiilor de la ultimul nivel sunt determinate de către funcția de utilitate și se numesc valori statice (se realizează evaluări statice).
- valorile minimax ale nodurilor interne sunt calculate în mod dinamic, în manieră bottom-up, nivel cu nivel, până când este atins nodul-rădăcină.
- valoarea rezultată este 3 și prin urmare cea mai bună mutare a lui MAX din poziția curentă este mutarea la stânga. Cel mai bun răspuns al lui MIN este mutarea la stânga. Această secvență a jocului poartă denumirea de variație principală. Ea definește jocul optim de tip minimax pentru ambele părți.
- se observă ca valoarea pozițiilor de-a lungul variației principale nu variază. Prin urmare, **mutările corecte sunt cele care conservă valoarea jocului**.

# Limitări date de resurse ale algoritmului MiniMax

1. Generează întreg arborele de joc până la stările terminale
2. Aplică funcția de utilitate fiecărei stării terminale - obține valoarea stării
3. Deplasează-te înapoi în arbore, de la nodurile-frunze spre nodul-rădăcină, determinând, corespunzător fiecărui nivel al arborelui, valorile care reprezintă utilitatea nodurilor aflate la acel nivel. Propagarea acestor valori la niveluri anterioare se face prin intermediul nodurilor-părinte succesive, conform următoarei reguli:
  - dacă starea-părinte este un nod de tip MAX, atribuie-i maximul dintre valorile avute de fii săi;
  - dacă starea-părinte este un nod de tip MIN, atribuie-i minimul dintre valorile avute de fii săi;
4. Ajuns în nodul-rădăcină, alege pentru MAX acea mutare care conduce la valoarea maximă. Mutarea se numește ***decizia minimax*** - maximizează utilitatea, în ipoteza că oponentul joacă perfect cu scopul de a o minimiza.

# Funcții de evaluare - șah

- Funcțiile de evaluare asociază un scor stărilor neterminale (este folosită de căutarea în adâncime limitată, căutarea în adâncime iterativă)



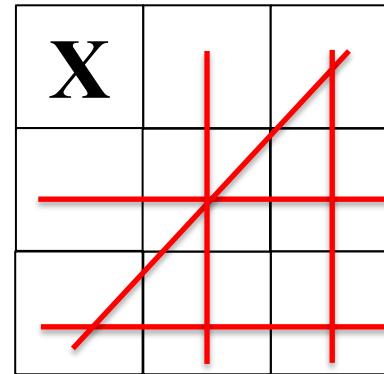
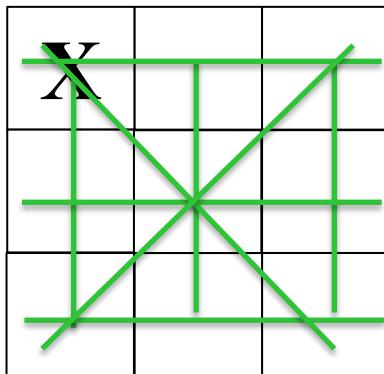
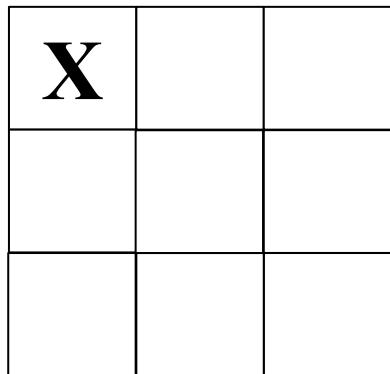
- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc: pentru șah se consideră funcții liniare în care ponderăm piesele de pe tablă

$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s)$$

- unde,  $f_1(s) = (\#\text{regine\_albe} - \#\text{regine\_negre})$ ,  $w_1 = 100$ , etc.

# Functii de evaluare – X și 0

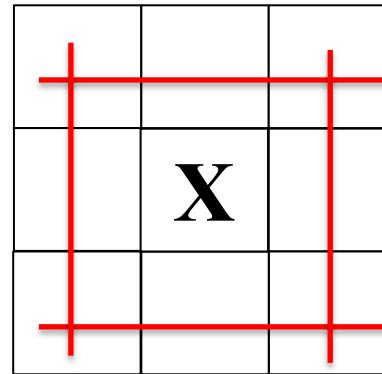
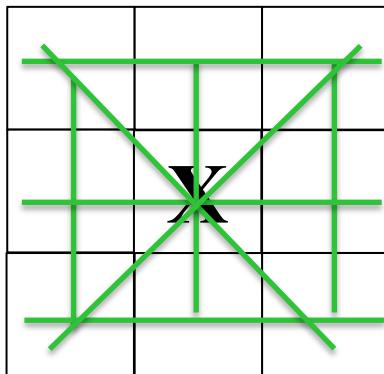
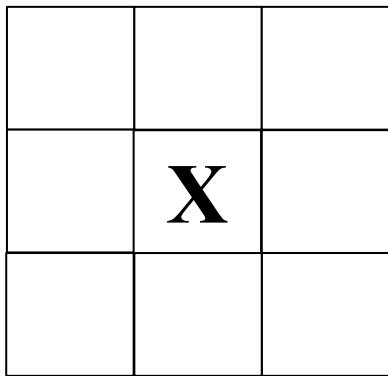
- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:



- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- în exemplul de mai sus:  $8 - 5 = 3$

# Functii de evaluare – X și 0

- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:

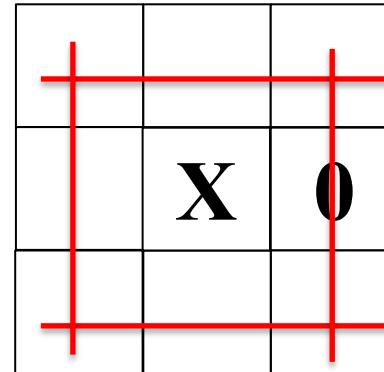
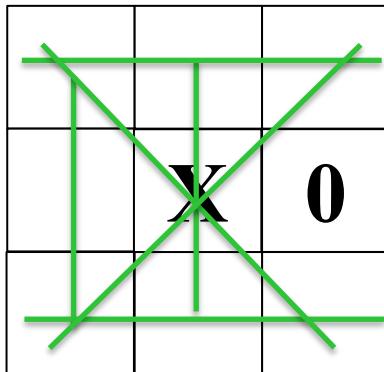


- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- În exemplul de mai sus:  $8 - 4 = 4$

# Functii de evaluare – X și 0

- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:

	X	0

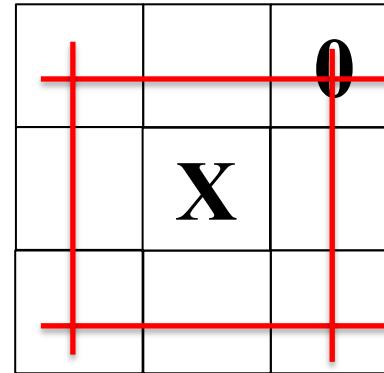
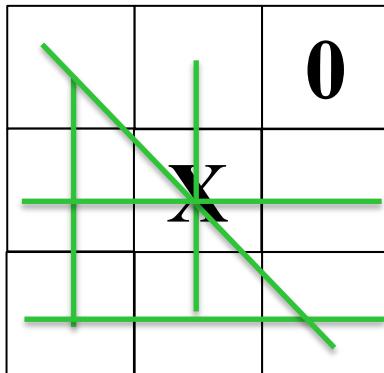


- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- În exemplul de mai sus:  $6-4 = 2$

# Functii de evaluare – X și 0

- o funcție de evaluare ideală returnează valoarea MiniMax a stării.
- diverse funcții de evaluare specifice pentru fiecare joc:

		0
	X	

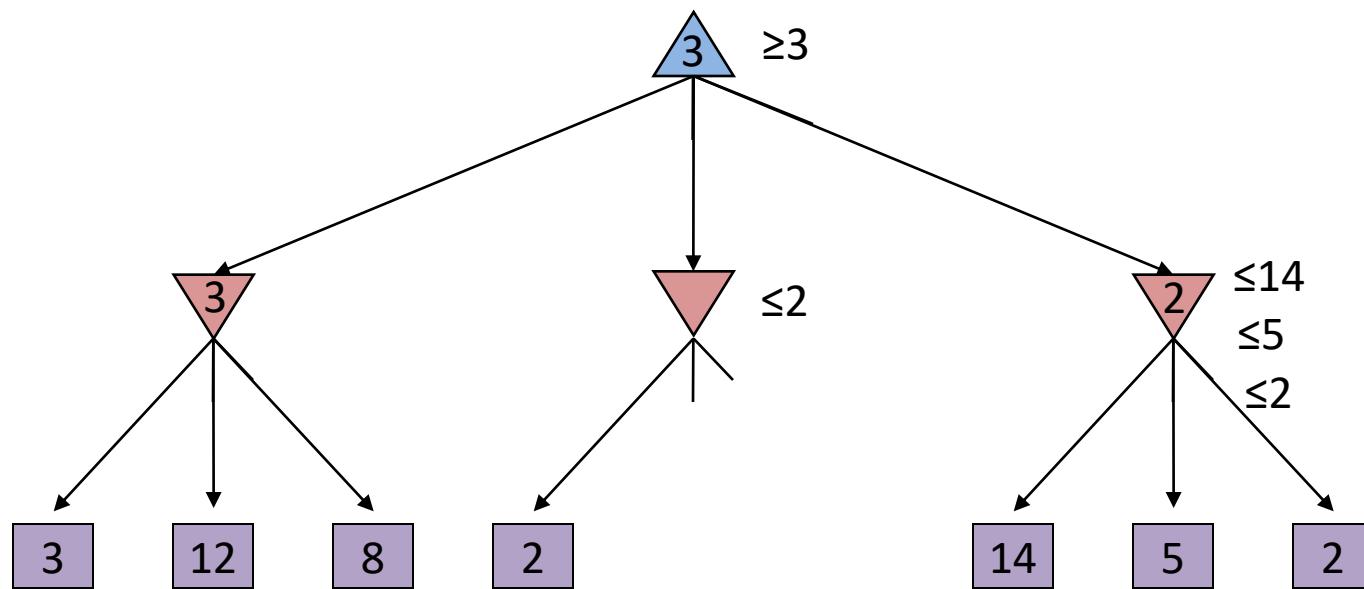


- exemplu de funcție de evaluare pentru X și 0: diferența dintre numărul de soluții disponibile pentru MAX și numărul de soluții disponibile pentru MIN
- În exemplul de mai sus:  $5 - 4 = 1$

# Accelerarea algoritmului MiniMax

**MAX**

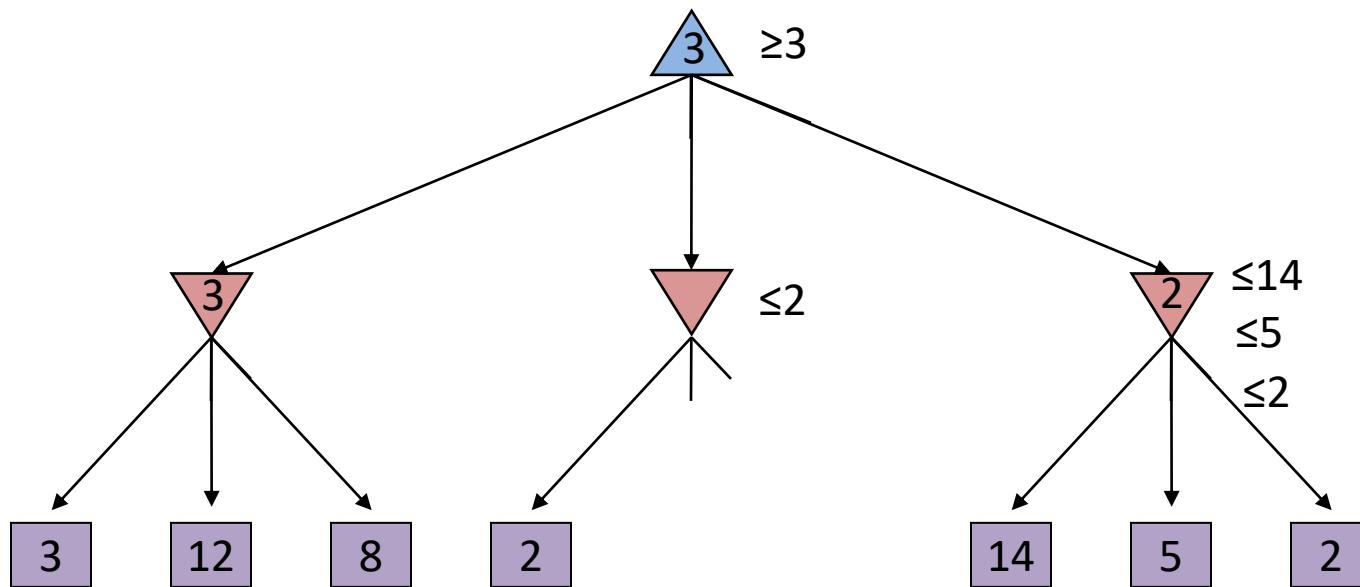
**MIN**



# Retezare alfa-beta

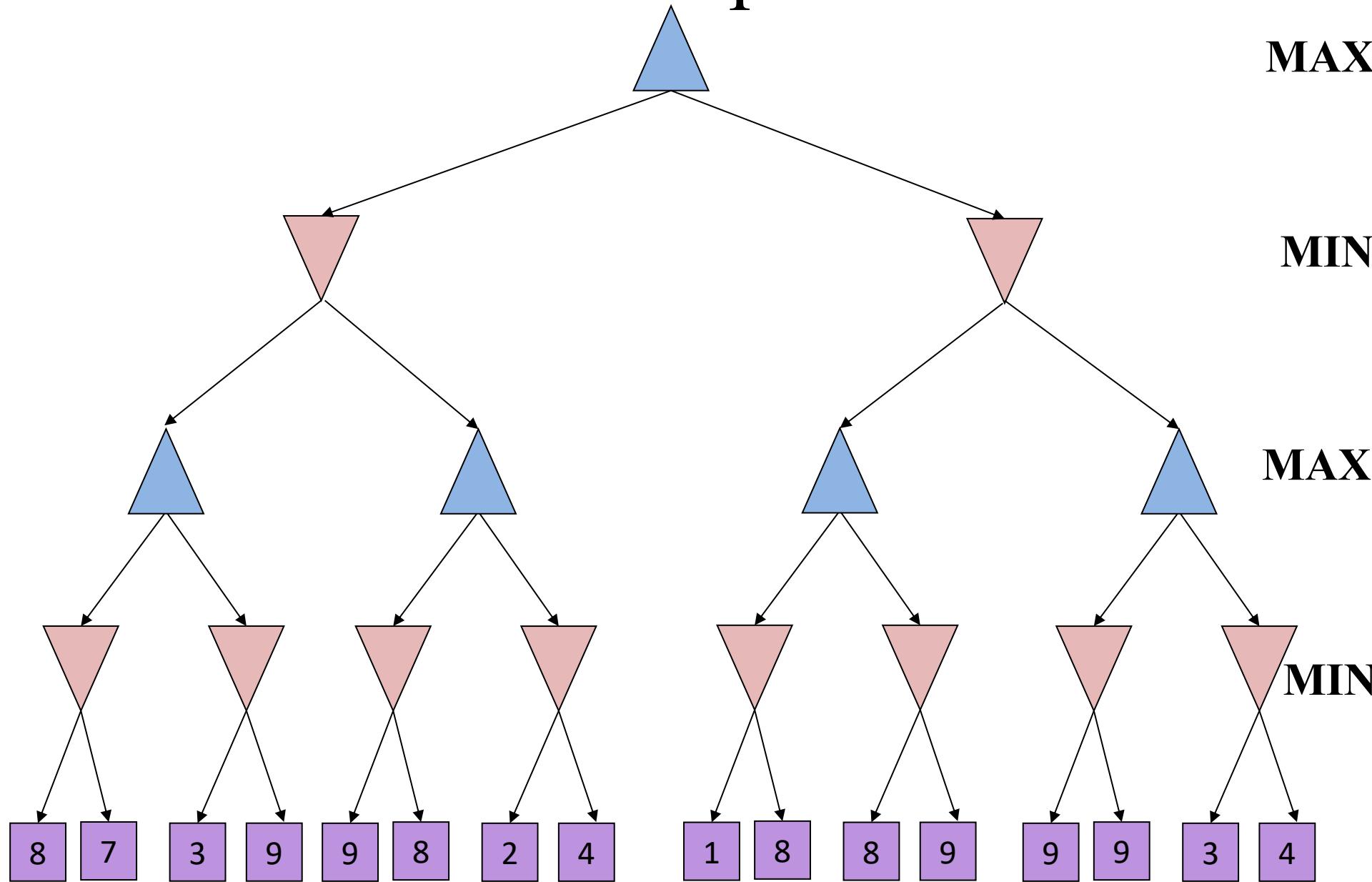
MAX

MIN

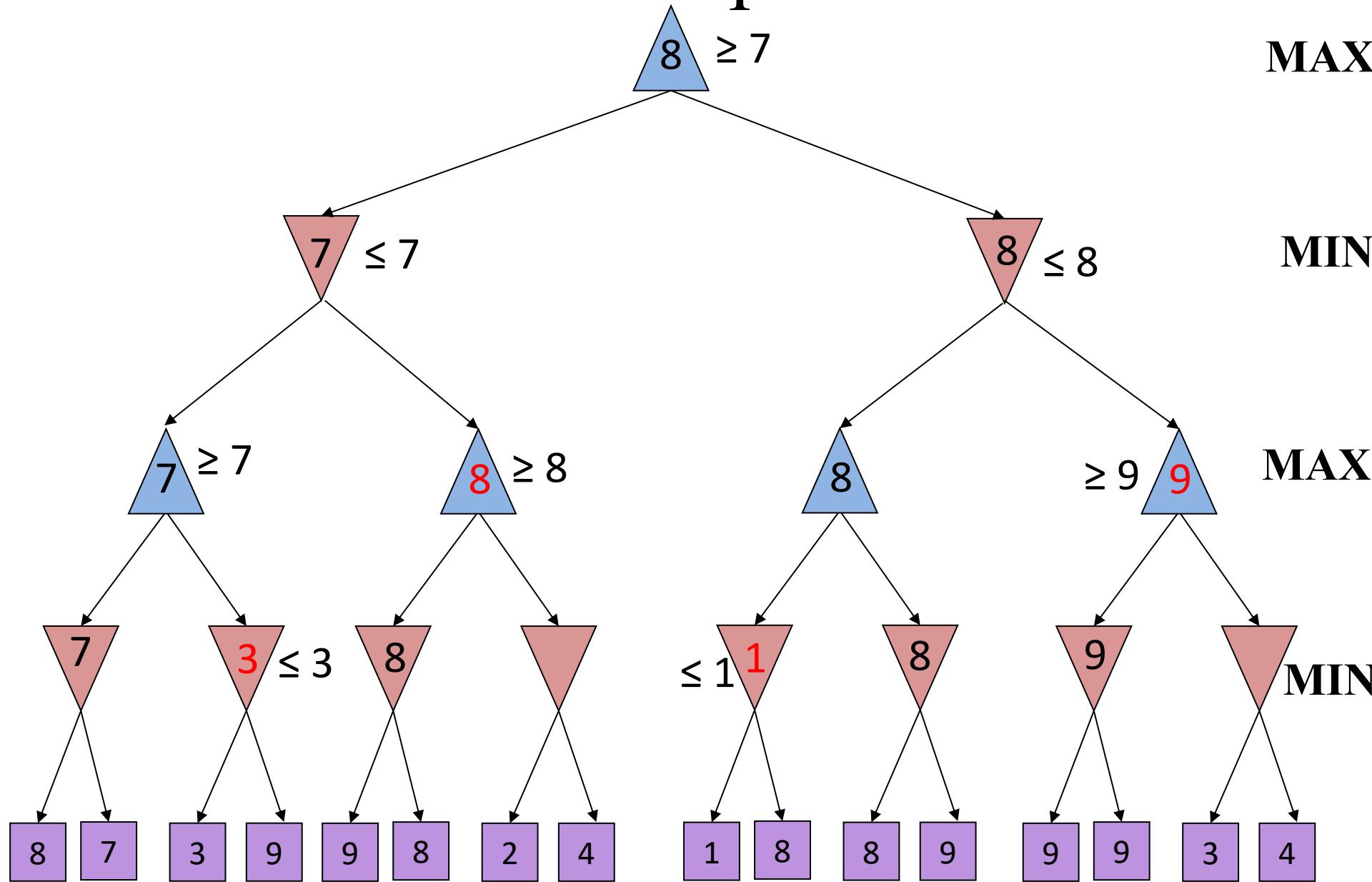


Tehnica de alfa-beta retezare, când este aplicată unui arbore de tip Minimax standard, va întoarce aceeași mutare pe care ar furniza-o și Algoritmul MiniMax, dar într-un timp mai scurt, întrucât realizează o retezare a unor ramuri (subarbore) ale arborelui care nu pot influența decizia finală și care nu mai sunt vizitate.

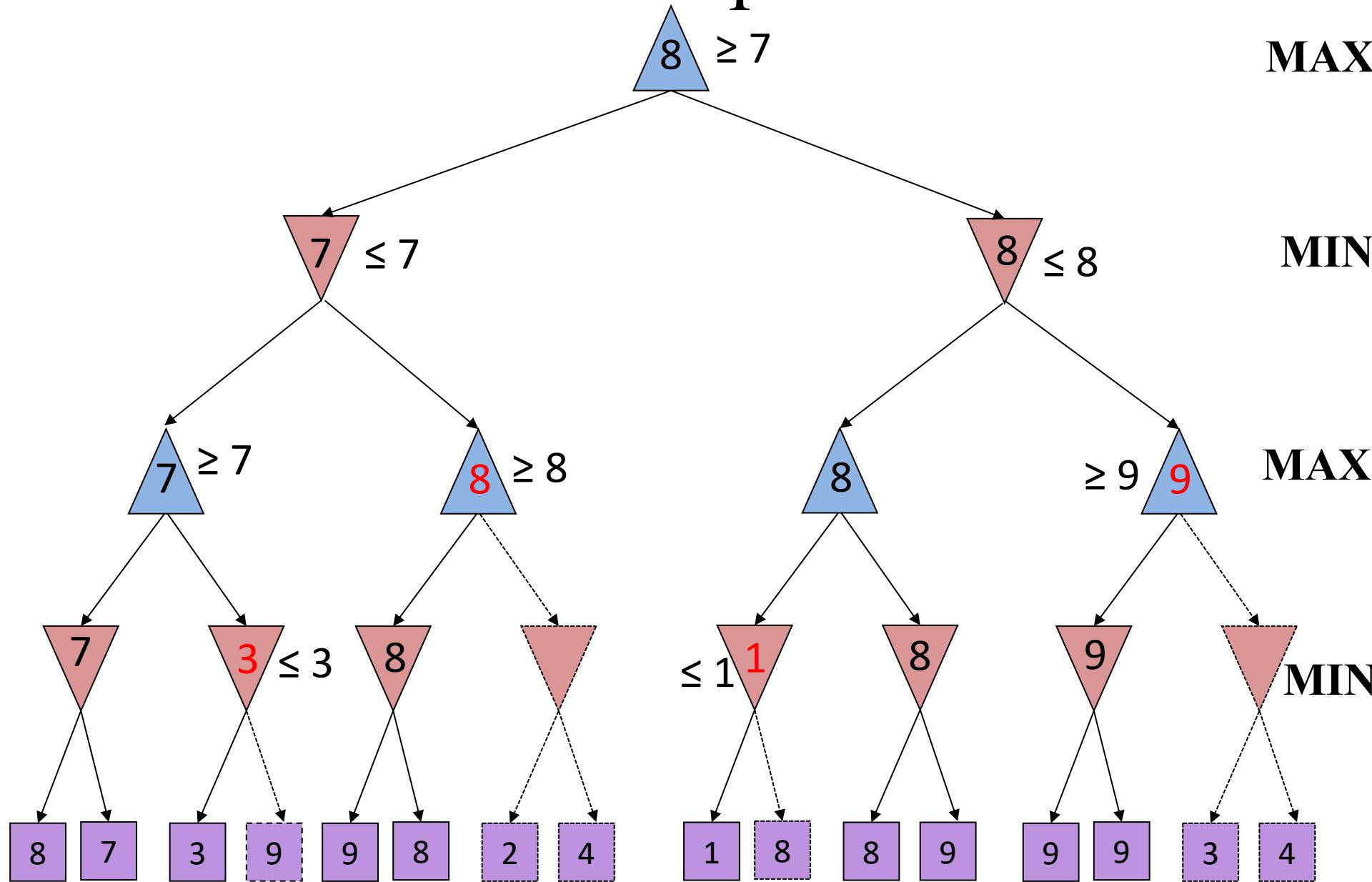
# Exemplu



# Exemplu



# Exemplu



# Cursul de azi

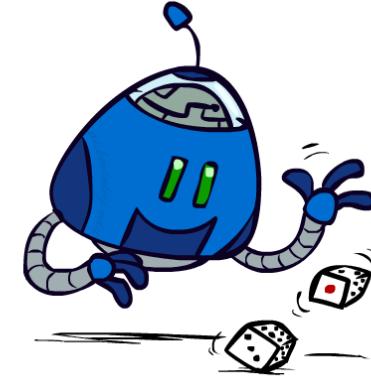
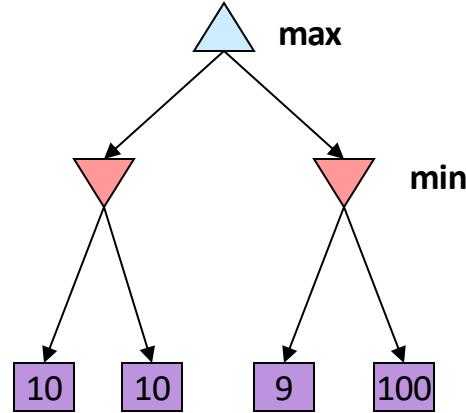
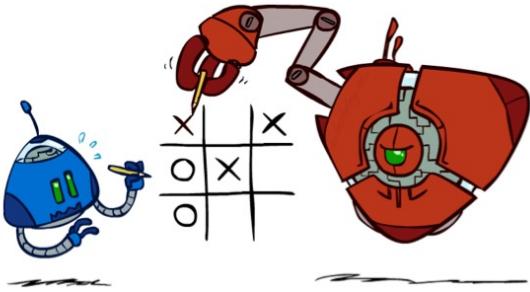
## 1. Căutare în jocuri cu incertitudine:

- căutare ExpectiMax

## 2. Examenul final

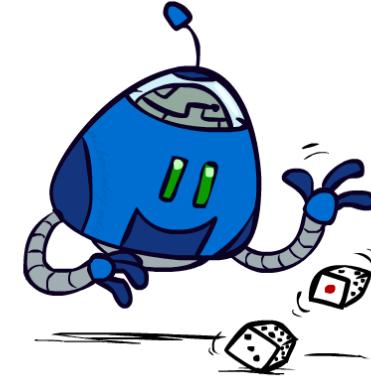
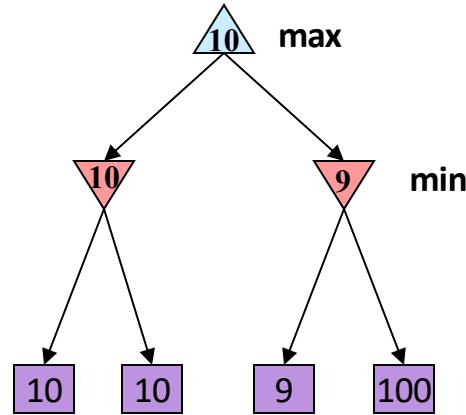
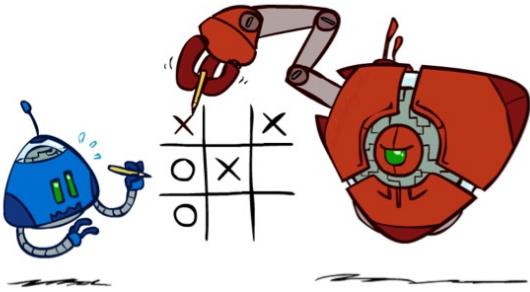
- organizarea examenului din acest an
- subiectele din anii trecuți
- rezolvările subiectelor din anii trecuți

# Cazul cel mai defavorabil vs cazul mediu



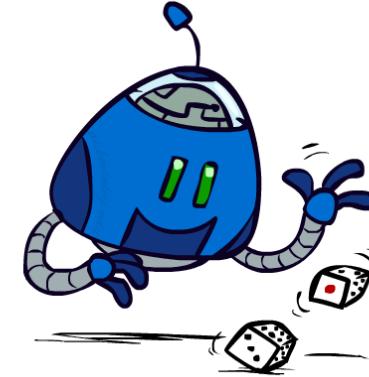
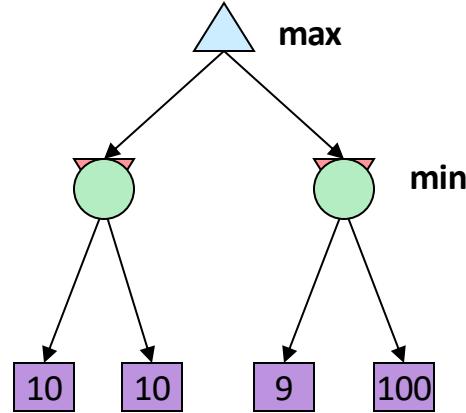
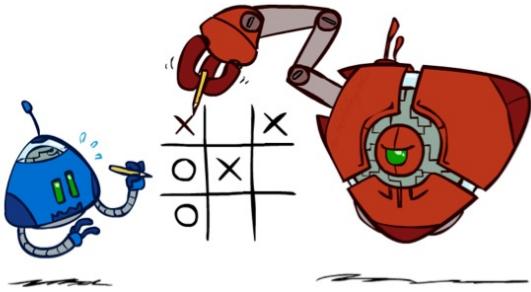
Idee: anumite acțiuni/rezultate sunt controlate de o anumită incertitudine, nu neapărat de un adversar!

# Cazul cel mai defavorabil vs cazul mediu



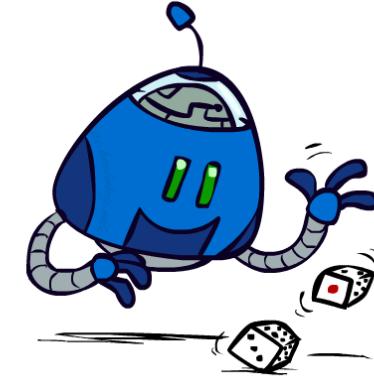
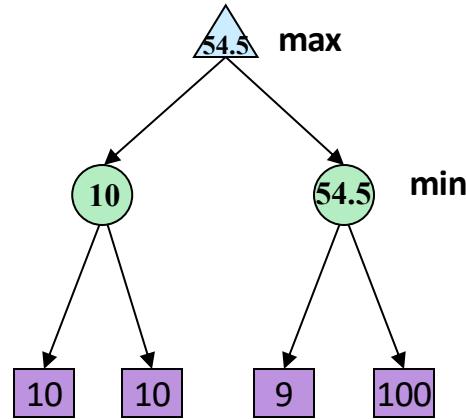
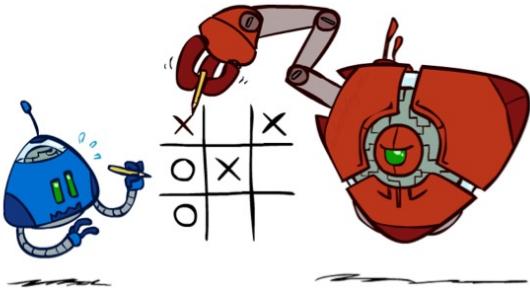
Idee: anumite acțiuni/rezultate sunt controlate de o anumită incertitudine, nu neapărat de un adversar!

# Cazul cel mai defavorabil vs cazul mediu



Idee: anumite acțiuni/rezultate sunt controlate de o anumită incertitudine, nu neapărat de un adversar!

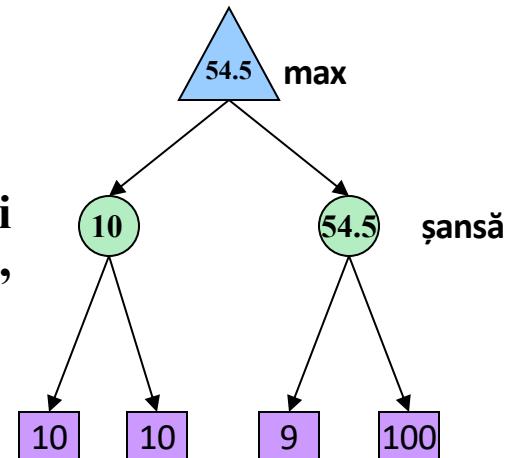
# Cazul cel mai defavorabil vs cazul mediu



Idee: anumite acțiuni/rezultate sunt controlate de o anumită incertitudine, nu neapărat de un adversar!

# Căutare ExpectiMax

- **De ce anumite acțiuni pot avea un rezultat imprevizibil:**
  - incertitudine în formă explicită: aruncarea unui zar
  - adversari impredictibili: pot avea un comportament aleator
  - anumite acțiuni pot eșua: un robot care execută anumite acțiuni poate avea roțile care alunecă
- **Valorile stărilor ar trebui să reflecte în aceste situații rezultatele/utilitățile/scorul pe cazul mediu (Expectation), nu pe cazul cel mai defavorabil (Min)**
- **Căutare ExpectiMax:** calculează scorul mediu pe baza unei strategii optime a jucătorului Max
  - noduri de tip max la fel ca în căutarea MiniMax
  - noduri de tip șansă sunt un fel de noduri de tip Min dar rezultatul/utilitatea/scorul este incert, fiind guvernăt de o distribuție de probabilitate
  - calculează **utilitățile medii** (media ponderată a nodurilor fii) – expectation (medie)
  - în lipsa unei distribuții de probabilitate asociate se poate calcula media nodurilor fii



# Implementare ExpectiMax

```
def value(stare):
```

dacă stare este o stare terminală: returnează utilitatea stării

dacă următorul agent care mută este MAX:  
returnează max-value(stare)

dacă următorul agent care mută este EXP: returnează exp-value(stare)

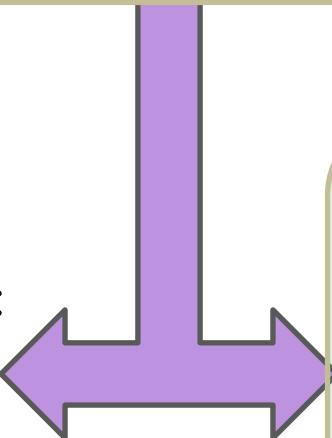
```
def max-value(stare):
```

initializează  $v = -\infty$

pentru fiecare successor al stării:

$$v = \max(v, \text{value(successor)})$$

returnează  $v$



```
def exp-value(state):
```

initializează  $v = 0$

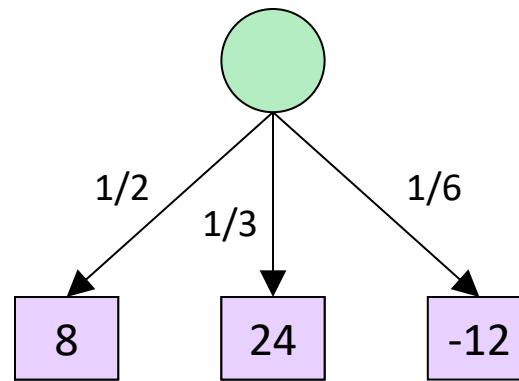
pentru fiecare succesor al stării:

$$v = v + p(\text{successor}) * \text{value}(\text{successor})$$

returnează  $v$

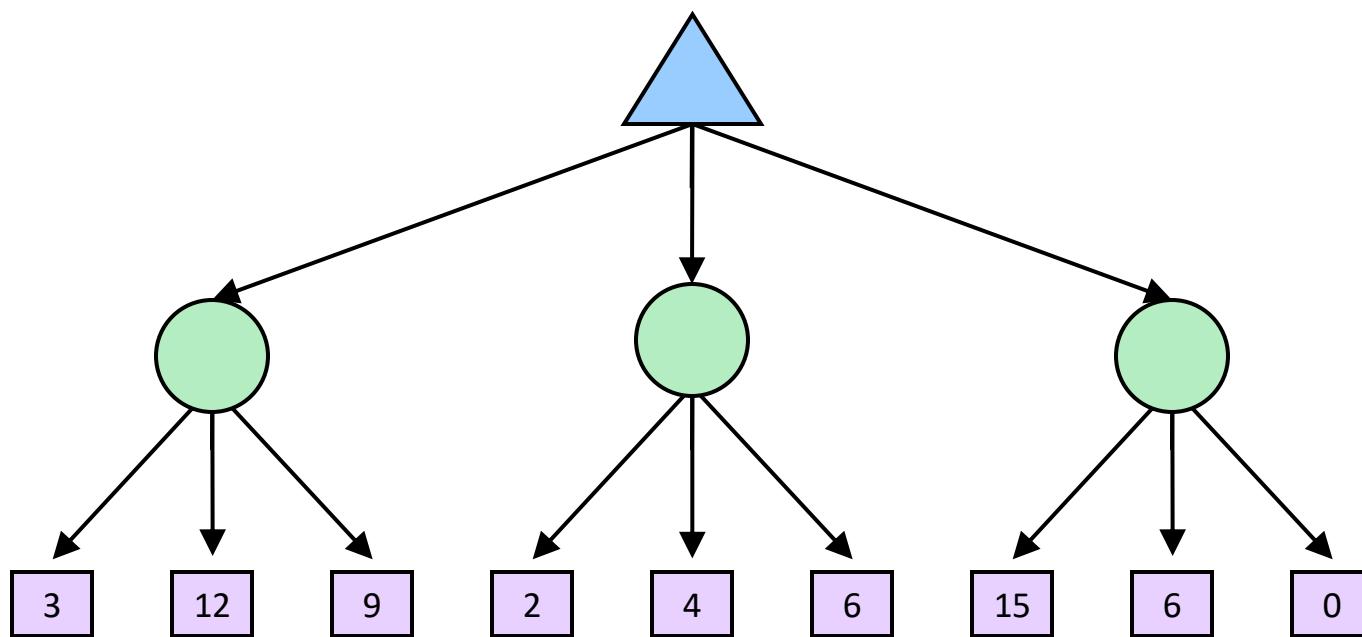
# Implementare ExpectiMax

```
def exp-value(state):  
    initializează v = 0  
    pentru fiecare succesor al stării:  
        v = v + p(successor) *  
            value(successor)  
    returnează v
```

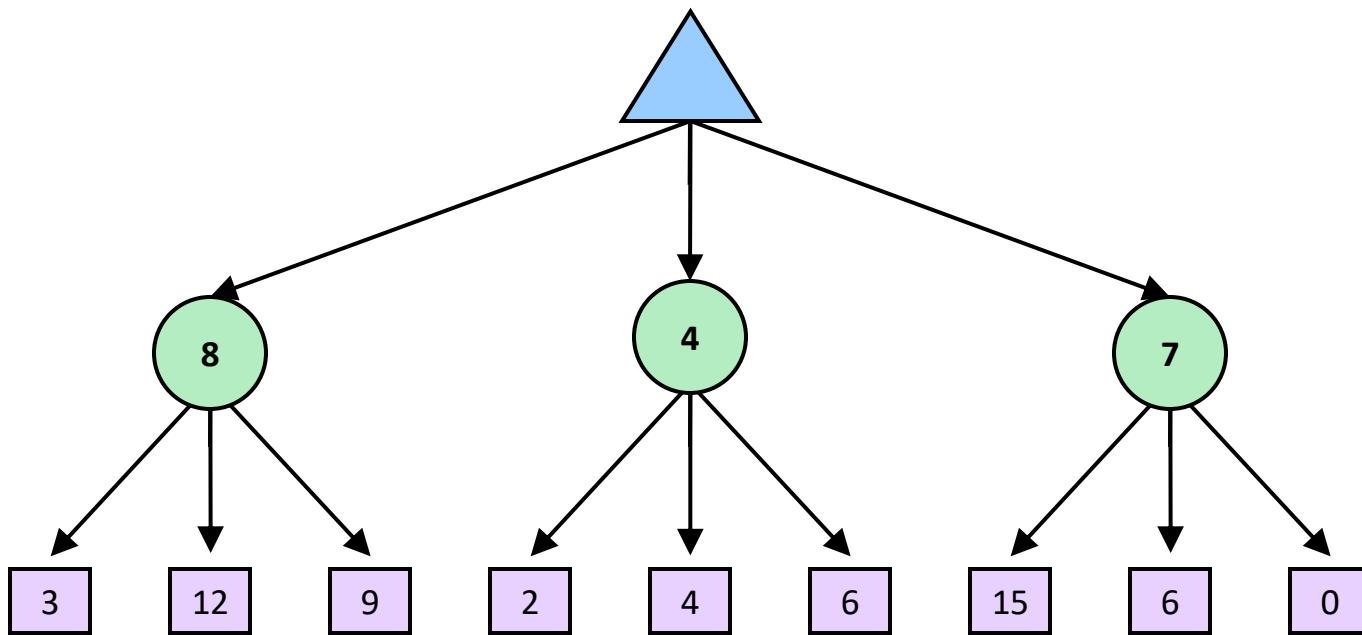


$$v = (1/2)(8) + (1/3)(24) + (1/6)(-12) = 10$$

# Exemplu ExpectiMax



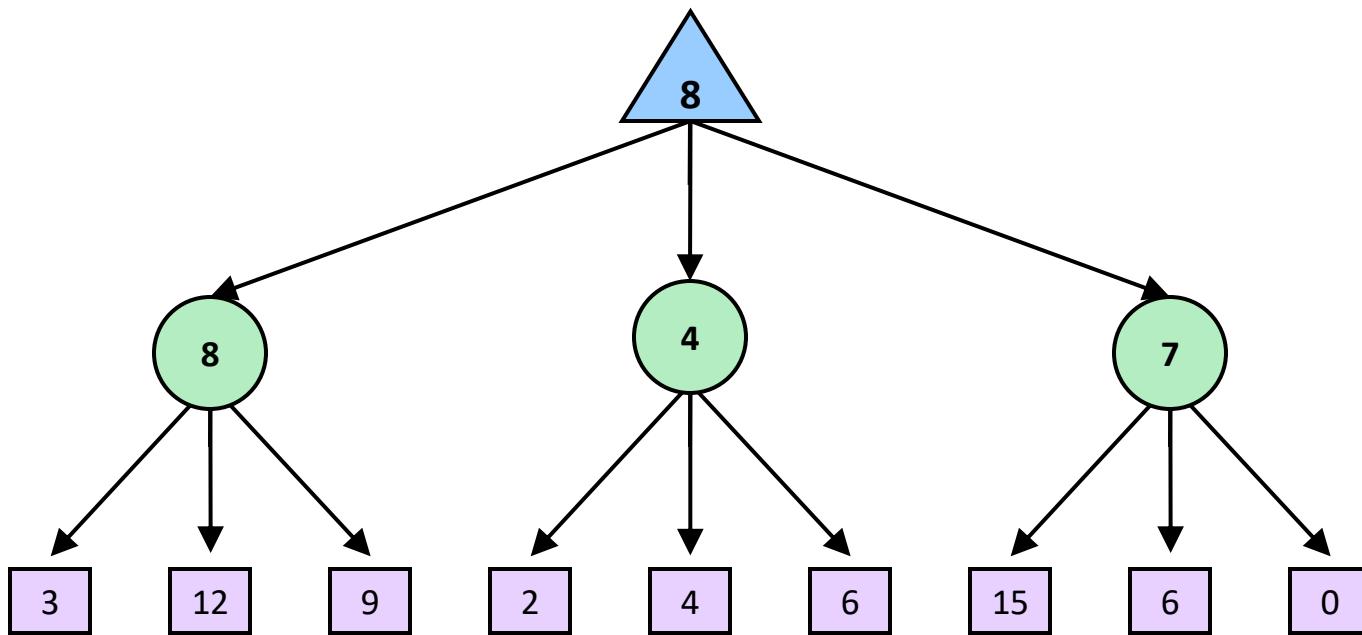
# Exemplu ExpectiMax



calculează **utilitățile medii** (media ponderată a nodurilor fii) – expectation (medie)

în lipsa unei distribuții de probabilitate asociate se poate calcula media nodurilor fii

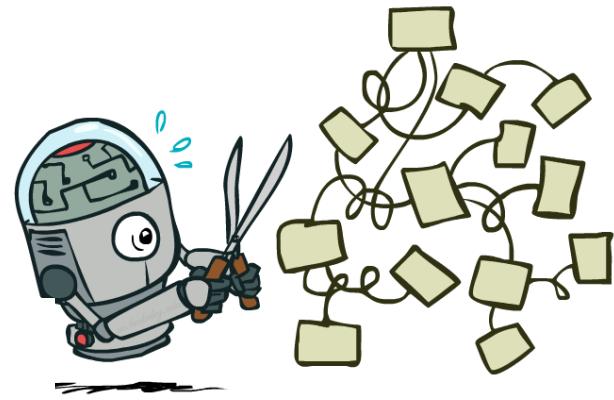
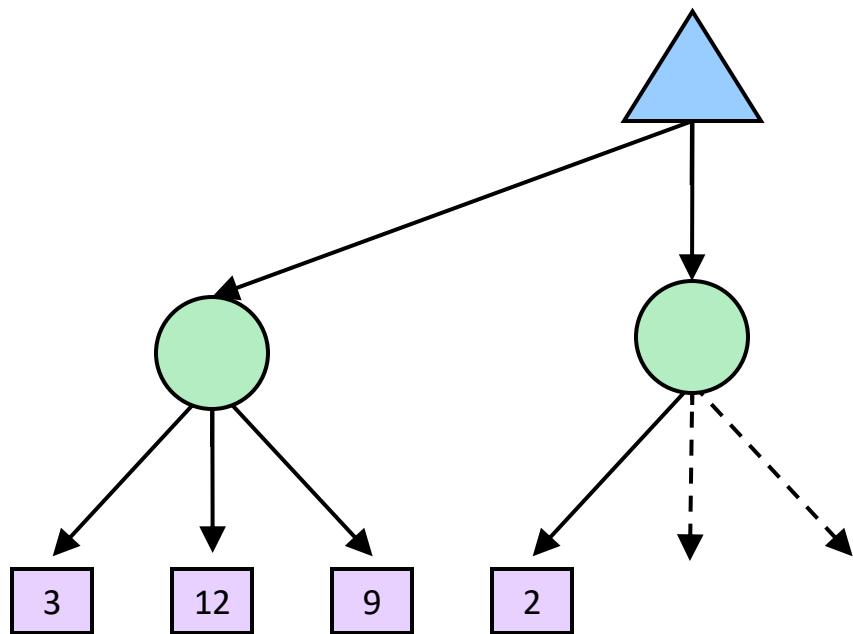
# Exemplu ExpectiMax



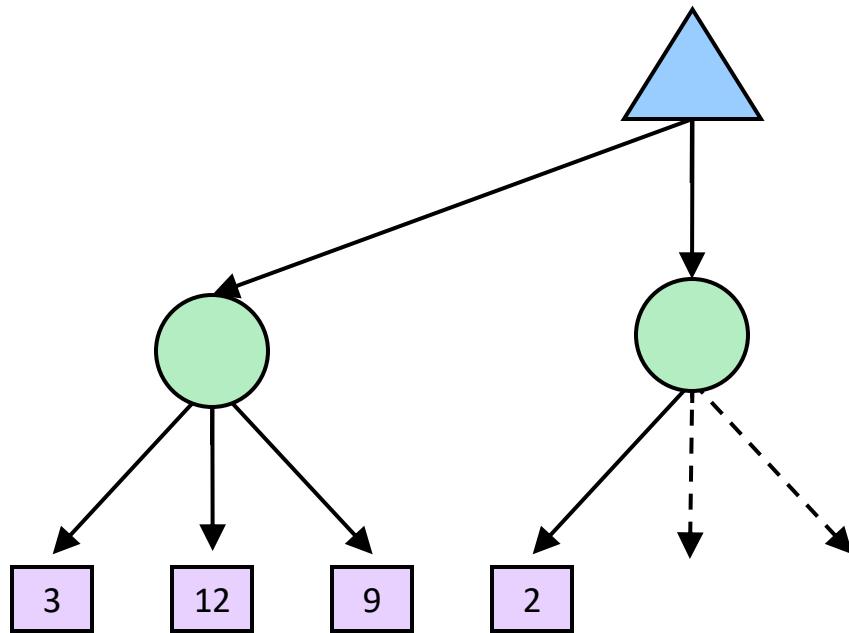
calculează **utilitățile medii** (media ponderată a nodurilor fii) – expectation (medie)

în lipsa unei distribuții de probabilitate asociate se poate calcula media nodurilor fii

# Accelerarea algoritmului ExpectiMax?

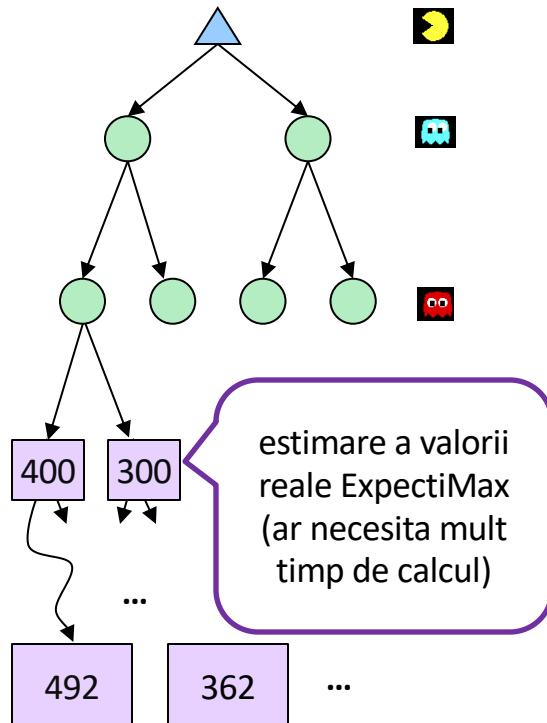


# Accelerarea algoritmului ExpectiMax?



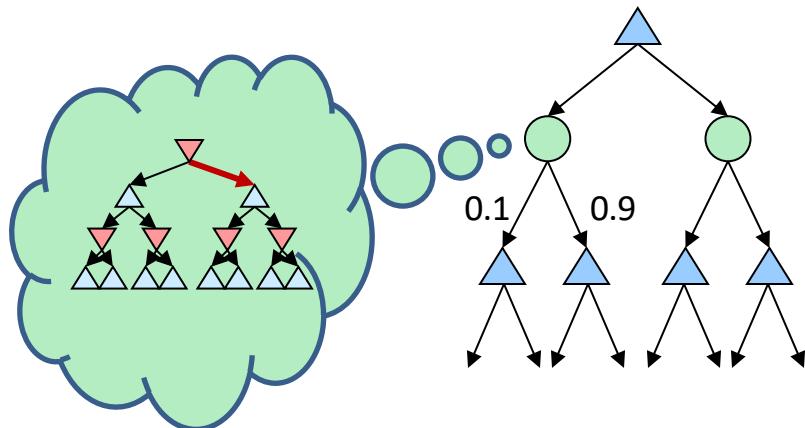
Diferit de algoritmul MiniMax care poate fi accelerat folosind alpha-beta retezare, nu putem accelera algoritmul ExpectiMax întrucât pot exista noduri la dreapta cu o valoare foarte mare care pot influența utilitatea medie.

# Accelerarea algoritmului ExpectiMax folosind căutare în adâncime limitată



# Exemplu ExpectiMax

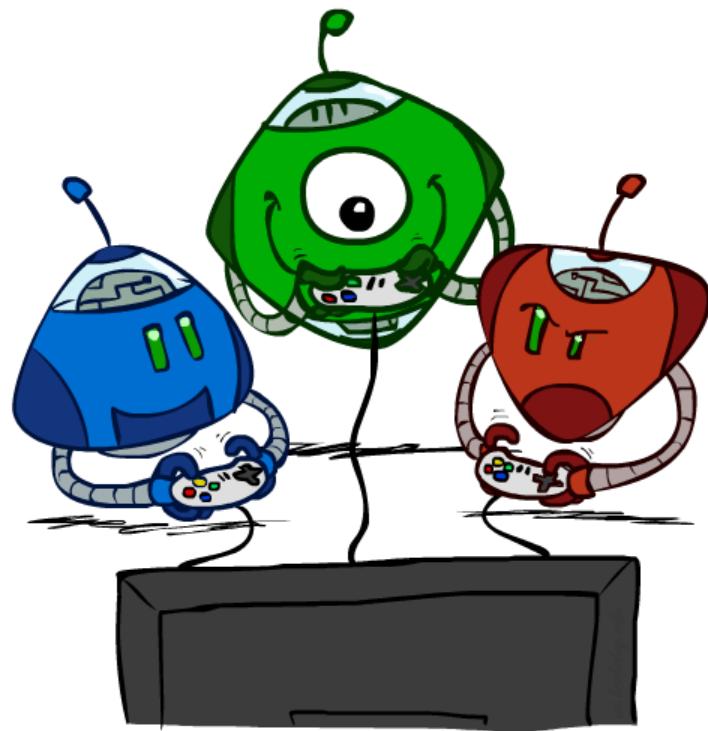
- Considerăm că știm strategia de joc a adversarului: în 80% din cazuri folosește o căutare minimax de nivel 2 (2 mutări) iar în 20% din cazuri mută aleator
- Întrebare: Ce fel de arbore de căutare folosim pentru modelarea problemei?



## ■ Răspuns: ExpectiMax!

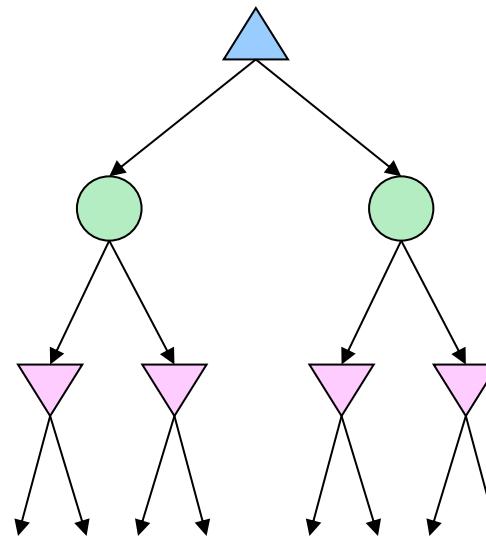
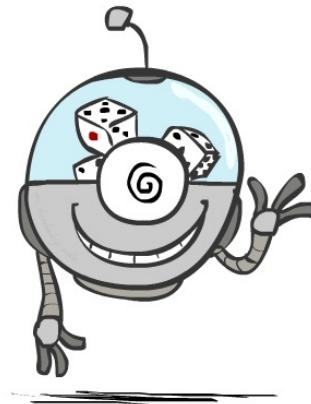
- Simulăm strategia adversarului în fiecare nod de tip sănsă și găsim probabilitățile asociate acțiunilor adversarului
- Foarte încet de calculat
- Putem simula că adversarul cunoaște strategia noastră – devine și mai încet
- ... spre deosebire de MiniMax, unde este suficient să calculăm o singură totul pentru a avea arborele minimax de joc

# Alte tipuri de jocuri



# Jocuri cu noduri mixte

- E.g. Backgammon (table)
- ExpectiMiniMax
  - Mediul (environment) este un “agent aleator” care mută (se dă cu zarul) după fiecare agent Min/Max
  - fiecare nod calculează utilitatea medie a fililor



# Exemplu: Backgammon (table)

- Aruncările cu zarul cresc factorul de ramificare  $b$ : 21 de posibilități cu 2 zaruri (1-1, 1-2, ..., 6-6)
  - Backgammon/Table  $\approx 20$  mutări legale
  - adâncime 2 =  $20 \times (21 \times 20)^3 = 1.2 \times 10^9$
- Pe măsură ce adâncimea crește, probabilitatea de a ajunge într-un anumit nod se micșorează
  - Utilitatea unei căutări se diminuează
  - Căutare limitată în adâncime nu pare aşa de rea
  - Accelerarea unei asemenea abordări este greu de realizat
- Istoria AI: TDGammon folosea o căutare de nivel 2 + funcție bună de evaluare + reinforcement learning: campion mondial la backgammon

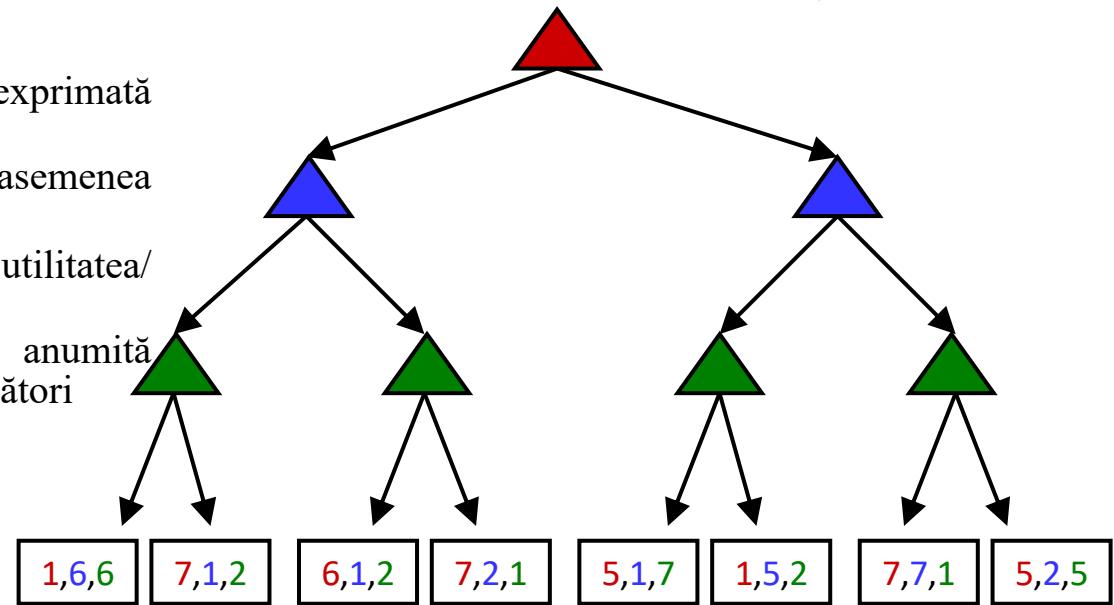
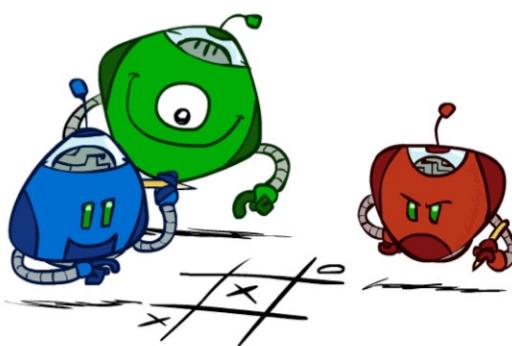
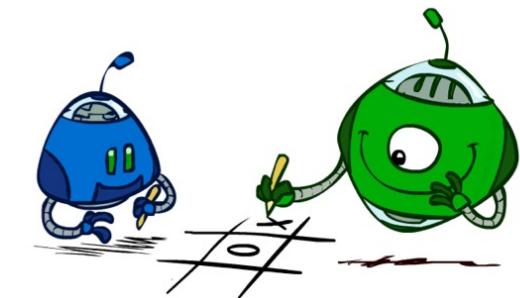


# Utilități multi-agent

- Dacă jocul nu este cu sumă zero sau are mai mult de doi jucători?

- Generalizare pentru MiniMax:

- Nodurile terminale au utilitatea exprimată printr-un tuplu
- Valorile nodurilor interne sunt de asemenea tupluri de utilitate
- Fiecare jucător maximizează utilitatea/ componenta proprie
- Poate apărea în mod dinamic o anumită cooperarea sau competiție între jucători

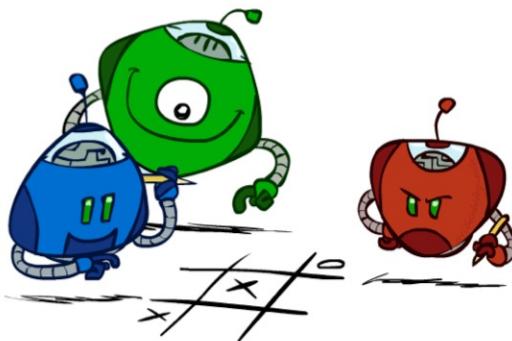
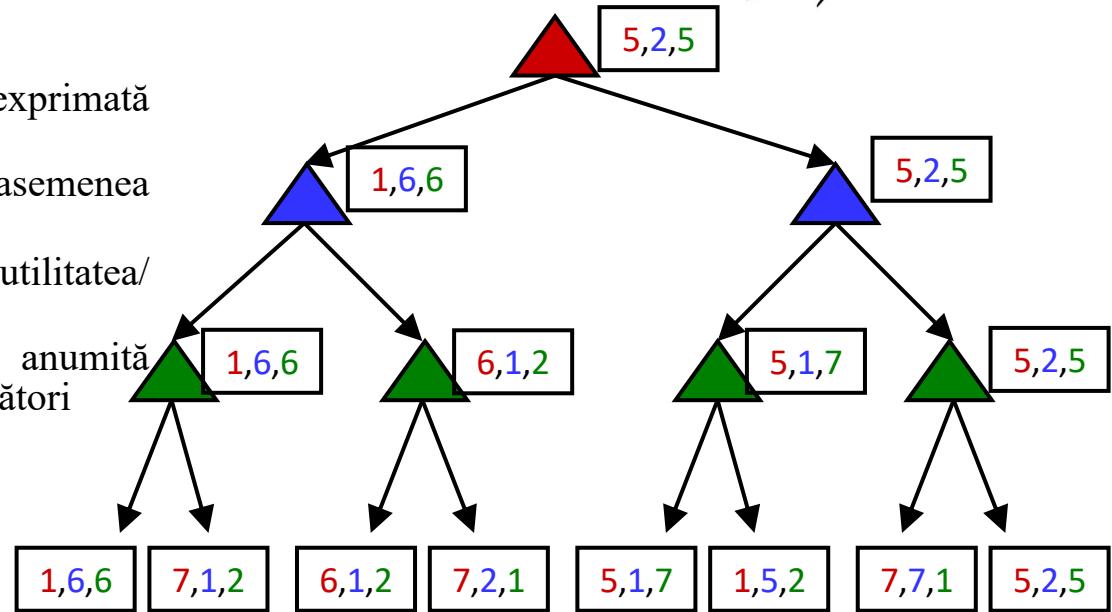
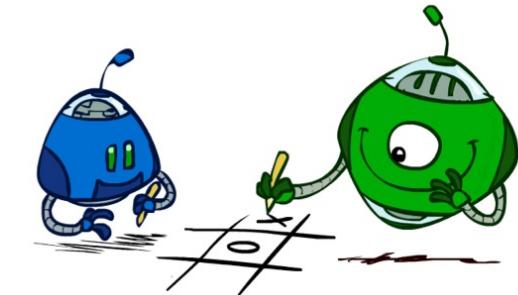


# Utilități multi-agent

- Dacă jocul nu este cu sumă zero sau are mai mult de doi jucători?

- Generalizare pentru MiniMax:

- Nodurile terminale au utilitatea exprimată printr-un tuplu
- Valorile nodurilor interne sunt de asemenea tupluri de utilitate
- Fiecare jucător maximizează utilitatea/ componenta proprie
- Poate apărea în mod dinamic o anumită cooperarea sau competiție între jucători



# Cursul de azi

## 1. Căutare în jocuri cu incertitudine:

- căutare ExpectiMax

## 2. Examenul final

- organizarea examenului din acest an
- subiectele din anii trecuți
- rezolvările subiectelor din anii trecuți

# Examen iarnă - evaluare și notare

Nota = min(Curs + Laborator + Proiect&Teme + BonusLab, 10)  
4p                3p                3p                maxim 1p

- Examen Curs (4 puncte) – scris, în sesiune, 2 ore + eventual bonus de la curs (nu puteți depăși 4p)
- Test Laborator (3 puncte) – în sesiune (în aceeași zi cu Examen Curs), 2 ore;
- Proiect (1,5 puncte) – proiect la prima parte (învățare automată, îl primiți în săptămâna 4), termen limită de predare – săptămâna 6, prezentarea proiectului în săptămâna 7.
- Teme (1,5 puncte) – teme la partea a doua (căutare informată și neinformată), prezentare în cadrul laboratorului.

Nu există praguri, note minime impuse. 4,99 înseamnă restanță.

# Organizarea examenului din acest an

- am propus zilele de 31 ianuarie, 1 sau 2 februarie:
  - recomandarea mea e să dăm ambele probe în aceeași zi
  - intre 9-11 test de laborator
  - intre 13-15 scris în amfiteatru
- Testul de laborator:
  - veți primi un subiect cu două probleme (una din prima parte – învățare automată, una din a doua parte – căutare informată și neinformată)
  - aveți voie cu un stick cu resurse
  - nu aveți voie cu laptop, toată lumea dă examen pe calculatoarele din FMI
  - acces la internet restricționat

# Organizarea examenului din acest an

- am propus zilele de 31 ianuarie, 1 sau 2 februarie:
  - recomandarea mea e să dăm ambele probe în aceeași zi
  - intre 9-11 test de laborator
  - intre 13-15 scris în amfiteatru
- Partea scrisă:
  - lucrare scrisă individual
  - două posibilități:
    - de tip open books (cu materialele tipărite pe bancă, fără dispozitive electronice)
    - de tip closed books (fără niciun fel de materiale) – 9 intrebari/probleme din toata materia

# Cursul de azi

## 1. Căutare în jocuri cu incertitudine:

- căutare ExpectiMax

## 2. Examenul final

- organizarea examenului din acest an
- subiectele din anii trecuți
- rezolvările subiectelor din anii trecuți

# Subiectul din 27 ianuarie 2020

## Examen la disciplina Inteligență Artificială Seria 35, 27 ianuarie 2020

### Subiectul 1. (3,5 puncte)

Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

- considerăm perceptronul din Figura 1a cu ponderile  $w_1$  și  $w_2$  care ponderează intrările  $x_1$  (abscisa) și  $x_2$  (ordonata) a unui punct, iar ponderea  $w_0$  reprezintă deplasarea (bias-ul). Funcția de activare a perceptronului este funcția  $\phi(x) = \text{hardlim}(x)$ , unde  $\text{hardlim}(x) = 1$ , dacă  $x \geq 0$  și  $\text{hardlim}(x) = 0$  dacă  $x < 0$ . Determinați matricea de confuzie a perceptonului cu ponderile  $w_0 = -2, w_1 = 1, w_2 = 1$  care clasifică exemple din mulțimea  $S$ . **(1 punct)**
- mulțimea de antrenare  $S$  nu este liniar separabilă. Totuși, dacă adăugăm la fiecare vector a treia coordonată, egală cu eticheta exemplului, obținem o nouă mulțime de antrenare care acum este liniar separabilă. Arătați acest lucru, dând exemplu de un clasificator liniar care separă perfect exemplele de antrenare din noua mulțime. Puteți folosi acest clasificator liniar pentru a eticheta exemple din mulțimea  $T$ ? **(1 punct)**
- construiți o rețea neuronală de perceptri care să învețe perfect mulțimea de antrenare. Etichetați punctele din mulțimea  $T$  folosind rețeaua astfel construită. **(1,5 puncte)**

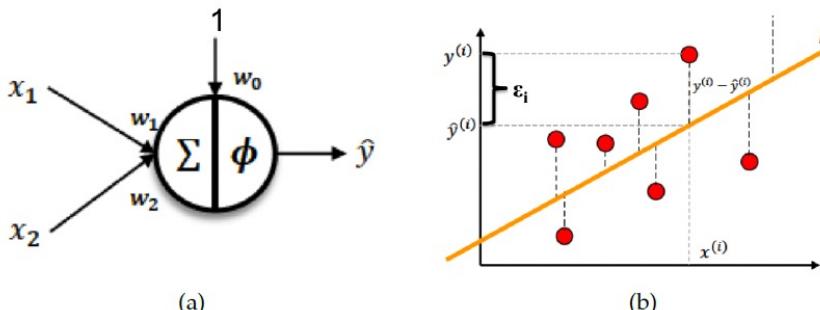
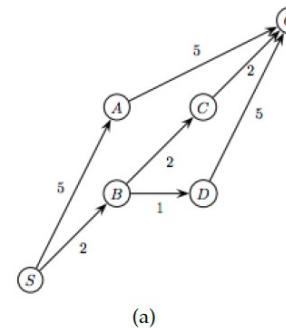


Figura 1: a. Arhitectura unui perceptron; b. Problema de regresie liniară prezentată la curs și evidențierea reziduuului  $\epsilon_i$ .



Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0, h_1, h_2$ .

### Subiectul 2. (1 punct)

Fie  $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$  o mulțime cu  $m$  exemple de antrenare, unde  $x^{(i)}, y^{(i)}$  sunt numere reale pentru  $i = 1, \dots, m$ . Regresia liniară simplă modelează legătura dintre variabila independentă  $x$  și variabila dependentă  $y$  folosind dreapta de ecuație  $h(x) = w_0 + w_1 \cdot x$ . Parametrii  $w_0$  și  $w_1$  se determină folosind tehnica metodei celor mai mici pătrate aplicată pe mulțimea de antrenare  $E$ . La curs a fost arătat că parametrii optimi au valorile următoare:

$$w_0 = \frac{1}{m} \sum_{i=0}^m y^{(i)} - w_1 \cdot \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$w_1 = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2},$$

unde  $\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$ , iar  $\bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$ .

Pentru fiecare exemplu de antrenare din  $E$  notăm cu  $\epsilon_i$  diferența dintre răspunsul corect  $y^{(i)}$  și predicția  $\hat{y}^{(i)}$  (vedeți acest lucru ilustrat și în Figura 1b). Cantitatea  $\epsilon_i$  se mai numește reziduu. Arătați că suma tuturor reziduurilor este nulă, adică:

$$\sum_{i=1}^m \epsilon_i = 0.$$

### Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a. S este starea inițială iar G este starea scop. O strategie de căutare care pornește din starea S și vrea să ajungă la starea G explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea A va fi explorată înaintea stării B). Considerăm euristicile  $h_0, h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (a) care este soluția problemei de căutare folosind strategiile de căutare în lățime și căutare în adâncime? **(0,5 puncte)**
- (b) care din euristicile  $h_0, h_1, h_2$  din Figura 2b sunt admisibile? Justificați răspunsul **(0,5 puncte)**
- (c) care este soluția returnată de algoritmul  $A^*$  pentru fiecare heuristică  $h_0, h_1, h_2$  în parte? Specificați soluțiile parțiale construite până la obținerea soluției finale. **(0,5 puncte)**
- (d) care este soluția returnată de algoritmul Greedy ce folosește heuristică  $h_1$ ? Specificați soluțiile parțiale construite până la obținerea soluției finale. **(0,5 puncte)**
- (e) care este soluția returnată de algoritmul de căutare uniformă după cost (UCS)? Specificați soluțiile parțiale construite până la obținerea soluției finale. **(0,5 puncte)**

#### **Subiectul 4. (2 puncte)**

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3b). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Căștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (a) Care este numărul de mutări maxim (= factorul de ramificare) pe care îl are la dispoziție fiecare jucător în orice moment al jocului? Justificați răspunsul. **(0,25 puncte)**
- (b) Considerăm că după 9 mutări ajungem cu jocul în starea dată de Figura 3b. Jucătorul 0 este acum la mutare. Desenați arborele de joc asociat, considerând drept rădăcină starea actuală a jocului. **(0,75 puncte)**
- (c) considerăm jucătorul X ca fiind MAX și jucătorul 0 ca fiind MIN. Funcția de utilitate asociată nodurilor terminale este următoarea: dacă jucătorul X câștigă atunci el

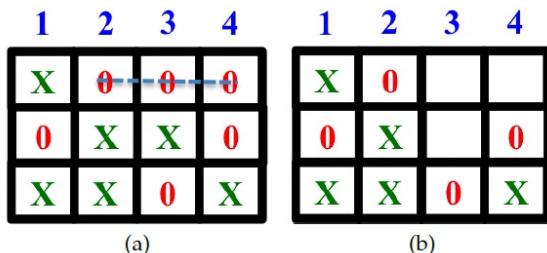


Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea  $-1$  (căștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

primește  $k$  puncte iar jucătorul 0 pierde  $k$  puncte, dacă jocul se termină la egalitate atunci fiecare are 0 puncte. Valoarea  $k$  atunci când jucătorul X câștigă este stabilită astfel:  $k = 3$  puncte pentru o linie realizată din simboluri dispuse pe diagonală,  $k = 2$  pentru o linie realizată din simboluri dispuse pe verticală și  $k = 1$  pentru o linie realizată din simboluri dispuse pe orizontală. Valoarea  $k$  atunci când jucătorul 0 câștigă pe cazurile enumerate anterior este  $-3, -2, -1$ . Folosind algoritmul Minimax etichetați arborele de joc de la punctul anterior cu valorile minimax asociate fiecărui nod. **(0,5 puncte)**

- (d) Folosiți algoritmul de retezare Alfa-Beta pentru accelerarea calcului valorilor Minimax. Ce noduri din arbore de joc nu vor fi explorate? Puteți reordona fiile fiecărui subarbore pentru a maximiza numărul de noduri care nu vor fi explorate de algoritmul Alfa-Beta? Justificați răspunsul. **(0,5 puncte)**

**Oficiu: 1 punct. Timp de lucru: 2 ore.**

# Subiectul din 1 februarie 2022 (pe Moodle)

1 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

2 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Consideram problema clasificarii literelor mici (clasele ‘a’, ‘b’, …, ‘z’) scrise de mana din alfabetul englez (in total 26 de clase) pe baza modelului Naïve Bayes. Datele de antrenare constau in 100 de exemple de antrenare cu imagini 28 x 28 pixeli pentru fiecare litera in parte. In total aveti 2600 exemple de antrenare. Consideram ca stim probabilitatea a-priori de aparitie a fiecarei litere pe baza studiului frecventei de aparitie a literelor in limba engleza. Pentru estimarea probabilitatii likelihood folosim modelul Naïve Bayes care considera ca fiecare pixel din imagine are o distributie normala specifica pozitiei pixelului din imagine si clasei literei.

1. cati parametri trebuie sa invete modelul vostru pentru a putea clasifica o litera 28 x 28?
2. cum ati invata parametri corespunzatori pixelului de pe linia 14, coloana 14 (centrul imaginii) din clasa ‘a’?

3 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

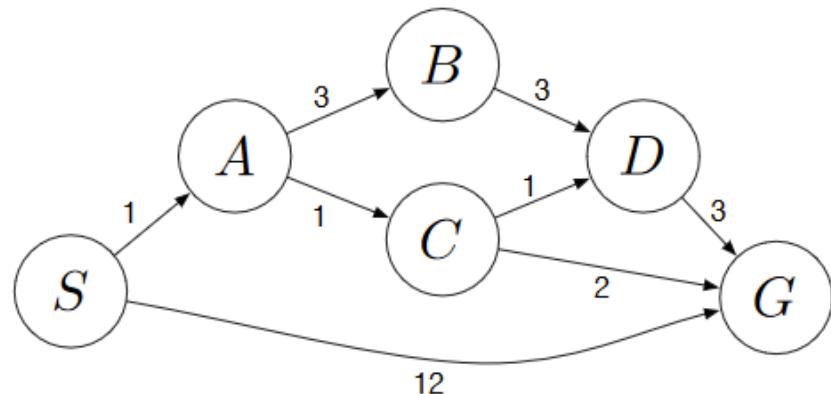
Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

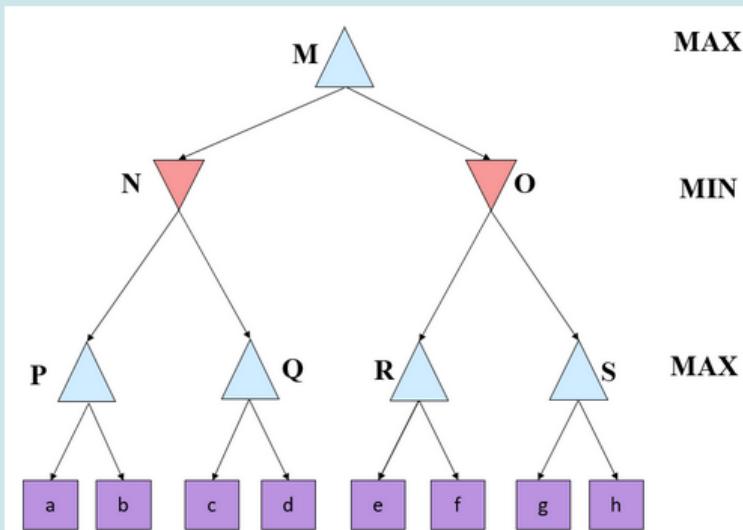
Consideram problema de cautare data de graful din figura de mai jos. S este starea initiala, G este starea scop. O strategie de cautare care porneste din starea S si vrea sa ajunga la starea G exploreaza starile intr-o anumita ordine. Implicit, se prefera explorarea starilor in ordine lexicografica. Realizati urmatoarele:

1. Care este solutia problemei de cautare folosind strategia de cautare in latime? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.
2. Care este solutia problemei de cautare folosind strategia de cautare uniforma dupa cost? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.

(Cand raspundeti la intrebari folositi ca notatie pentru o solutie partiala urmatoare notatie: S-A-D-G - solutie partiala invalida pentru cazul de fata)



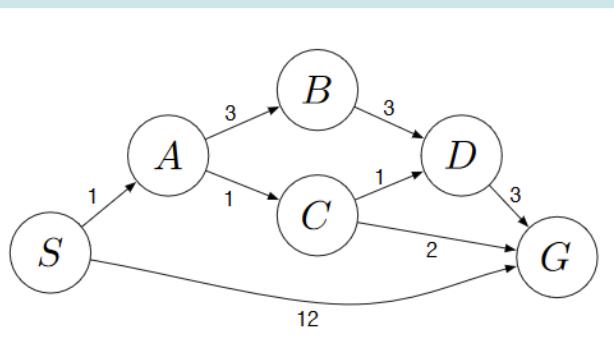
Consideram arborele de cautare din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN vrea sa minimizeze castigul lui MAX. Nodurile terminale au valorile a = 5, b = -10, c = 10, d = -2, e = 1, f = -1, g = 15, h = -1. Care sunt valorile Minimax ale nodurilor M, N, O, P, Q, R, S?



Consideram problema de cautare data de graful din figura de mai jos. S este starea initiala, G este starea scop. O strategie de cautare care porneste din starea S si vrea sa ajunga la starea G exploreaza starile intr-o anumita ordine. Implicit, se prefera explorarea starilor in ordine lexicografica (la costuri egale). Consideram euristicile  $h_1$  si  $h_2$  din tabelul de mai jos. Realizati urmatoarele:

1. Care din euristicile  $h_1$  si  $h_2$  este admisibila? Justificati raspunsul.
2. Care este solutia problemei de cautare folosind algoritmul A\* cu euristica  $h_1$ ? Specificati solutiile pariale construite in frontiera pana la obtinerea solutiei finale.

(Cand raspundeti la intrebari folositi ca notatie pentru o solutie parciala urmatoare notatie: S-A-D-G - solutie invalida pentru cazul de fata)



State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

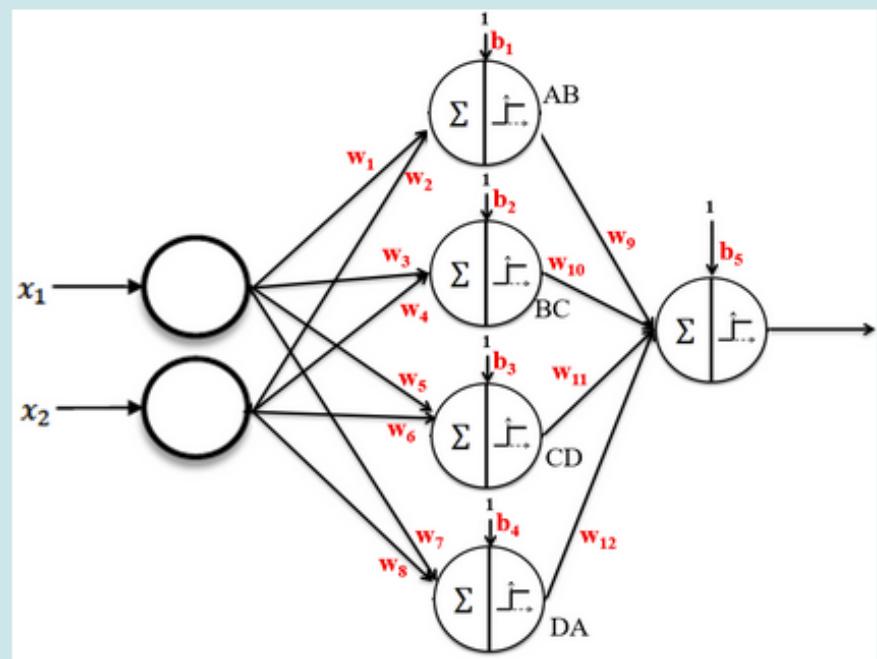
7 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram patratul de varfuri A(10,10), B(10,-10), C(-10,-10), D(-10,10) si reteaua neuronală din figura de mai jos care dorește să implementeze funcția indicator a patratului ABCD după logica descrisă.



8 intrebare

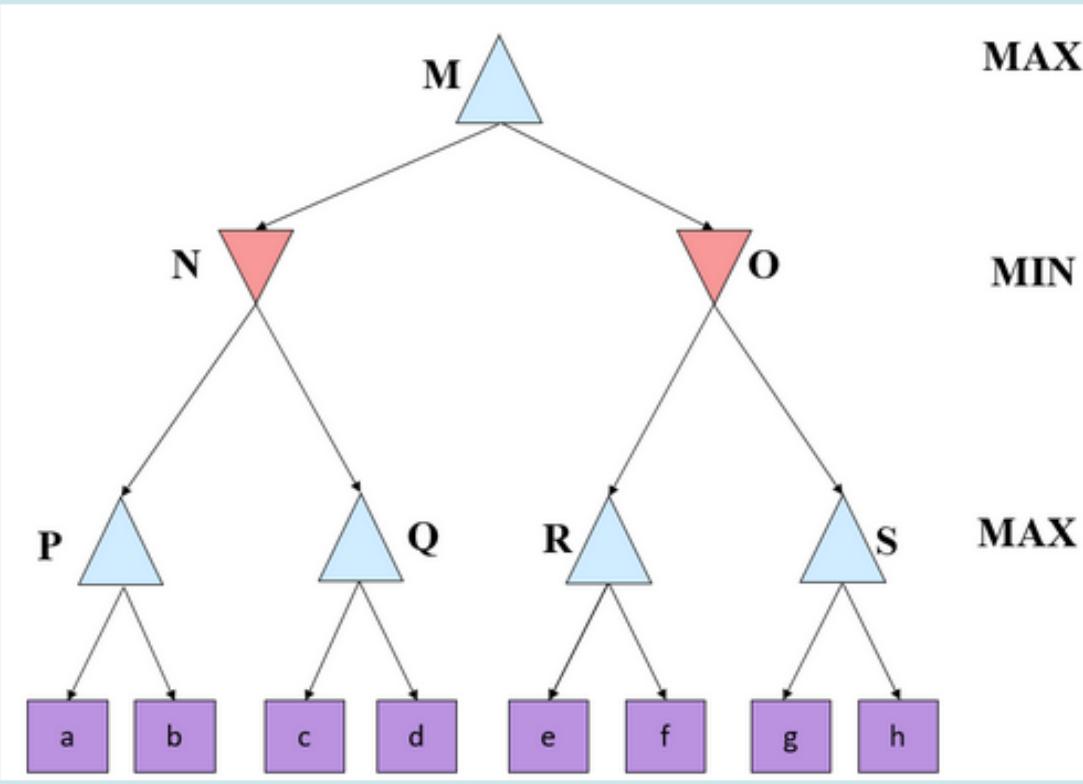
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram problema regresiei liniare simple pe multimea de antrenare  $S = \{(3,0), (4,2), (6,0), (7,2)\}$ . Notăm cu  $w = (w_0, w_1)$  astfel încât  $y = w_0 + w_1 * x$  este dreapta soluție a regresiei liniare. Cât este  $w$ ?

Consideram arborele de joc din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN este responsabil sa minimizeze castigul lui MAX. Aplicati algoritmul alfa-beta in calcul valorii Minimax a radacinii M. Ce noduri sunt retezate (nu sunt vizitate)? Justificati raspunsul.



10 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram o problema de clasificare binara cu puncte din  $\mathbb{R}$  (dreapta reala). Fie multimea de exemple de antrenare  $P = \{(-1,1), (0,0), (1,0), (2,0), (3,1), (4,1)\}$  unde fiecare exemplu de antrenare e reprezentat de coordonata sa = abscisa si de eticheta corespunzatoare (0 sau 1). Scrieti ponderile  $w_0$  (biasul),  $w_1$  (ponderea pentru abscisa) a unui perceptron cu functia de activare hardlim care obtine acuratetea cea mai mare pe multimea P.

# Cursul de azi

## 1. Căutare în jocuri cu incertitudine:

- căutare ExpectiMax

## 2. Examenul final

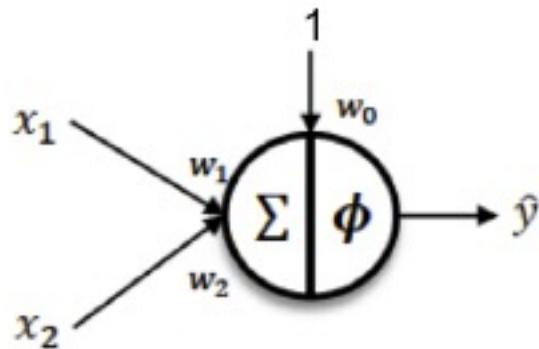
- organizarea examenului din acest an
- subiectele din anii trecuți
- rezolvările subiectelor din anii trecuți

# Subiectul 1 a - enunt

## Subiectul 1. (3,5 puncte)

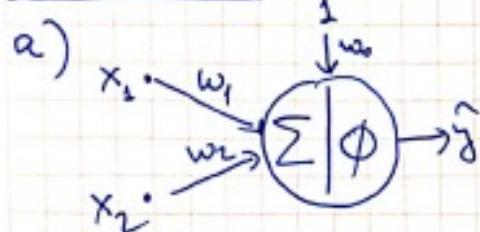
Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

- (a) considerăm perceptronul din Figura 1a cu ponderile  $w_1$  și  $w_2$  care ponderează intrările  $x_1$  (abscisa) și  $x_2$  (ordonata) a unui punct, iar ponderea  $w_0$  reprezintă deplasarea (bias-ul). Funcția de activare a perceptronului este funcția  $\phi(x) = \text{hardlim}(x)$ , unde  $\text{hardlim}(x) = 1$ , dacă  $x \geq 0$  și  $\text{hardlim}(x) = 0$  dacă  $x < 0$ . Determinați matricea de confuzie a perceptronului cu ponderile  $w_0 = -2$ ,  $w_1 = 1$ ,  $w_2 = 1$  care clasifică exemple din mulțimea  $S$ . (1 punct)



(a)

### Subiectul 1



$$\hat{y} = \text{hardlim}(w_0 + w_1 x_1 + w_2 x_2) =$$

$$= \begin{cases} 1, & w_0 + w_1 x_1 + w_2 x_2 \geq 0 \\ 0, & \text{altele} \end{cases}$$

$$w_0 = -2, w_1 = 1, w_2 = 1$$

$$\hat{y} = \begin{cases} 1, & x_1 + x_2 - 2 \geq 0 \\ 0, & x_1 + x_2 - 2 < 0 \end{cases}$$

$$x_1 = 0,$$

$$x_2 = 0,$$

$$y = 0$$

$$\hat{y} = 0$$

$$x_1 = 1$$

$$x_2 = 0$$

$$y = 1$$

$$\hat{y} = 1$$

$$x_1 = 2$$

$$x_2 = -1$$

$$y = -1$$

$$\hat{y} = 0$$

$$x_1 = 2$$

$$x_2 = 2$$

$$y = 1$$

$$\hat{y} = 1$$

$$x_1 = 5$$

$$x_2 = 3$$

$$y = 0$$

$$\hat{y} = 1$$

$$x_1 = -2$$

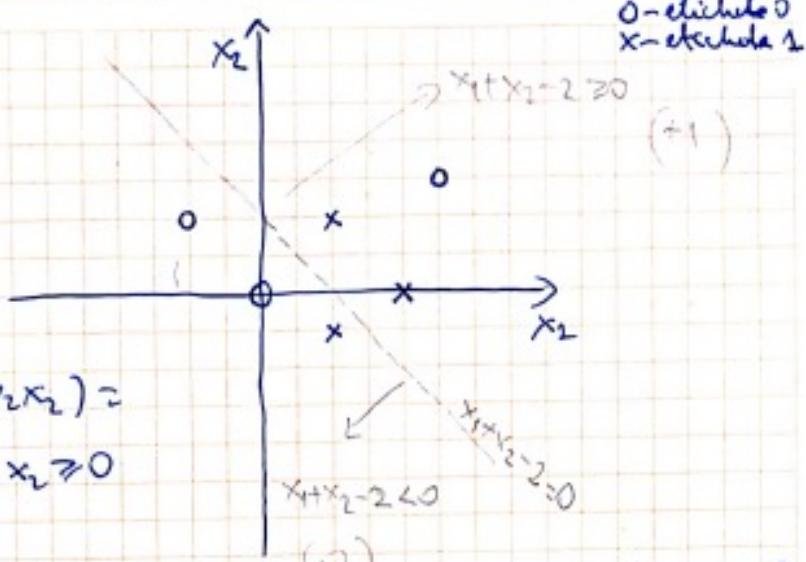
$$x_2 = 2$$

$$y = 0$$

$$\hat{y} = 0$$

elicitate  
proba

elicitate reală	$\hat{y}$	0	1
0	2	1	
1	1	2	



0-elicitate 0  
x-elicitate 1

$$x_1 + x_2 - 2 \geq 0$$

(+1)

$$x_1 + x_2 - 2 \leq 0$$

(0)

$$S = \{(1, 0)^T, 0\}, \{(4, 0)^T, 1\}, \dots, \{(-2, 2)^T, 0\}\}$$

$$T = \{(1, 1)^T, (5, 5)^T\}$$

Natura de conținut asociată perceptronului

# Subiectul 1b - enunt

## Subiectul 1. (3,5 puncte)

Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

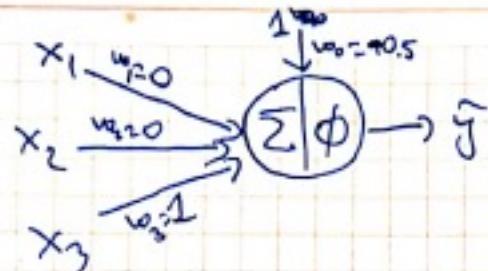
- (b) mulțimea de antrenare  $S$  nu este liniar separabilă. Totuși, dacă adăugăm la fiecare vector a treia coordonată, egală cu eticheta exemplului, obținem o nouă mulțime de antrenare care acum este liniar separabilă. Arătați acest lucru, dând exemplu de un clasificator liniar care separă perfect exemplele de antrenare din noua mulțime. Puteți folosi acest clasificator liniar pentru a eticheta exemple din mulțimea  $T$ ? (1 punct)

# Subiectul 1b - rezolvare

b)  $S_{\text{misi}} = \{( (9,0,0)^T, 0 ), ( (-4,0,1)^T, 1 ), ( (2,-1,1)^T, 1 ), ( (2,2,1)^T, 1 ), ( (5,3,0)^T, 0 ), ( (-2,2,0)^T, 0 ) \}$

Adaugăm practic dimensiunea  $x_3 = \text{înălțime}$ . Trăiești parțialele  
poziții palmei înălțimea 1, restul are înălțimea 0. Multimea multime  
de antrenare este liniștită separabilă, există o funcție de hiperplanuri  
(plane) care separă exemplul pozitiv de cel negativ.

O posibilitate este  ~~$w_0=0, w_1=2, w_2=0, w_3=1$~~ ,  $w_0=-0.5$ ,  $w_1=1$ :  $x_3 - 0.5 \geq 0$



$$\hat{y} = \text{hardim}(x_3 - 0.5) = \begin{cases} 1, & x_3 \geq 0.5 \\ 0, & x_3 < 0.5 \end{cases}$$

Ne putem folosi acest clasificator pentru a stabili exemple din  
multimea T între care lipsește o treia componentă (care reprezintă  
etichete pe care vrem să le prevedem).

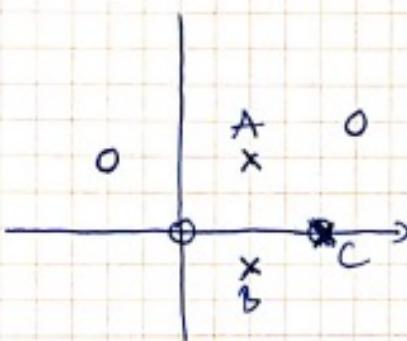
# Subiectul 1c - enunt

## Subiectul 1. (3,5 puncte)

Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

- (c) construiți o rețea neuronală de perceptroni care să învețe perfect mulțimea de antrenare. Etichetați punctele din mulțimea  $T$  folosind rețeaua astfel construită. (1,5 puncte)

c)

 $A(2, 2), B(-2, -2), C(4, 0)$ 

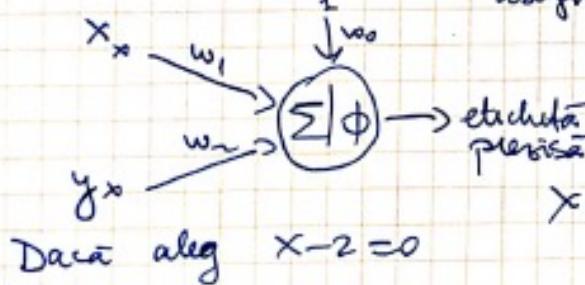
$$AB: \frac{x-x_A}{y-y_A} = \frac{x_B-x_A}{y_B-y_A}$$

$$\frac{x-2}{y-2} = \frac{-2-2}{-1-2} \Leftrightarrow x=2.$$

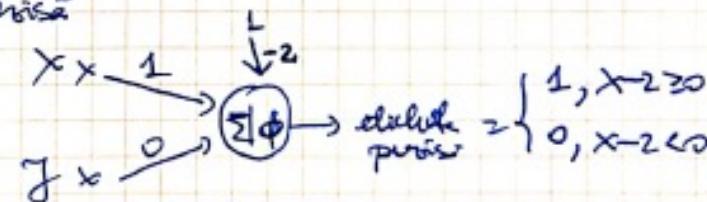
$$x-2=0.$$

Pentru dreapta  $AB$  am doar două ecuații  $\begin{cases} x-2=0 \\ -x+2=0 \end{cases}$ .

Pe care v-aleg? (soluția mea presupune a constura o rețea care asigură 1 punct pe tot perimetrul din triunghiul  $ABC$ ).

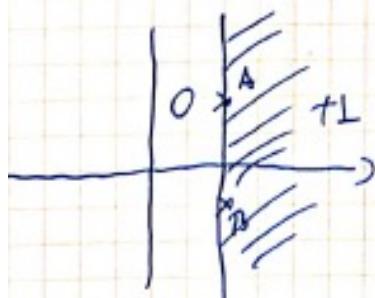


$$\text{Dacă aleg } x-2=0$$

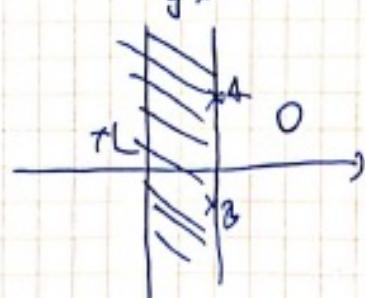


$$\text{Dacă aleg } -x+2=0$$

$$\begin{aligned} & \text{Inputs: } x \rightarrow 1, y \rightarrow 0 \rightarrow \Sigma\phi \rightarrow \text{efectuare} = \begin{cases} 1, & x-2 \geq 0 \\ 0, & x-2 < 0 \end{cases} \\ & \text{Inputs: } x \rightarrow -1, y \rightarrow 0 \rightarrow \Sigma\phi \rightarrow \text{efectuare} = \begin{cases} 1, & -x+2 \geq 0 \\ 0, & -x+2 < 0 \end{cases} \end{aligned}$$



Cazul  $x-2 \geq 0$



Cazul  $-x+2 \geq 0$

# Subiectul 1c - rezolvare

Vom se prelua punctul C(4,0) și vom lăsa eluată L. Astăzi vom lucra  
cu  $x+2y=0$

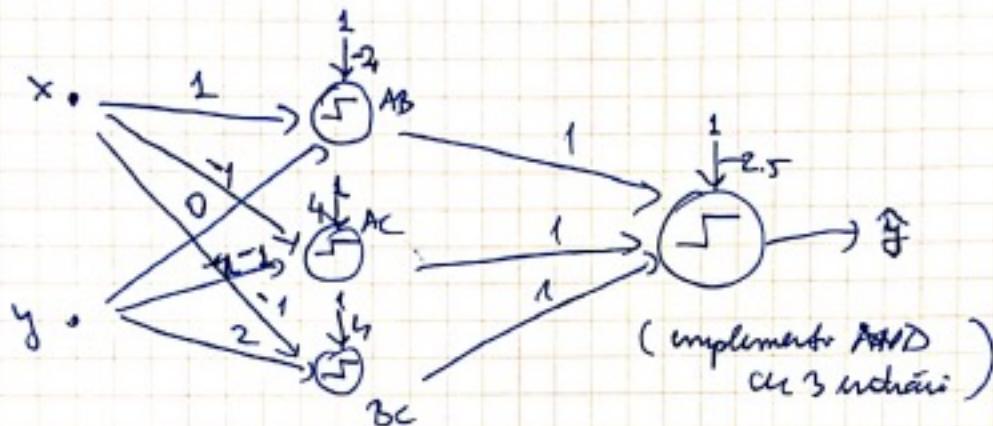
Analog, obținem ecuațiile partii dreptele AC și BC

$$AC: x+y-4=0$$

$$\boxed{-x-y+4=0}$$

$$BC: x-2y-4=0$$

$$\boxed{-x+2y+4=0}$$



$T = \{(1,1)^T, (5,5)^T\} : \text{Ambele puncte vor fi etablate ca } 0.$

# Subiectul 2 - enunt

## Subiectul 2. (1 punct)

Fie  $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$  o mulțime cu  $m$  exemple de antrenare, unde  $x^{(i)}, y^{(i)}$  sunt numere reale pentru  $i = 1, \dots, m$ . Regresia liniară simplă modelează legătura dintre variabila independentă  $x$  și variabila dependentă  $y$  folosind dreapta de ecuație  $h(x) = w_0 + w_1 \cdot x$ . Parametrii  $w_0$  și  $w_1$  se determină folosind tehnica metodei celor mai mici pătrate aplicată pe mulțimea de antrenare  $E$ . La curs a fost arătat că parametri optimi au valorile următoare:

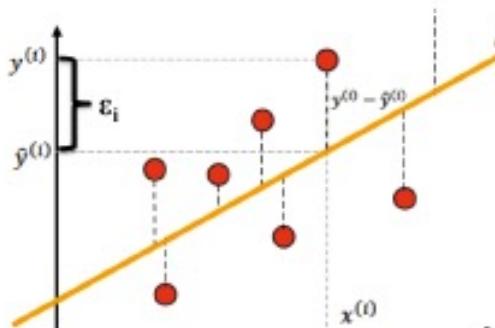
$$w_0 = \frac{1}{m} \sum_{i=0}^m y^{(i)} - w_1 \cdot \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$w_1 = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2},$$

unde  $\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$ , iar  $\bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$ .

Pentru fiecare exemplu de antrenare din  $E$  notăm cu  $\varepsilon_i$  diferența dintre răspunsul corect  $y^{(i)}$  și predicția  $\hat{y}^{(i)}$  (vedeți acest lucru ilustrat și în Figura 1b). Cantitatea  $\varepsilon_i$  se mai numește reziduu. Arătați că suma tuturor reziduurilor este nulă, adică:

$$\sum_{i=1}^m \varepsilon_i = 0.$$



(b)

# Subiectul 2 - rezolvare

Subiectul 2

$$\Sigma = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

Regressia liniara cu conditie dreapta de incerte  $h(x) = w_0 + w_1 x$

$$w_0 = \frac{1}{m} \sum_{i=1}^m y^{(i)} - w_1 \cdot \frac{1}{m} \sum_{i=1}^m x^{(i)} = \bar{y} - w_1 \cdot \bar{x}$$

$$w_1 = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2}$$

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}, \quad \bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$$

$$\varepsilon_i = y^{(i)} - \hat{y}^{(i)} = y^{(i)} - (w_0 + w_1 x^{(i)})$$

Astăzi că  $\sum_{i=1}^m \varepsilon_i = 0$ .

$$\sum_{i=1}^m \varepsilon_i = \sum_{i=1}^m (y^{(i)} - w_0 - w_1 x^{(i)}) = \sum_{i=1}^m y^{(i)} - m \cdot w_0 - w_1 \cdot \sum_{i=1}^m x^{(i)}$$

$$= m \cdot \bar{y} - m(\bar{y} - w_1 \cdot \bar{x}) - w_1 \cdot m \cdot \bar{x}$$

$$= m \bar{y} - m \bar{y} + m w_1 \cdot \bar{x} - w_1 \cdot m \cdot \bar{x}$$

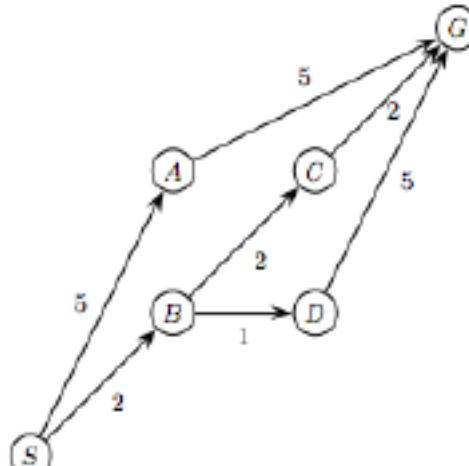
$$= 0.$$

# Subiectul 3a – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (a) care este soluția problemei de căutare folosind strategiile de căutare în lățime și căutare în adâncime? (0,5 puncte)



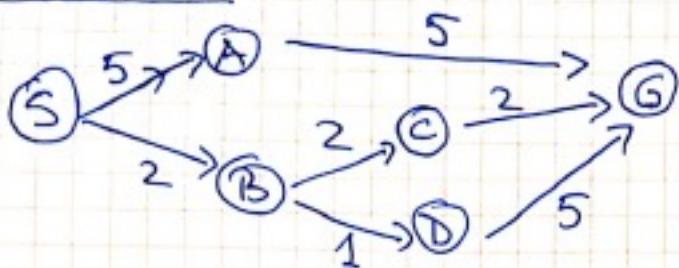
(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

Solutie 3



a)  $F = \{S\} \rightarrow$  frontieră initială  
(contine soluția parțială)

Pas 1:  $F = \{ \underline{S \rightarrow A}, S \rightarrow B \}$

(am expandat nodul S în ordine lexicografică  
pt căutare în labirint)

Pas 2:  $F = \{ \underline{\begin{matrix} S \rightarrow A \rightarrow G \\ \text{solutie} \end{matrix}}, S \rightarrow B \}$

Solutie căutată în labirint este  $S \rightarrow A \rightarrow G$

Căutarea în adâncime:

$F = \{S\}$

Pas 1:  $F = \{ \underline{S \rightarrow A}, S \rightarrow B \}$

$F = \{ \underline{\begin{matrix} S \rightarrow A \rightarrow G \\ \text{solutie} \end{matrix}}, S \rightarrow B \}$

Solutie căutată în adâncime este  $S \rightarrow A \rightarrow G$

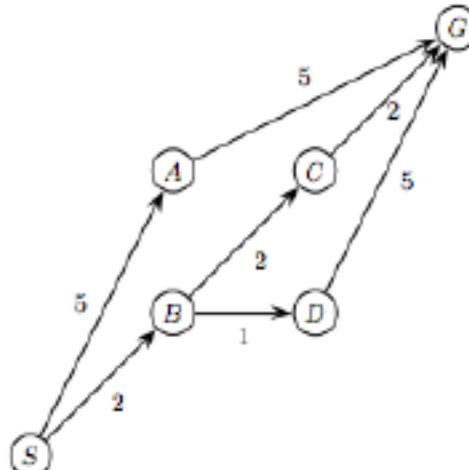
Frontieră	$h_0$	$h_1$	$h_2$	$h^*$
S	0	5	6	6
A	0	3	5	5
B	0	4	2	4
C	0	2	5	2
D	0	5	3	5
G	0	0	0	0

# Subiectul 3b – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (b) care din euristicile  $h_0$ ,  $h_1$ ,  $h_2$  din Figura 2b sunt admisibile? Justificați răspunsul (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3b – rezolvare

b) Curiștici  $h$  este admisibilă dacă ea subestimează peisajul mod în următorul nod al nodului  $n$  către cel mai apropiat nod scop  $0 \leq h(n) \leq h^*(n)$ , unde  $h^*(n)$  este costul real către cel mai apropiat nod scop.

Calculăm  $h^*$  pt fiecare nod:

$$h^*(S)=6, h^*(A)=5, h^*(B)=4, h^*(C)=2, h^*(D)=5, h^*(G)=0$$

Adaugăm coloana  $h^*$  în tabelul de sus.

Observăm că nodurile  $h_2$  sunt curiștici admisibile, dar  $h_2$  nu este admisibilă întrucât  $h_2(C)=5 > h_2^*(C)=2$ .

-5-

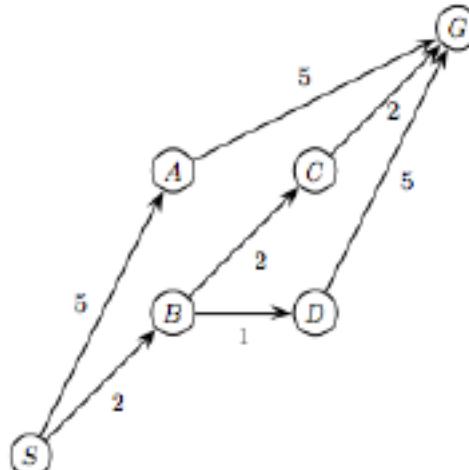
Stare	$h_0$	$h_1$	$h_2$	$h^*$
S	0	5	6	6
A	0	3	5	5
B	0	4	2	4
C	0	2	5	2
D	0	5	3	5
G	0	0	0	0

# Subiectul 3c – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (c) care este soluția returnată de algoritmul  $A^*$  pentru fiecare heuristică  $h_0$ ,  $h_1$ ,  $h_2$  în parte?  
Specificați soluțiile parțiale construite până la obținerea soluției finale. (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3c – rezolvare

c) Pentru h(x) ( $\Rightarrow$  existăca banală  $A^x$  se comportă ca UCS)

$$f(n) = g(n) + h(n)$$

$\downarrow$  cu exact  $\downarrow$  cu estimare pe baza existenței  
cât am sănătății că rămășăte să fac

$$\begin{aligned} P_{n,0}: F &= \{S\} \\ &\underline{f(S) \geq 0} \end{aligned}$$

$$P_{n,1}: F = \left\{ \begin{array}{l} S \rightarrow B, \\ f(S \rightarrow B) = 2 + 0 = 2 \end{array}, \quad S \rightarrow A \right\} \\ f(S \rightarrow A) = 5 + 0 = 5$$

$$P_{n,2}: F = \left\{ \begin{array}{l} S \rightarrow B \rightarrow D, \\ f(n) = 3, \quad S \rightarrow B \rightarrow C, \\ f(n) = 4, \quad S \rightarrow A \end{array} \right\} \\ f(n) = 5$$

$$P_{n,3}: F = \left\{ \begin{array}{l} S \rightarrow B \rightarrow D \rightarrow G, \\ f(n) = 8, \quad S \rightarrow B \rightarrow C, \\ f(n) = 6, \quad S \rightarrow A \end{array} \right\} \\ f(n) = 5$$

$$P_{n,4}: F = \left\{ \begin{array}{l} S \rightarrow B \rightarrow D \rightarrow G, \\ f(n) = 8, \quad S \rightarrow B \rightarrow C \rightarrow G, \\ f(n) = 6, \quad S \rightarrow A \end{array} \right\} \\ f(n) = 5$$

$$P_{n,5}: F = \left\{ \begin{array}{l} S \rightarrow B \rightarrow D \rightarrow G, \\ f(n) = 8, \quad S \rightarrow B \rightarrow C \rightarrow G, \\ f(n) = 6, \quad S \rightarrow A \rightarrow G \end{array} \right\} \\ f(n) = 10$$

Soluția este  $S \rightarrow B \rightarrow C \rightarrow G$

# Subiectul 3c – rezolvare

Pentru h<sub>1</sub>

Pas 0:  $F = \{S\}$   
 $f(n) = 2 + 5 = 5$

Pas 1:  $F = \left\{ S \rightarrow A, S \rightarrow B \right\}$   
 $f(n) = 5 + 3 = 8$ ,  $\underline{f(n_1) = 2 + 4 = 6}$

Pas 2:  $F = \left\{ S \rightarrow A, S \rightarrow B \rightarrow C, S \rightarrow B \rightarrow D \right\}$   
 $f(n) = 8$ ,  $\underline{f(n_1) = 4 + 2 = 6}$ ,  $\underline{f(n_1) = 3 + 5 = 8}$

Pas 3:  $F = \left\{ S \rightarrow A, S \rightarrow B \rightarrow C \rightarrow G, S \rightarrow B \rightarrow D \right\}$   
 $f(n) = 8$ ,  $\underline{f(n_1) = 6 + 0 = 6}$ ,  $\underline{f(n_1) = 3 + 5 = 8}$   
Solutie

Solutia este  $S \rightarrow B \rightarrow C \rightarrow G$

# Subiectul 3c – rezolvare

Pentru h<sub>2</sub>

Pass 0:  $F = \{S\}$ ,  
 $f(m) = 0 + 6 = 6$

Pass 1:  $F = \{S \rightarrow A, S \rightarrow B\}$ ,  
 $f(m) = 2 + 5 = 10, f(m) = 2 + 2 = 4$

Pass 2:  $F = \{S \rightarrow A, S \rightarrow B \rightarrow C, S \rightarrow B \rightarrow D\}$ ,  
 $f(m) = 10, f(m) = 4 + 5 = 9, f(m) = 3 + 3 = 6$

Pass 3:  $F = \{S \rightarrow A, S \rightarrow B \rightarrow C, S \rightarrow B \rightarrow D \rightarrow G\}$ ,  
 $f(m) = 10, f(m) = 9, f(m) = 8 + 10 = 18$

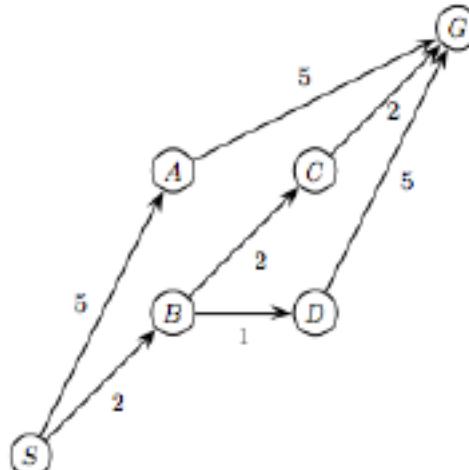
Soluție astă  $S \rightarrow B \rightarrow D \rightarrow G$ .

# Subiectul 3d – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (d) care este soluția returnată de algoritmul de căutare Greedy ce folosește euristica  $h_1$ ? Specificați soluțiile parțiale construite până la obținerea soluției finale. (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3d – rezolvare

d) Greedy cu  $h_1$

Pas 0:  $F = \{S\}$   
 $h_1(S) = 5$

Pas 1:  $F = \left\{ S \xrightarrow{\underline{h_1(w)=3}}, S \xrightarrow{\underline{h_1(w)=4}} A, B \right\}$

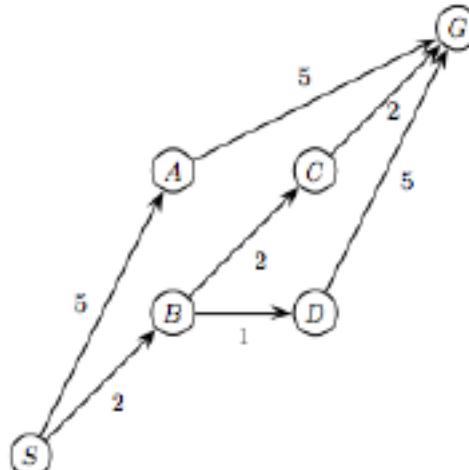
Pas 2:  $F = \left\{ S \xrightarrow{\underline{h_2(w)=0}} A \xrightarrow{\underline{h_2(w)=4}} G, S \xrightarrow{\underline{h_2(w)=4}} B \right\}$   
Solutie

# Subiectul 3e – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (e) care este soluția returnată de algoritmul de căutare uniformă după cost (UCS)? Specificați soluțiile parțiale construite până la obținerea soluției finale. (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3e – rezolvare

e)  $UCS = A^*$  cu  $h_0$  = punctul c) prima parte

c) Pentru  $h_0$  ( $\Rightarrow$  există banală  $A^*$  se comportă ca UCS)

$$f(n) = g(n) + h(n)$$

$\downarrow$  cost exact       $\downarrow$  cost estimat pe baza existenței  
cât am sănăt      către care să fac

$$\begin{aligned} P_{A,0}: \quad F = & \{ S \} \\ & \underline{f(S)=0} \end{aligned}$$

$$P_{A,1}: \quad F = \{ S \xrightarrow{B} B, \quad S \xrightarrow{A} A \}$$

$\underline{f(S \xrightarrow{B} B) = 2 + 0 = 2}$        $\underline{f(S \xrightarrow{A} A) = 5 + 0 = 5}$

$$P_{A,2}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{C} C, \quad S \xrightarrow{A} A \}$$

$\underline{f(n)=3}$        $\underline{f(n)=4}$        $\underline{f(n)=5}$

$$P_{A,3}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{D} D \xrightarrow{G} G, \quad S \xrightarrow{B} B \xrightarrow{C} C, \quad S \xrightarrow{A} A \}$$

$\underline{f(n)=8}$        $\underline{f(n)=7}$        $\underline{f(n)=5}$

$$P_{A,4}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{D} D \xrightarrow{G} G, \quad S \xrightarrow{B} B \xrightarrow{C} C \xrightarrow{G} G, \quad S \xrightarrow{A} A \}$$

$\underline{f(n)=8}$        $\underline{f(n)=6}$        $\underline{f(n)=5}$

$$P_{A,5}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{D} D \xrightarrow{G} G, \quad S \xrightarrow{B} B \xrightarrow{C} C \xrightarrow{G} G, \quad S \xrightarrow{A} A \xrightarrow{G} G \}$$

$\underline{f(n)=8}$        $\underline{f(n)=6}$        $\underline{f(n)=10}$

Soluția este  $S \xrightarrow{B} B \xrightarrow{C} C \xrightarrow{G} G$

# Subiectul 4a – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (a) Care este numărul de mutări maxim (= factorul de ramificare) pe care îl are la dispoziție fiecare jucător în orice moment al jocului? Justificați răspunsul. (0,25 puncte)

1	2	3	4
X	0	0	0
0	X	X	0
X	X	0	X

(a)

1	2	3	4
X	0		
0	X		0
X	X	0	X

(b)

Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea -1 (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

# Subiectul 4a – rezolvare

## Subiectul 4

- a) Numărul maxim de mutări (= factorul de ranificare) pe care îl are la dispoziție fiecare jucător = numărul de coloane libere. Inițial, toate coloanele sunt libere, deci  $b = 4$ , spașterea jucătorului să poată întâmpina ca numai o singură coloană să fie liberă, deci  $b = 1$ .

# Subiectul 4b – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (b) Considerăm că după 9 mutări ajungem cu jocul în starea dată de Figura 3b. Jucătorul 0 este acum la mutare. Desenați arborele de joc asociat, considerând drept rădăcină starea actuală a jocului. (0,75 puncte)

1	2	3	4
X	0	0	0
0	X	X	0
X	X	0	X

(a)

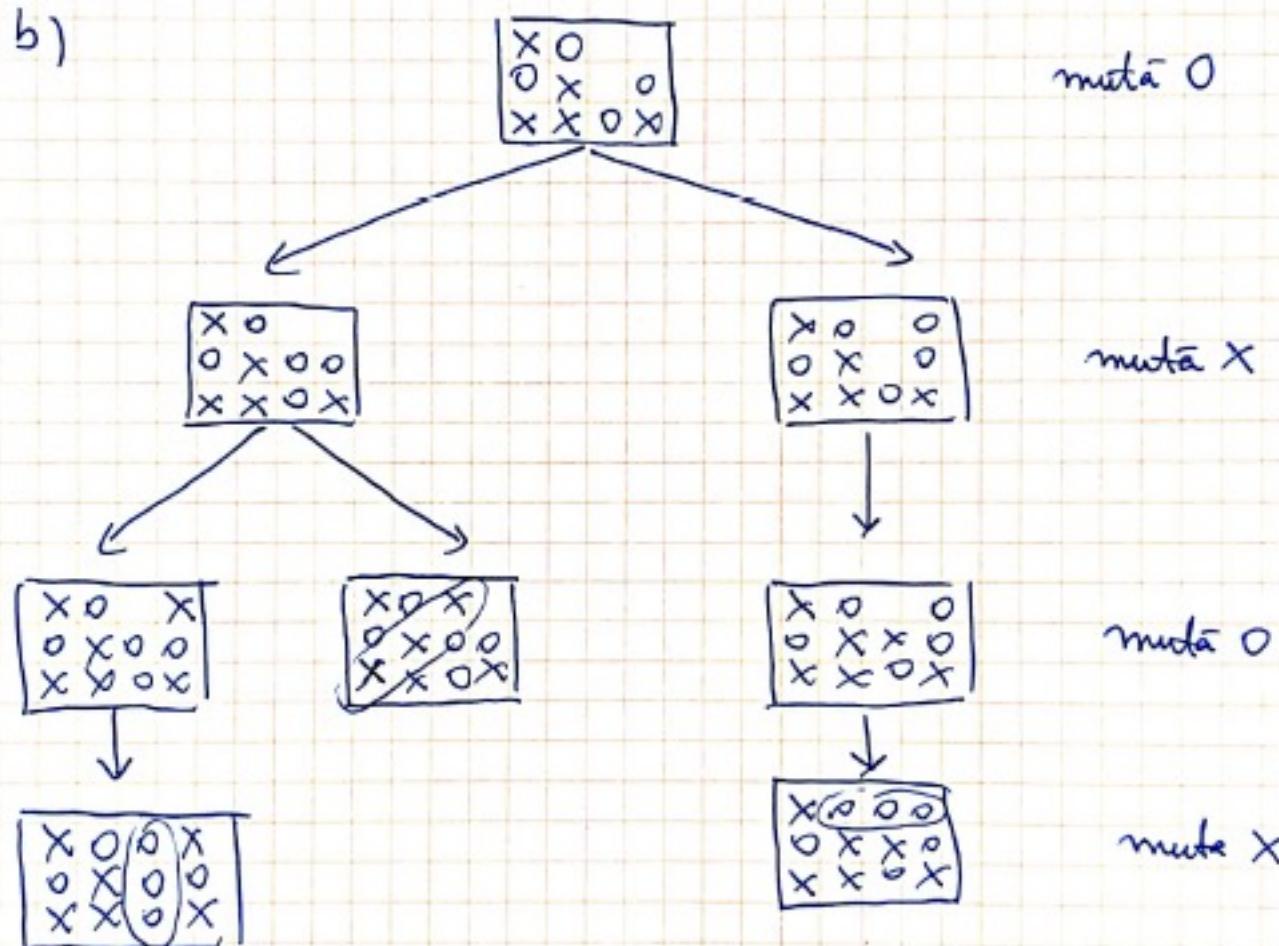
1	2	3	4
X	0		
0	X		0
X	X	0	X

(b)

Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea -1 (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

# Subiectul 4b – rezolvare

b)



# Subiectul 4c – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (c) considerăm jucătorul X ca fiind MAX și jucătorul 0 ca fiind MIN. Funcția de utilitate asociată nodurilor terminale este următoarea: dacă jucătorul X câștigă atunci el primește  $k$  puncte iar jucătorul 0 pierde  $k$  puncte, dacă jocul se termină la egalitate atunci fiecare are 0 puncte. Valoarea  $k$  atunci când jucătorul X câștigă este stabilită astfel:  $k = 3$  puncte pentru o linie realizată din simboluri dispuse pe diagonală,  $k = 2$  pentru o linie realizată din simboluri dispuse pe verticală și  $k = 1$  pentru o linie realizată din simboluri dispuse pe orizontală. Valoarea  $k$  atunci când jucătorul 0 câștigă pe cazurile enumerate anterior este  $-3, -2, -1$ . Folosind algoritmul Minimax etichetați arborele de joc de la punctul anterior cu valorile minimax asociate fiecărui nod. (0,5 puncte)

1	2	3	4
1	X	0	0
2	0	X	X
3	X	X	0
4			X

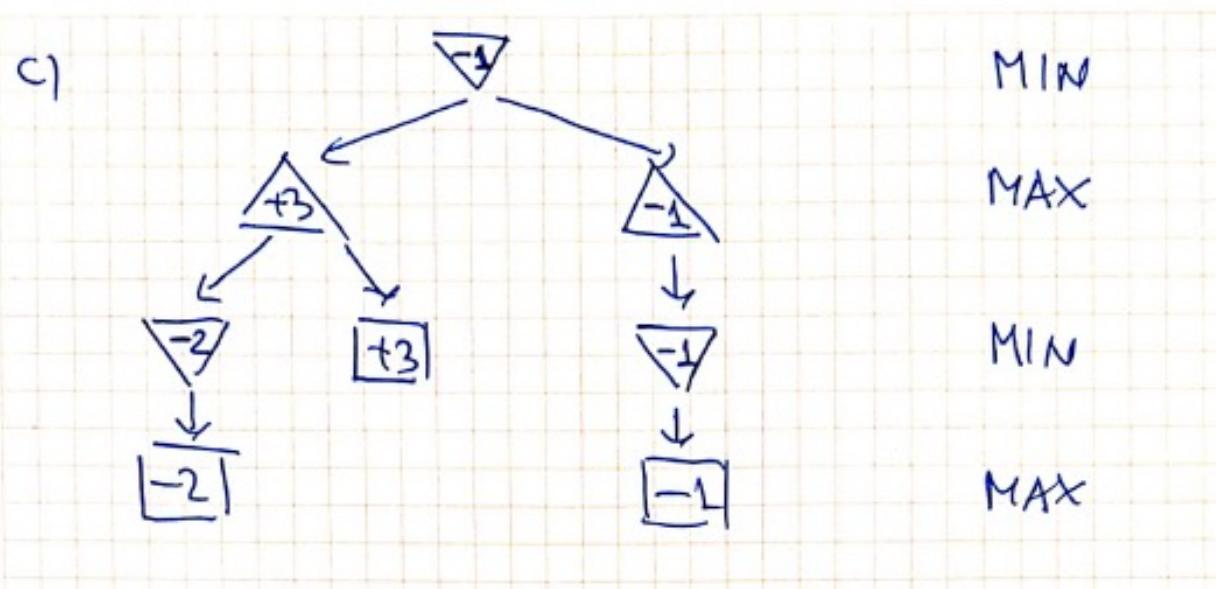
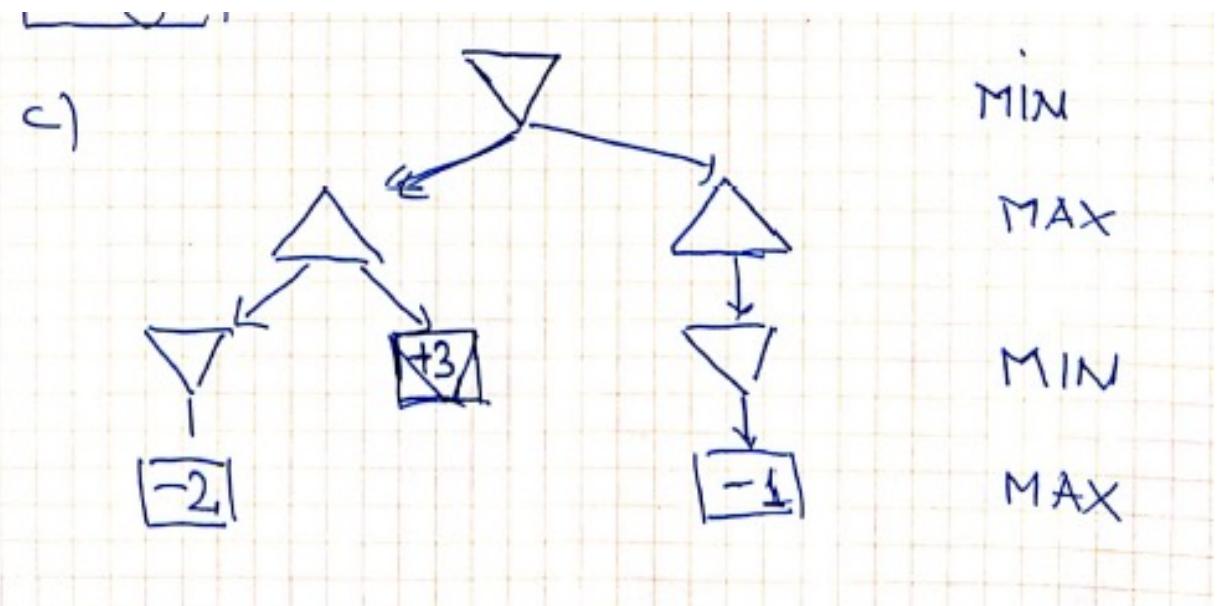
(a)

1	2	3	4
1	X	0	
2	0	X	
3	X	X	0
4			X

(b)

Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea  $-1$  (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

# Subiectul 4c – rezolvare



# Subiectul 4d – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (d) Folosiți algoritmul de retezare Alfa-Beta pentru accelerarea calcului valorilor Minimax. Ce noduri din arbore de joc nu vor fi explorate? Puteți reordona fiile fiecărui subarbore pentru a maximiza numărul de noduri care nu vor fi explorate de algoritmul Alfa-Beta? Justificați răspunsul. (0,5 puncte)

1	2	3	4
X	0	0	0
0	X	X	0
X	X	0	X

(a)

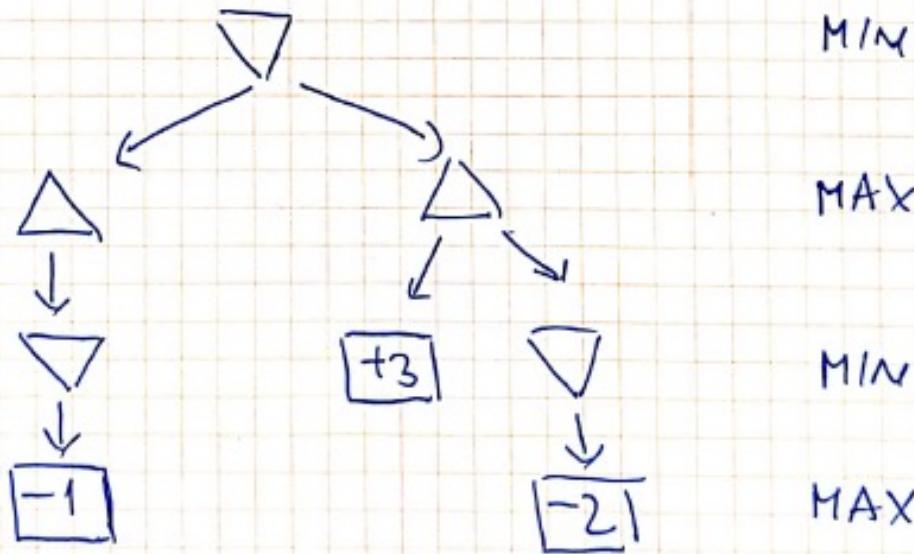
1	2	3	4
X	0		
0	X		0
X	X	0	X

(b)

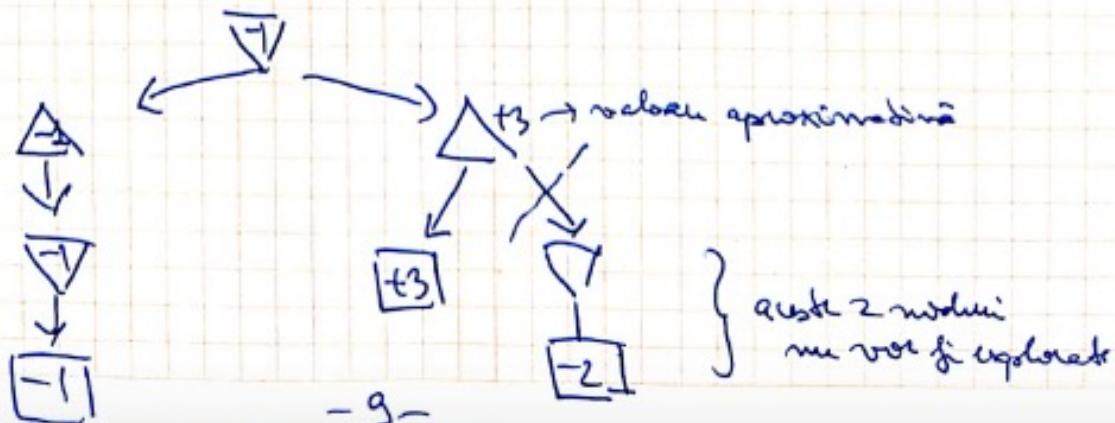
Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea -1 (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

d) Cu Alfa-Beta, în configurația actuală nu se poate face nicio reținere.

Rearanjăm arborele astfel:



Aplicam Alfa-Beta



**3** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), 0, (3,-2), 0, (5,-2), 0, (7,-1), 1, (7,2), 1, (7,4), 1\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspusul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

3 intrebare

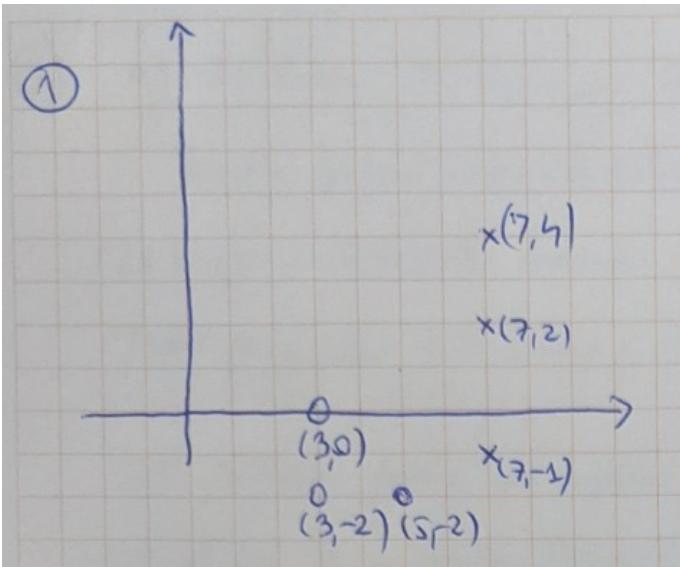
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.



3 intrebare

Nu a primit  
răspuns încă

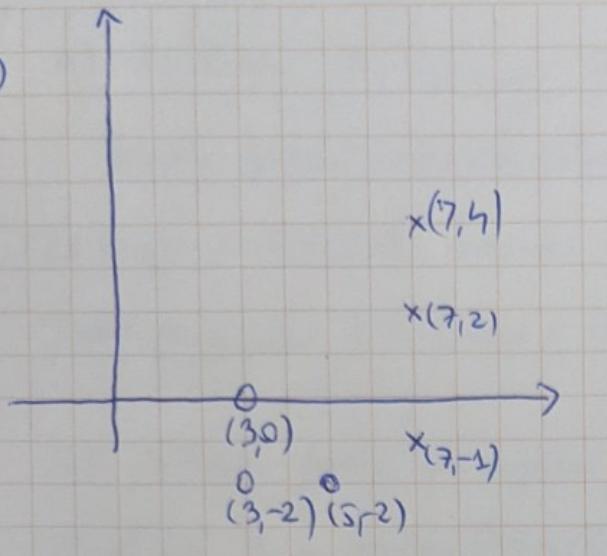
Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspusul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

①



1. Considerăm punctul  $(7,0)$ .

Cei mai apropiati 3 vecini sunt :

- $(7,-1)$  distanta 1 (distanta euclidiana)
- $(7,2)$  distanta 1
- $(5,-2)$  distanta  $\sqrt{8}$  ( $\sqrt{(7-5)^2 + (0-(-2))^2}$ )

3 intrebare

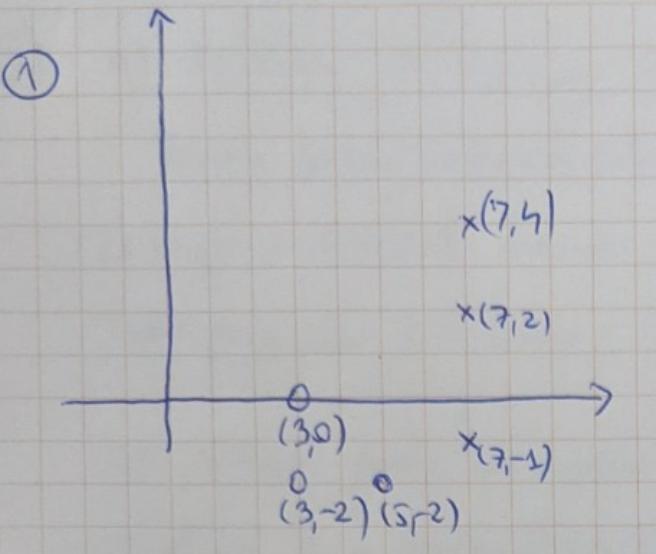
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu  $k=3$ . Fie multimea  $S$  de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.



2. Considerem punctul  $(3,0)$

Cei mai apropiati 3 vecini în sensul distantei Manhattan sunt:

- $(3,0)$  distanță Manhattan 0
- $(3,-2)$  distanță Manhattan 2  $(|3-3| + |0-(-2)|)$
- $(5,-2)$  distanță Manhattan 4  $(|3-5| + |0-(-2)|)$

3 intrebare

Nu a primit  
răspuns încă

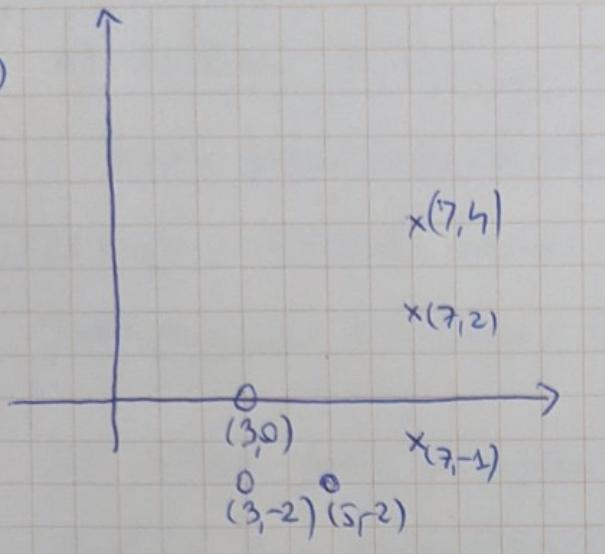
Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu  $k=3$ . Fie multimea  $S$  de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), ((5,-2),0), ((7,-1),1), ((7,2),1), ((7,4),1)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

①



3. Punctul  $(5,1)$ :

- folosind distanta Euclidiana cei 3 vecini sunt  $(3,0)$  - etichetă 0  $\sqrt{51}$   
 $(7,2)$  - etichetă 1  $\sqrt{55}$   
 $(7,-1)$  - etichetă 1  $\sqrt{58}$

$\Rightarrow$  va fi clasificat cu eticheta 1

- folosind distanta Manhattan cei 3 vecini sunt:  $(3,0)$  - etichetă 0  $(3)$   
 $(7,2)$  - etichetă 1  $(3)$   
 $(5,-2)$  - etichetă 0  $(3)$

$\Rightarrow$  va fi clasificat cu etichetă 0

**2** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

¶ Întrebare cu  
flag

Consideram problema clasificarii literelor mici (clasele ‘a’, ‘b’, …, ‘z’) scrise de mana din alfabetul englez (in total 26 de clase) pe baza modelului Naïve Bayes. Datele de antrenare constau in 100 de exemple de antrenare cu imagini 28 x 28 pixeli pentru fiecare litera in parte. In total aveti 2600 exemple de antrenare. Consideram ca stim probabilitatea a-priori de aparitie a fiecarei litere pe baza studiului frecventei de aparitie a literelor in limba engleza. Pentru estimarea probabilitatii likelihood folosim modelul Naïve Bayes care considera ca fiecare pixel din imagine are o distributie normala specifica pozitiei pixelului din imagine si clasei literei.

1. cati parametri trebuie sa invete modelul vostru pentru a putea clasifica o litera 28 x 28?
2. cum ati invata parametri corespunzatori pixelului de pe linia 14, coloana 14 (centrul imaginii) din clasa ‘a’?

2 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram problema clasificarii literelor mici (clasele 'a', 'b', ..., 'z') scrise de mana din alfabetul englez (in total 26 de clase) pe baza modelului Naïve Bayes. Datele de antrenare constau in 100 de exemple de antrenare cu imagini 28 x 28 pixeli pentru fiecare litera in parte. In total aveti 2600 exemple de antrenare. Consideram ca stim probabilitatea a-priori de aparitie a fiecarei litere pe baza studiului frecventei de aparitie a literelor in limba engleza. Pentru estimarea probabilitatii likelihood folosim modelul Naïve Bayes care considera ca fiecare pixel din imagine are o distributie normala specifica pozitiei pixelului din imagine si clasei literei.

1. cati parametri trebuie sa invete modelul vostru pentru a putea clasifica o litera 28 x 28?
2. cum ati invata parametri corespunzatori pixelului de pe linia 14, coloana 14 (centrul imaginii) din clasa 'a'?

② a) 26 litere 'a', 'b', 'c', ... 'z'

Pentru fiecare litera (in total 26) invatați puncte fizice sau pozitii din cel 784

( $28 \times 28$ ) și distribuția normală  $N(\mu, \sigma^2)$

Total parametri: 26 litere  $\times$  784 pozitii  $\times$  2 parametri

b) Selectați toate exemplul (în număr de 100) cu litera 'a'. Selectați valoarele pixelelor de pe poziție (14,14) în total 100 de valori. Calculați apoi media  $\mu$  și deviație standard  $\sigma$ .

**1** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

1 intrebare

Nu a primit  
răspuns încă

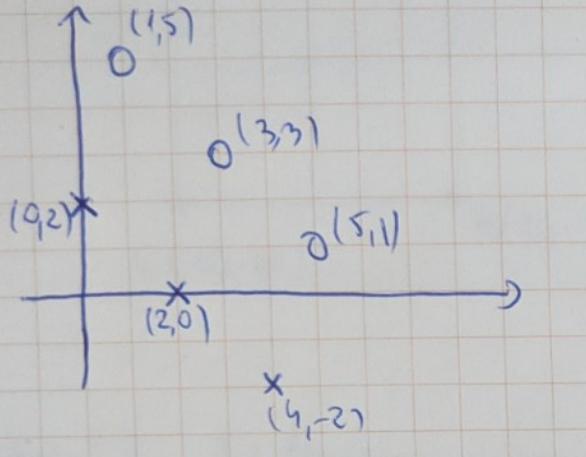
Marcat din 1,00

Întrebare cu  
flag

Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

③



1 intrebare

Nu a primit  
răspuns încă

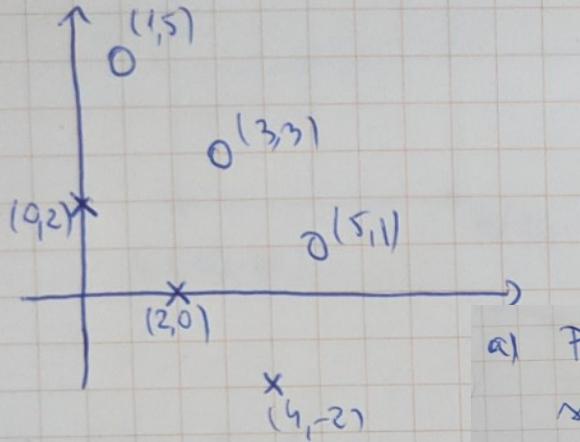
Marcat din 1,00

Întrebare cu  
flag

Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

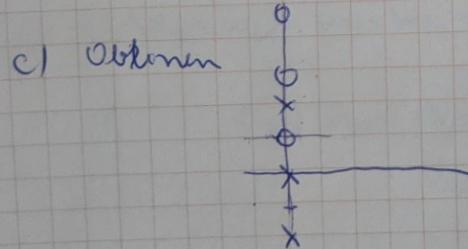
③



a)  $P$  este liniar separabila, spre exemplu dreapta de ecuatie  $y+x-4=0$  separa perfect clasele

b) Obtemem   
 $P' = \{(0,1), (2,1), (4,1), (1,0), (3,0), (5,0)\}$

Multimea  $P'$  nu mai este liniar separabila, nu putem avea un hiperplan (-punct) care sa separe perfect cele 2 clase



$P'' = \{(2,1), (0,1), (-2,1), (5,0), (3,0), (1,0)\}$

Multimea  $P''$  nu este liniar separabila, nu putem avea un hiperplan (-punct) care sa separe perfect cele 2 clase

**10** întrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Consideram o problema de clasificare binara cu puncte din  $\mathbf{R}$  (dreapta reala). Fie multimea de exemple de antrenare  $P = \{(-1,1), (0,0), (1,0), (2,0), (3,1), (4,1)\}$  unde fiecare exemplu de antrenare e reprezentat de coordonata sa = abscisa si de eticheta corespunzatoare (0 sau 1). Scrieti ponderile  $w_0$  (biasul),  $w_1$  (ponderea pentru abscisa) a unui perceptron cu functia de activare hardlim care obtine acuratetea cea mai mare pe multimea P.

10 intrebare

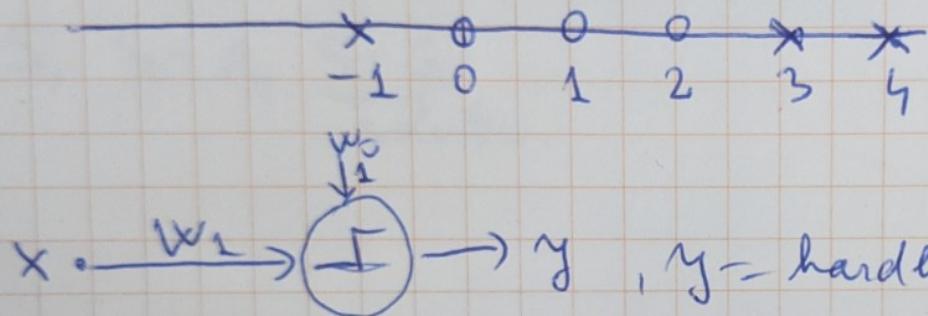
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram o problema de clasificare binara cu puncte din  $\mathbf{R}$  (dreapta reala). Fie multimea de exemple de antrenare  $P = \{(-1,1), (0,0), (1,0), (2,0), (3,1), (4,1)\}$  unde fiecare exemplu de antrenare e reprezentat de coordonata sa = abscisa si de eticheta corespunzatoare (0 sau 1). Scrieti ponderile  $w_0$  (biasul),  $w_1$  (ponderea pentru abscisa) a unui perceptron cu functia de activare hardlim care obtine acuratetea cea mai mare pe multimea P.

7



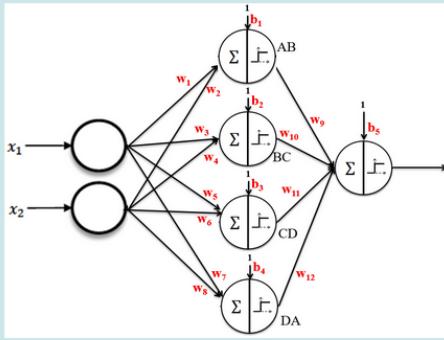
$$y = \text{hardlim}(w_1 \cdot x + w_0) = \begin{cases} 1, & w_1 \cdot x + w_0 \geq 0 \\ 0, & \text{altele} \end{cases}$$

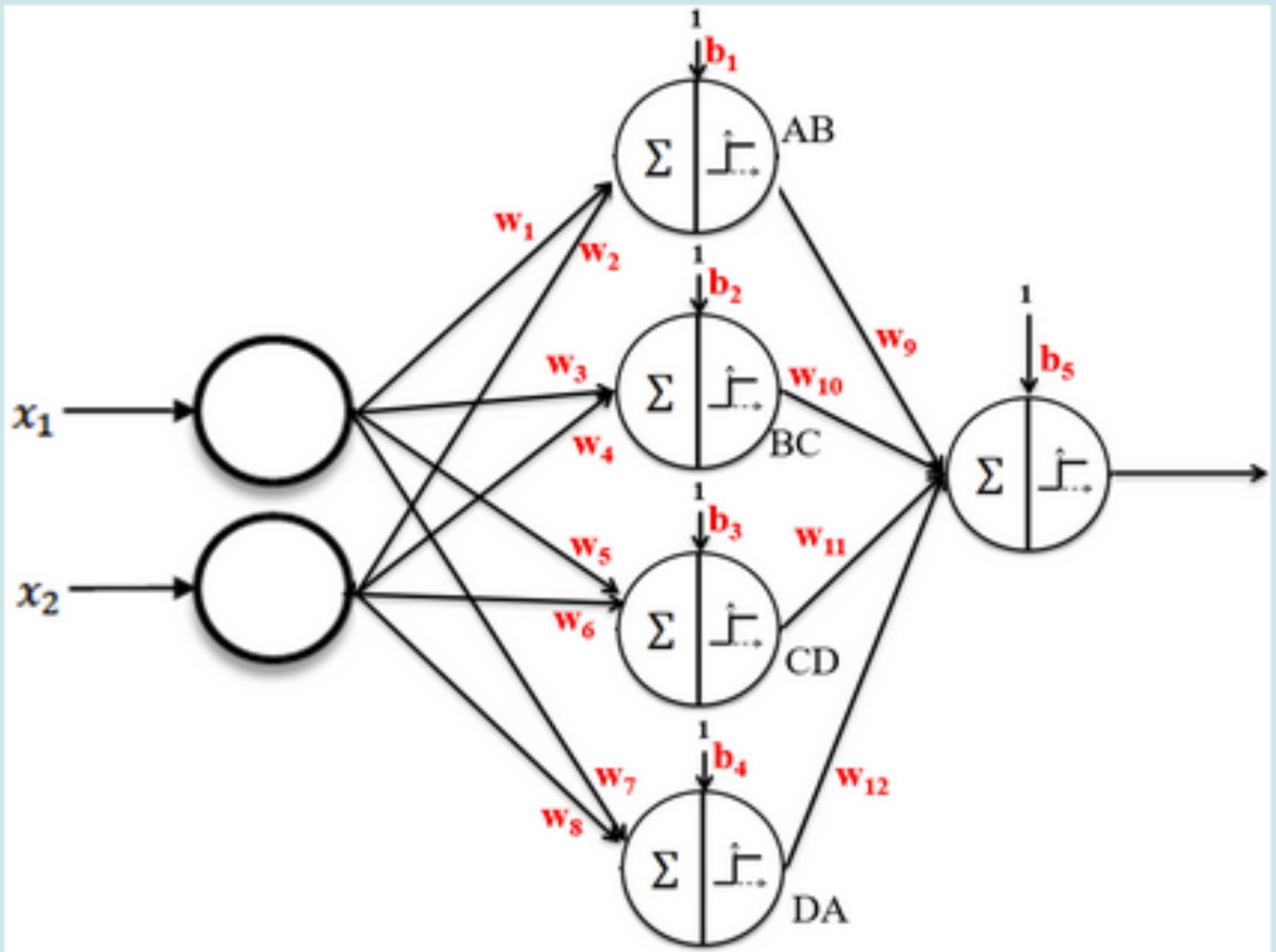
$$w_1 = 1$$

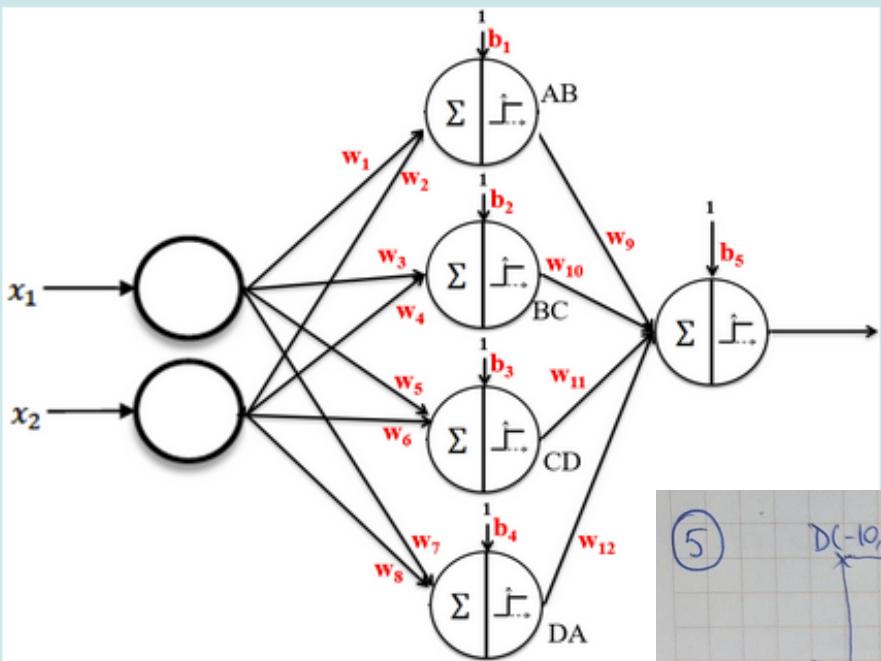
$$w_0 = 2.5$$

$$y = \begin{cases} 1, & x \geq 2.5 \\ 0, & \text{altele} \end{cases}$$

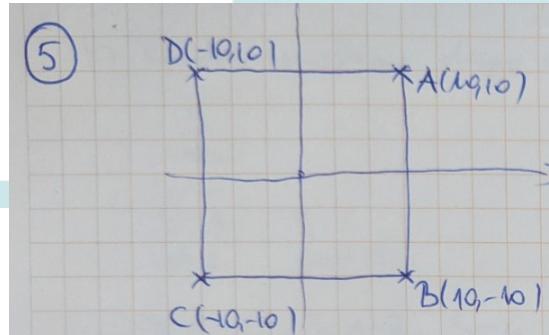
Consideram patratul de varfuri A(10,10), B(10,-10), C(-10,-10), D(-10,10) si reteaua neuronală din figura de mai jos care dorește să implementeze funcția indicator a patratului ABCD: pentru fiecare punct din interiorul sau de pe frontieră patratului rețea trebuie să aibă ieșirea 1, în rest ieșirea rețelei ar trebui să fie 0. Rețea neuronală de perceptri sunt două straturi ascunse. Pe primul strat ascuns sunt patru perceptri care implementează ecuațiile dreptelor AB, BC, CD, DA astfel încât punctele din patrat primesc eticheta 1 din partea tuturor celor 4 perceptri. Al doilea strat ascuns are un singur perceptor care implementează funcția AND de 4 variabile (ieșirile perceptronilor de pe primul strat ascuns). Un punct este în patrat dacă toți cei 4 perceptri de pe primul strat au ieșirea 1. Un punct din exteriorul patratului este caracterizat de faptul că cel puțin un perceptor de pe primul strat ascuns va avea ieșirea 0. Funcția de activare a tuturor perceptronilor din desen este hardlim. Scrieți un set de valori pentru ponderile  $w_1, w_2, \dots, w_{12}$  și bias-urile  $b_1, b_2, \dots, b_5$  astfel încât rețea neuronală obținută să implementeze funcția indicator a patratului ABCD după logica descrisă.







trebuie a implementa functia indicator a patratului ABCD: pentru fiecare punct din interiorul sau de pe frontieră patratului rețea trebuie să aibă ieșirea 1 (pentru fiecare punct din interiorul sau de pe frontieră patratului rețea trebuie să aibă ieșirea 1). Un punct este în patrat dacă toți cei 4 perceptri din primul strat au ieșirea 1. Un punct din exteriorul patratului percepatorilor din desen este hardlim. Scrieți un set de valori pentru ponderile  $w_1, w_2, \dots, w_{12}$  și bias-urile  $b_1, b_2, \dots, b_5$  astfel încât rețea



$$\begin{aligned} AB: & x_1 = 10, \text{ sau } -x_1 = -10 \\ & x_1 - 10 = 0, \quad -x_1 + 10 = 0. \end{aligned}$$

Perceptorul care implementă cheiept AB  
trebuie să dea output-ul 1 pt totuști  
din interiorul patratului. Toate părțile le

stăgează în 10, trebuie să primească 1 ( $x_1 \leq 10$ ). De cînd că ieșirea este  
 $-x_1 + 10 = 0$  ( $w_1 = -1, w_2 = 0, b_1 = 10$ ).

Similar obținem BC:  $x_2 + 10 = 0$  ( $w_3 = 0, w_4 = 1, b_2 = 10$ )

CD:  $x_2 - 10 = 0$  ( $w_5 = 1, w_6 = 0, b_3 = 10$ )

AD:  $-x_2 + 10 = 0$  ( $w_7 = 0, w_8 = -1, b_4 = 10$ )

Perceptorul de pe stratul 2 implementă AND, deci pot combinde

$$w_9 = w_{10} = w_{11} = w_{12} = 1, \quad b_5 = -35 \quad (\text{ sau } -36, -32, -31, \text{ etc})$$

**8** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Consideram problema regresiei liniare simple pe multimea de antrenare  $S = \{(3,0), (4,2), (6,0), (7,2)\}$ . Notam cu  $w = (w_0, w_1)$  astfel incat  $y = w_0 + w_1 * x$  este dreapta solutie a regresiei liniare. Cat este  $w$ ?

8 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

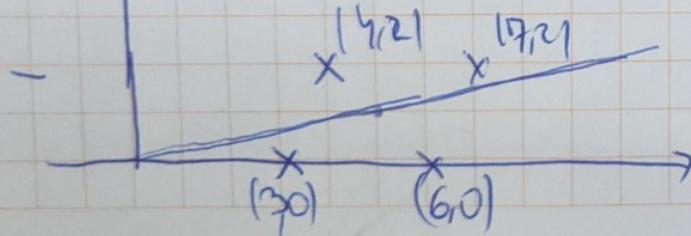
Consideram problema regresiei liniare simple pe multimea de antrenare  $S = \{(3,0), (4,2), (6,0), (7,2)\}$ . Notam cu  $w = (w_0, w_1)$  astfel incat  $y = w_0 + w_1 \cdot x$  este dreapta solutie a regresiei liniare. Cat este  $w$ ?

⑥

$$S = \{(3,0), (4,2), (6,0), (7,2)\} \quad w = (w_0, w_1) \quad \bar{x} = 5, \bar{y} = 1$$
$$w_1 = \frac{\sum (x^i - \bar{x})(y^i - \bar{y})}{\sum (x^i - \bar{x})^2} = \frac{(3-5)(0-1) + \dots + (7-5)(2-1)}{(3-5)^2 + \dots + (7-5)^2}$$

$$= \frac{2}{10} = \frac{1}{5}$$

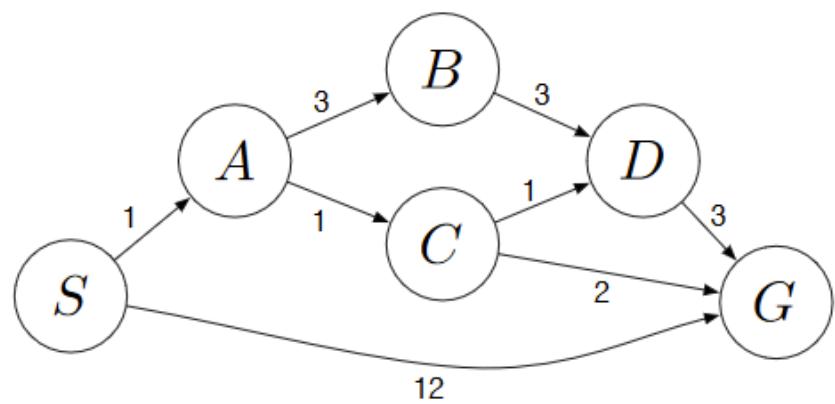
$$w_0 = \frac{1}{m} [\sum y^i - w_1 \sum x^i] = \bar{y} - w_1 \bar{x} =$$
$$= 1 - \frac{1}{5} \cdot 5 = 0$$



Consideram problema de cautare data de graful din figura de mai jos. S este starea initiala, G este starea scop. O strategie de cautare care porneste din starea S si vrea sa ajunga la starea G exploreaza starile intr-o anumita ordine. Implicit, se prefera explorarea starilor in ordine lexicografica. Realizati urmatoarele:

1. Care este solutia problemei de cautare folosind strategia de cautare in latime? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.
2. Care este solutia problemei de cautare folosind strategia de cautare uniforma dupa cost? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.

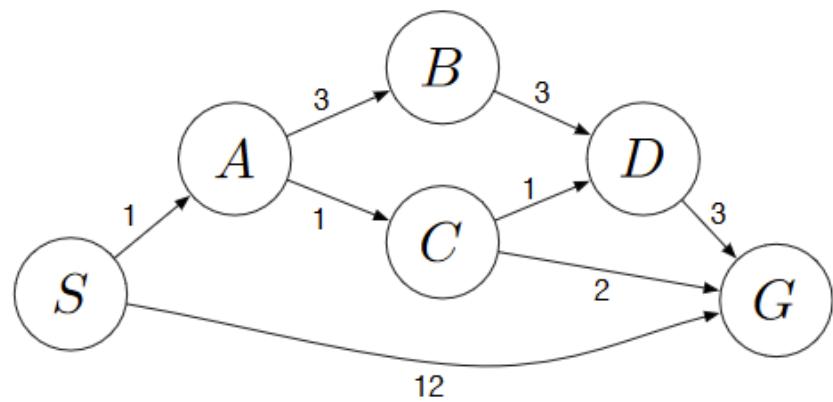
(Cand raspundeti la intrebari folositi ca notatie pentru o solutie parciala urmatoare notatie: S-A-D-G - solutie parciala invalida pentru cazul de fata)



Consideram problema de cautare data de graful din figura de mai jos. S este starea initiala, G este starea scop. O strategie de cautare care porneste din starea S si vrea sa ajunga la starea G exploreaza starile intr-o anumita ordine. Implicit, se prefera explorarea starilor in ordine lexicografica. Realizati urmatoarele:

1. Care este solutia problemei de cautare folosind strategia de cautare in latime? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.
2. Care este solutia problemei de cautare folosind strategia de cautare uniforma dupa cost? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.

(Cand raspundeti la intrebari folositi ca notatie pentru o solutie parciala urmatoare notatie: S-A-D-G - solutie parciala invalida pentru cazul de fata)



⑦ a) Cautare in latime:

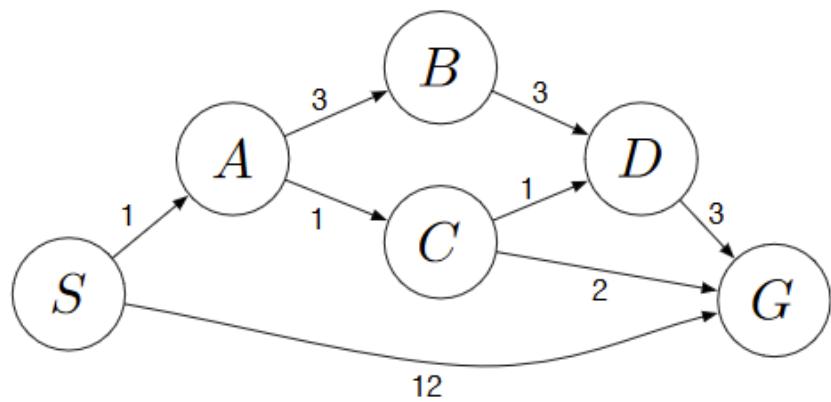
S,

$S \rightarrow A$ ,  $S \rightarrow G$   
solutie

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează starile într-o anumita ordine. Implicit, se preferă explorarea starilor în ordine lexicografică. Realizați urmatoarele:

1. Care este soluția problemei de căutare folosind strategia de căutare în latime? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.
2. Care este soluția problemei de căutare folosind strategia de căutare uniformă după cost? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție parțială invalidă pentru cazul de față)



⑦ a) Căutare în latime:

$$\begin{array}{c} S, \\ S-A, \frac{S-G}{\text{solutie}} \end{array}$$

b) căutare uniformă după cost

$$\begin{array}{c} S \\ \text{cost} = 0 \end{array}$$

$$\begin{array}{c} S-A \\ \text{cost } 1 \end{array}, \begin{array}{c} S-G \\ \text{cost } 12 \end{array}$$

$$\begin{array}{c} S-A-C \\ \text{cost } 2 \end{array}, \begin{array}{c} S-A-B \\ \text{cost } 3 \end{array}, \begin{array}{c} S-G \\ \text{cost } 12 \end{array}$$

$$\begin{array}{c} S-A-C-D \\ \text{cost } 3 \end{array}, \begin{array}{c} S-A-C-G \\ \text{cost } 4 \end{array}, \begin{array}{c} S-A-B \\ \text{cost } 3 \end{array}, \begin{array}{c} S-G \\ \text{cost } 12 \end{array}$$

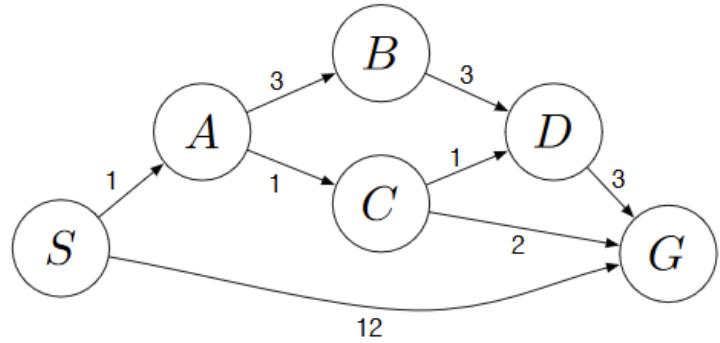
$$\begin{array}{c} S-A-C-D-G \\ \text{cost } 6 \end{array}, \begin{array}{c} S-A-C-G \\ \text{cost } 4 \end{array}, \begin{array}{c} S-A-B \\ \text{cost } 3 \end{array}, \begin{array}{c} S-G \\ \text{cost } 12 \end{array}$$

$$\begin{array}{c} S-A-C-D-G, \begin{array}{c} (S-A-C-G) \\ \text{cost } 4 \end{array}, \begin{array}{c} S-A-B-D \\ \text{cost } 7 \end{array}, \begin{array}{c} S-G \\ \text{cost } 12 \end{array} \\ \text{solutie} \end{array}$$

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează stările într-o anumita ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (la costuri egale). Considerăm euristicile  $h_1$  și  $h_2$  din tabelul de mai jos. Realizați următoarele:

1. Care din euristicile  $h_1$  și  $h_2$  este admisibilă? Justificați răspunsul.
2. Care este soluția problemei de căutare folosind algoritmul A\* cu heuristică  $h_1$ ? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție invalidă pentru cazul de față)

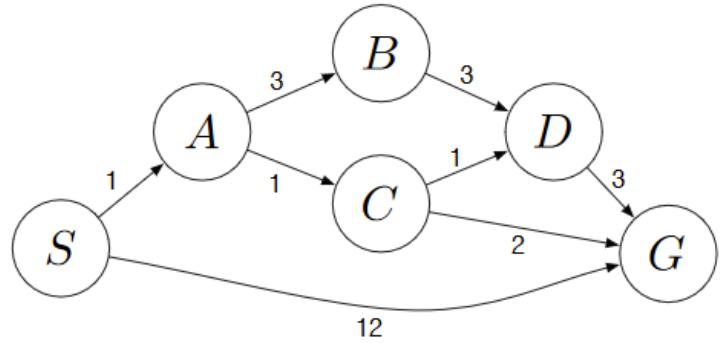


State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează stările într-o anumita ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (la costuri egale). Considerăm euristicile  $h_1$  și  $h_2$  din tabelul de mai jos. Realizați următoarele:

1. Care din euristicile  $h_1$  și  $h_2$  este admisibilă? Justificați răspunsul.
2. Care este soluția problemei de căutare folosind algoritmul A\* cu heuristică  $h_1$ ? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție invalidă pentru cazul de față)



State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

⑧ a)  $h$  este admisibilă dacă substanțnează costul optim al găsirii mod.

$$h^*(S) = 4$$

$$h_1(S) = 5 \Rightarrow h_1 \text{ nu este admisibilă}$$

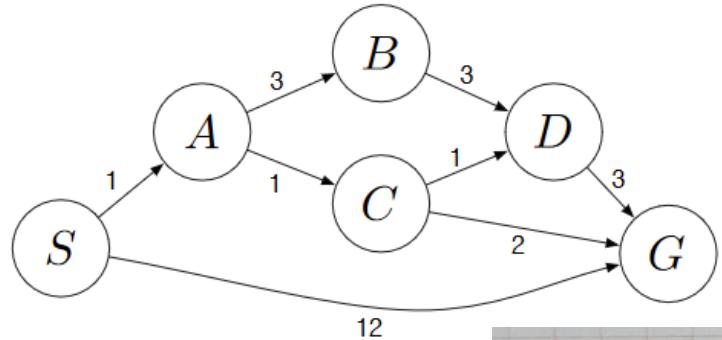
$h_2$  verifică condiția  $h_2(N) \leq h^*(N)$  pentru orice nod N

$\Rightarrow h_2$  este admisibila

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează stările într-o anumita ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (la costuri egale). Considerăm euristicile  $h_1$  și  $h_2$  din tabelul de mai jos. Realizați următoarele:

1. Care din euristicile  $h_1$  și  $h_2$  este admisibilă? Justificați răspunsul.
2. Care este soluția problemei de căutare folosind algoritmul A\* cu heuristică  $h_1$ ? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție invalidă pentru cazul de față)



State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

b)  $f(n) = g(n) + h(n)$   
 tractat (exact) vechi (șters)

$S(5)$   
 $g=9, h=5$

$S-A(4), S-G(12)$   
 $g=1, h=3 \quad g=12, h=0$

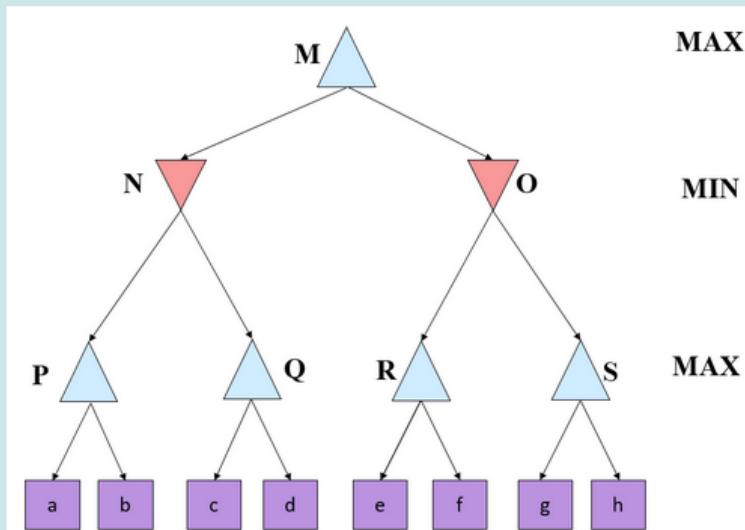
$S-A-C(4), S-A-B(10), S-G(12)$   
 $g=2, h=2 \quad g=4, h=6 \quad g=12, h=0$

$S-A-C-D(6), S-A-C-G(4), S-A-B(10), S-C(12)$   
 $g=3, h=3 \quad g=4, h=\infty \quad g=4, h=6 \quad g=12, h=0$

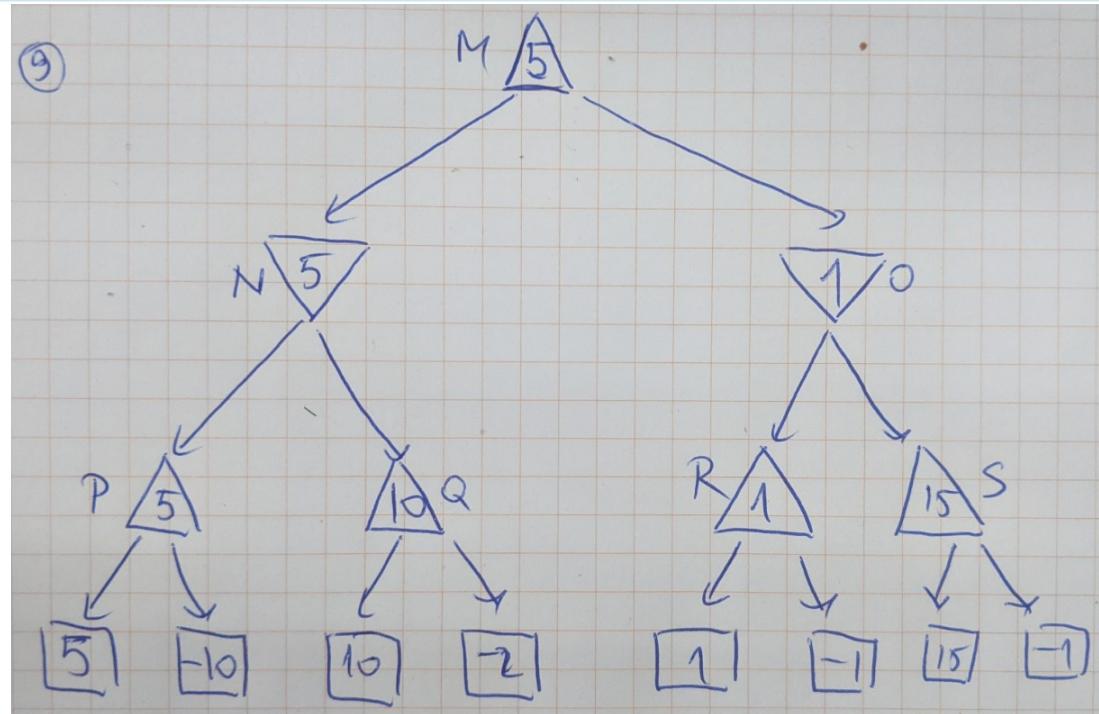
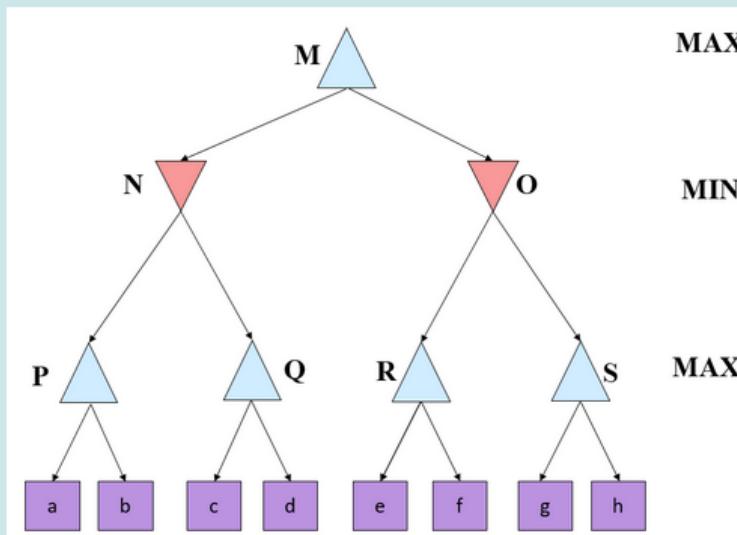
soluție

Stare	$h_1$
S	5
A	3
B	6
C	2
D	3
G	0

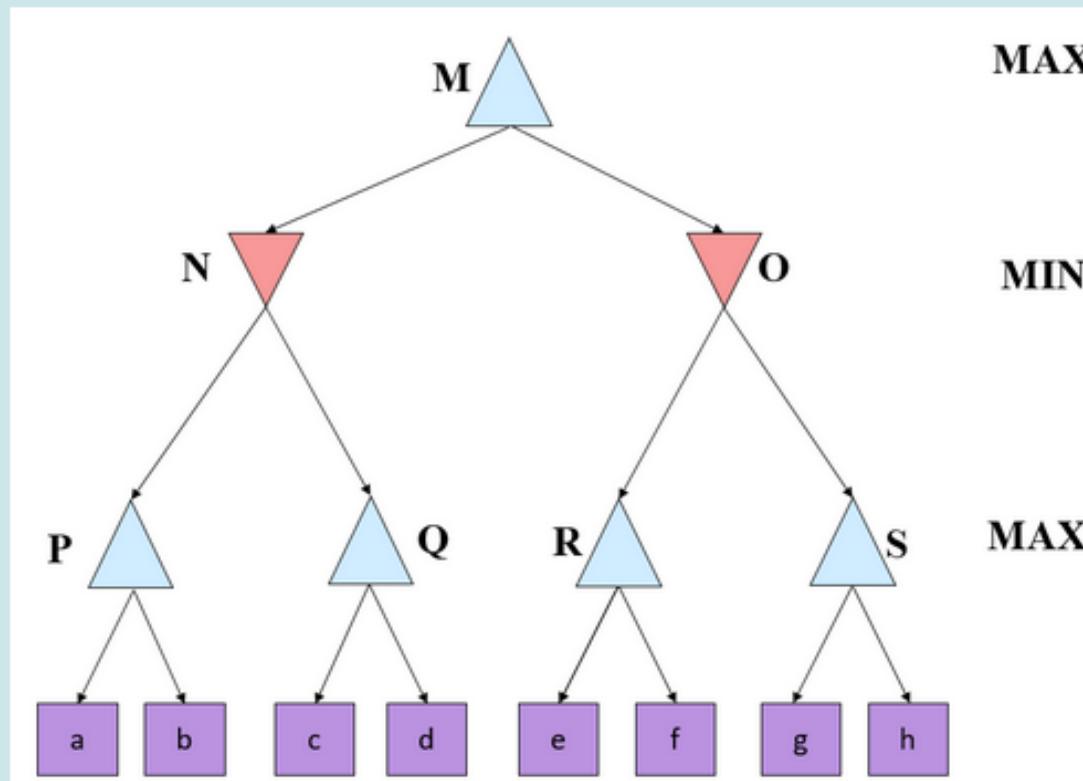
Consideram arborele de cautare din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN vrea sa minimizeze castigul lui MAX. Nodurile terminale au valorile a = 5, b = -10, c = 10, d = -2, e = 1, f = -1, g = 15, h = -1. Care sunt valorile Minimax ale nodurilor M, N, O, P, Q, R, S?



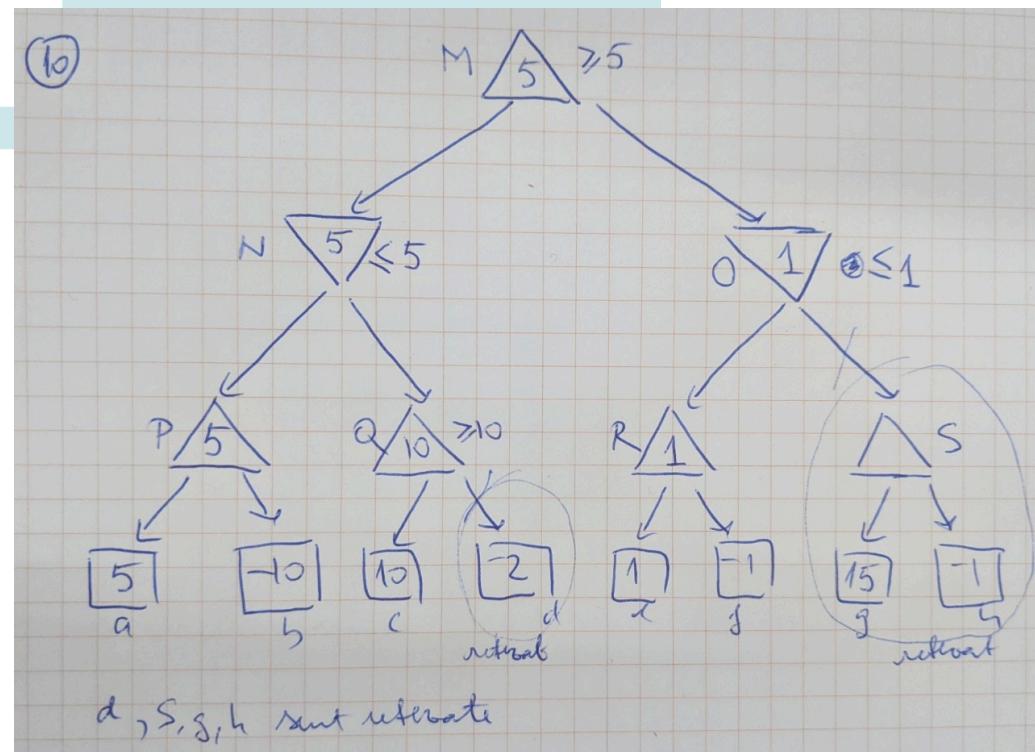
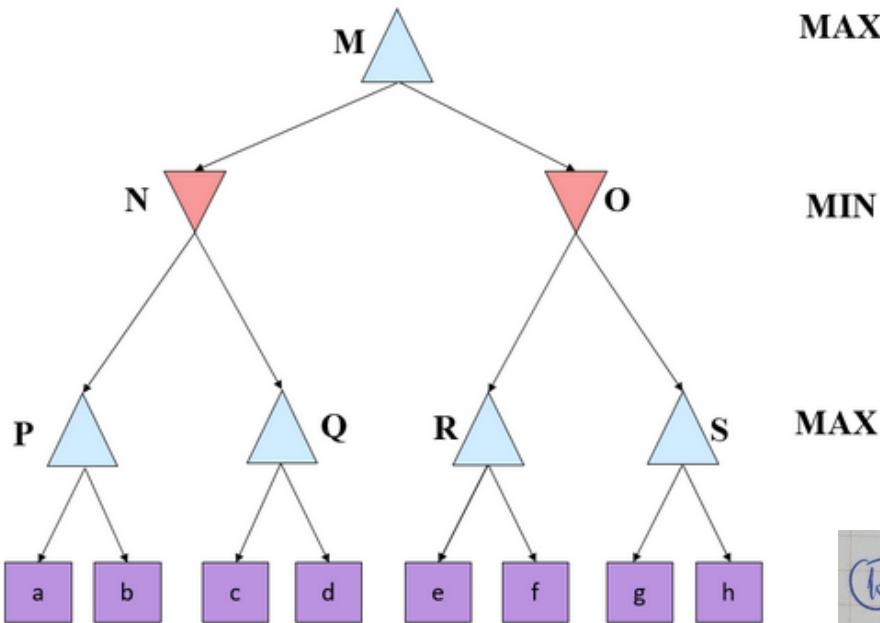
Consideram arborele de cautare din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN vrea sa minimizeze castigul lui MAX. Nodurile terminale au valorile a = 5, b = -10, c = 10, d = -2, e = 1, f = -1, g = 15, h = -1. Care sunt valorile Minimax ale nodurilor M, N, O, P, Q, R, S?



Consideram arborele de joc din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN aplică algoritmul alfa-beta în calcul valorii Minimax a radacinii M. Ce noduri sunt retezate (nu sunt vizitate)? Justificati raspunsul.



Consideram arborele de joc din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN va calcula valoarea minimă a radacinii M. Ce noduri sunt retezate (nu sunt vizitate)? Justificati raspunsul.



# Inteligentă Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023  
Cursul 13

# Recapitulare – cursul trecut

## 1. Căutare în jocuri cu incertitudine:

- căutare ExpectiMax

## 2. Examenul final

- organizarea examenului din acest an
- subiectele din anii trecuți
- rezolvările subiectelor din anii trecuți

# Examen iarnă - evaluare și notare

Nota = min(Curs + Laborator + Proiect&Teme + BonusLab, 10)  
4p                3p                3p                maxim 1p

- Examen Curs (4 puncte) – scris, în sesiune, 2 ore + eventual bonus de la curs (nu puteți depăși 4p)
- Test Laborator (3 puncte) – în sesiune (în aceeași zi cu Examen Curs), 2 ore;
- Proiect (1,5 puncte) – proiect la prima parte (învățare automată, îl primiți în săptămâna 4), termen limită de predare – săptămâna 6, prezentarea proiectului în săptămâna 7.
- Teme (1,5 puncte) – teme la partea a doua (căutare informată și neinformată), prezentare în cadrul laboratorului.

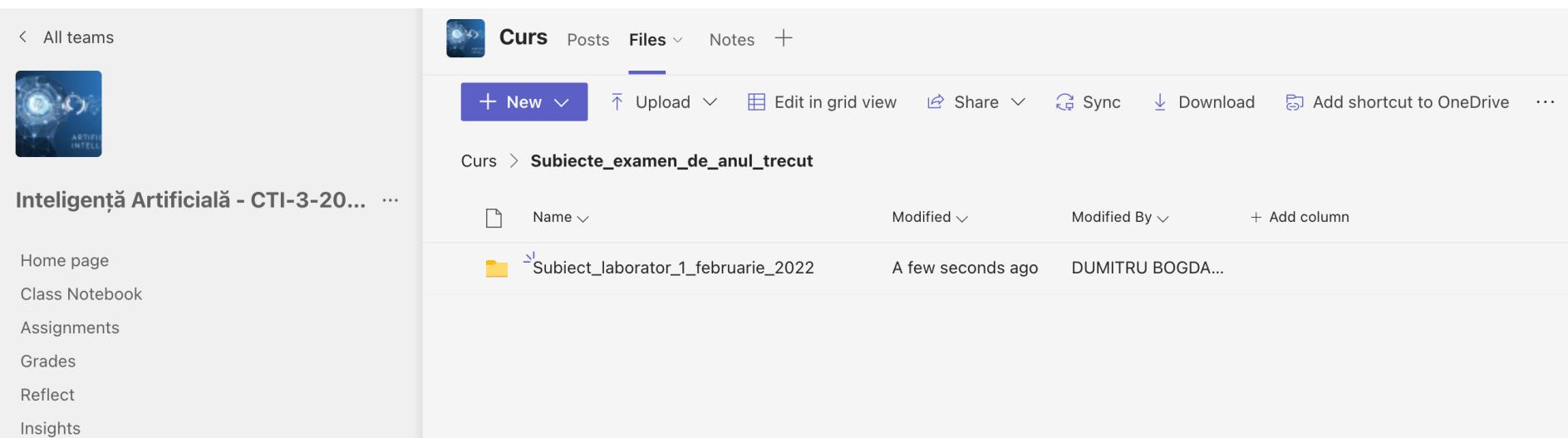
Nu există praguri, note minime impuse. 4,99 înseamnă restanță.

# Organizarea examenului din acest an

- Examenul scris:
  - data 2 februarie, intre orele 9-11 in amf. Pompeiu si Titeica
  - open books vs closed books?
- Testul de laborator:
  - data 2 februarie, intre orele 12-14 in salile amf. Pompeiu si Titeica
  - veți primi un subiect cu două probleme (una din prima parte – învățare automată, una din a doua parte – căutare informată și neinformată)
  - aveți voie cu un stick cu resurse
  - nu aveți voie cu laptop, toată lumea dă examen pe calculatoarele din FMI
  - acces la internet restrictionat

Inteligenta artificiala  
Conf.dr. Alexe B.  
Ora 9.00 – 11.00 Scris  
Amf. Pompeiu si Titeica  
Ora 12.00-14.00 – Laborator  
Salile 201, 204, 303, 308

# Organizarea examenului din acest an



The screenshot shows a Microsoft Teams interface for a channel named 'Curs'. The left sidebar lists team members and channels: 'All teams', 'Inteligentă Artificială - CTI-3-20...', 'Home page', 'Class Notebook', 'Assignments', 'Grades', 'Reflect', and 'Insights'. The main area displays a file list titled 'Subiecte\_examen\_de\_anul\_trecut'. The list includes a single item: 'Subiect\_laborator\_1\_februarie\_2022' (modified 'A few seconds ago' by 'DUMITRU BOGDA...'). The top navigation bar includes 'Posts', 'Files', 'Notes', and various sharing and sync options.

O să afișăm până pe 1 februarie situația fiecărui student (notă proiect Kaggle, notă teme partea a doua, bonus laborator, bonus curs)

# Subiectul din 27 ianuarie 2020

## Examen la disciplina Inteligență Artificială Seria 35, 27 ianuarie 2020

### Subiectul 1. (3,5 puncte)

Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

- considerăm perceptronul din Figura 1a cu ponderile  $w_1$  și  $w_2$  care ponderează intrările  $x_1$  (abscisa) și  $x_2$  (ordonata) a unui punct, iar ponderea  $w_0$  reprezintă deplasarea (bias-ul). Funcția de activare a perceptronului este funcția  $\phi(x) = \text{hardlim}(x)$ , unde  $\text{hardlim}(x) = 1$ , dacă  $x \geq 0$  și  $\text{hardlim}(x) = 0$  dacă  $x < 0$ . Determinați matricea de confuzie a perceptonului cu ponderile  $w_0 = -2, w_1 = 1, w_2 = 1$  care clasifică exemple din mulțimea  $S$ . **(1 punct)**
- mulțimea de antrenare  $S$  nu este liniar separabilă. Totuși, dacă adăugăm la fiecare vector a treia coordonată, egală cu eticheta exemplului, obținem o nouă mulțime de antrenare care acum este liniar separabilă. Arătați acest lucru, dând exemplu de un clasificator liniar care separă perfect exemplele de antrenare din noua mulțime. Puteți folosi acest clasificator liniar pentru a eticheta exemple din mulțimea  $T$ ? **(1 punct)**
- construiți o rețea neuronală de perceptri care să învețe perfect mulțimea de antrenare. Etichetați punctele din mulțimea  $T$  folosind rețeaua astfel construită. **(1,5 puncte)**

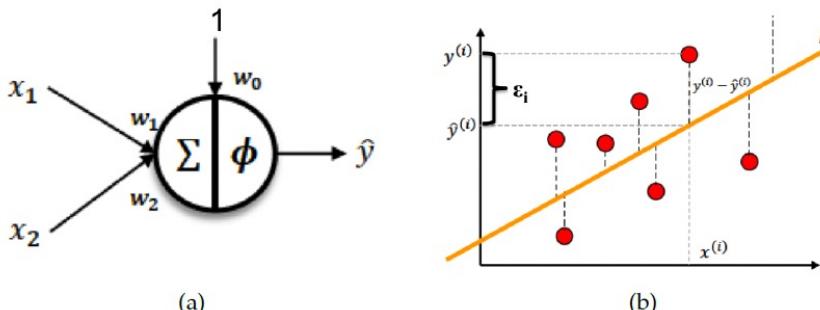
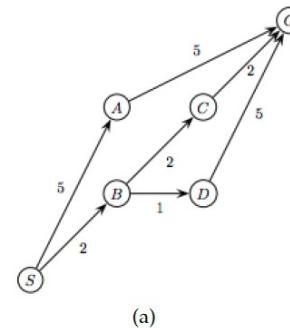


Figura 1: a. Arhitectura unui perceptron; b. Problema de regresie liniară prezentată la curs și evidențierea reziduuului  $\epsilon_i$ .



Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0, h_1, h_2$ .

### Subiectul 2. (1 punct)

Fie  $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$  o mulțime cu  $m$  exemple de antrenare, unde  $x^{(i)}, y^{(i)}$  sunt numere reale pentru  $i = 1, \dots, m$ . Regresia liniară simplă modeleză legătura dintre variabila independentă  $x$  și variabila dependentă  $y$  folosind dreapta de ecuație  $h(x) = w_0 + w_1 \cdot x$ . Parametrii  $w_0$  și  $w_1$  se determină folosind tehnica metodei celor mai mici pătrate aplicată pe mulțimea de antrenare  $E$ . La curs a fost arătat că parametrii optimi au valorile următoare:

$$w_0 = \frac{1}{m} \sum_{i=0}^m y^{(i)} - w_1 \cdot \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$w_1 = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2},$$

unde  $\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$ , iar  $\bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$ .

Pentru fiecare exemplu de antrenare din  $E$  notăm cu  $\epsilon_i$  diferența dintre răspunsul corect  $y^{(i)}$  și predicția  $\hat{y}^{(i)}$  (vedeți acest lucru ilustrat și în Figura 1b). Cantitatea  $\epsilon_i$  se mai numește rezidu. Arătați că suma tuturor reziduurilor este nulă, adică:

$$\sum_{i=1}^m \epsilon_i = 0.$$

### Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a. S este starea initială iar G este starea scop. O strategie de căutare care pornește din starea S și vrea să ajungă la starea G explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea A va fi explorată înaintea stării B). Considerăm euristicile  $h_0, h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (a) care este soluția problemei de căutare folosind strategiile de căutare în lățime și căutare în adâncime? **(0,5 puncte)**
- (b) care din euristicile  $h_0, h_1, h_2$  din Figura 2b sunt admisibile? Justificați răspunsul **(0,5 puncte)**
- (c) care este soluția returnată de algoritmul  $A^*$  pentru fiecare heuristică  $h_0, h_1, h_2$  în parte? Specificați soluțiile parțiale construite până la obținerea soluției finale. **(0,5 puncte)**
- (d) care este soluția returnată de algoritmul Greedy ce folosește heuristică  $h_1$ ? Specificați soluțiile parțiale construite până la obținerea soluției finale. **(0,5 puncte)**
- (e) care este soluția returnată de algoritmul de căutare uniformă după cost (UCS)? Specificați soluțiile parțiale construite până la obținerea soluției finale. **(0,5 puncte)**

#### **Subiectul 4. (2 puncte)**

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3b). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Căștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (a) Care este numărul de mutări maxim (= factorul de ramificare) pe care îl are la dispoziție fiecare jucător în orice moment al jocului? Justificați răspunsul. **(0,25 puncte)**
- (b) Considerăm că după 9 mutări ajungem cu jocul în starea dată de Figura 3b. Jucătorul 0 este acum la mutare. Desenați arborele de joc asociat, considerând drept rădăcină starea actuală a jocului. **(0,75 puncte)**
- (c) considerăm jucătorul X ca fiind MAX și jucătorul 0 ca fiind MIN. Funcția de utilitate asociată nodurilor terminale este următoarea: dacă jucătorul X câștigă atunci el

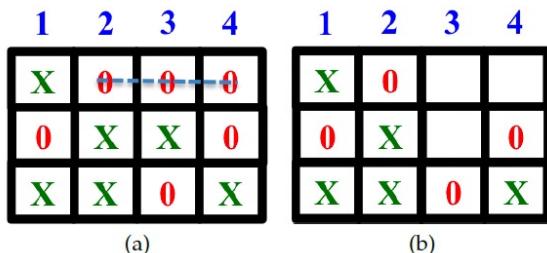


Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea  $-1$  (căștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

primește  $k$  puncte iar jucătorul 0 pierde  $k$  puncte, dacă jocul se termină la egalitate atunci fiecare are 0 puncte. Valoarea  $k$  atunci când jucătorul X câștigă este stabilită astfel:  $k = 3$  puncte pentru o linie realizată din simboluri dispuse pe diagonală,  $k = 2$  pentru o linie realizată din simboluri dispuse pe verticală și  $k = 1$  pentru o linie realizată din simboluri dispuse pe orizontală. Valoarea  $k$  atunci când jucătorul 0 câștigă pe cazurile enumerate anterior este  $-3, -2, -1$ . Folosind algoritmul Minimax etichetați arborele de joc de la punctul anterior cu valorile minimax asociate fiecărui nod. **(0,5 puncte)**

- (d) Folosiți algoritmul de retezare Alfa-Beta pentru accelerarea calcului valorilor Minimax. Ce noduri din arbore de joc nu vor fi explorate? Puteți reordona fiile fiecărui subarbore pentru a maximiza numărul de noduri care nu vor fi explorate de algoritmul Alfa-Beta? Justificați răspunsul. **(0,5 puncte)**

**Oficiu: 1 punct. Timp de lucru: 2 ore.**

# Cursul de azi

## 1. Examenul final

- rezolvările subiectelor din anii trecuți

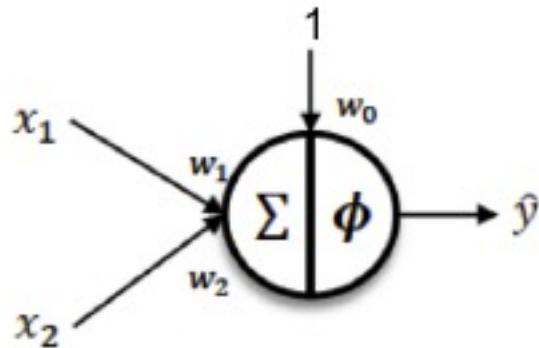
## 2. Prezentare curs optional CAVA

# Subiectul 1 a - enunt

## Subiectul 1. (3,5 puncte)

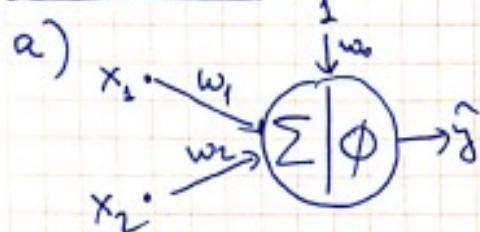
Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

- (a) considerăm perceptronul din Figura 1a cu ponderile  $w_1$  și  $w_2$  care ponderează intrările  $x_1$  (abscisa) și  $x_2$  (ordonata) a unui punct, iar ponderea  $w_0$  reprezintă deplasarea (bias-ul). Funcția de activare a perceptronului este funcția  $\phi(x) = \text{hardlim}(x)$ , unde  $\text{hardlim}(x) = 1$ , dacă  $x \geq 0$  și  $\text{hardlim}(x) = 0$  dacă  $x < 0$ . Determinați matricea de confuzie a perceptronului cu ponderile  $w_0 = -2$ ,  $w_1 = 1$ ,  $w_2 = 1$  care clasifică exemple din mulțimea  $S$ . (1 punct)



(a)

### Subiectul 1



$$\hat{y} = \text{hardlim}(w_0 + w_1 x_1 + w_2 x_2) =$$

$$= \begin{cases} 1, & w_0 + w_1 x_1 + w_2 x_2 \geq 0 \\ 0, & \text{altele} \end{cases}$$

$$w_0 = -2, w_1 = 1, w_2 = 1$$

$$\hat{y} = \begin{cases} 1, & x_1 + x_2 - 2 \geq 0 \\ 0, & x_1 + x_2 - 2 < 0 \end{cases}$$

$$x_1 = 0,$$

$$x_2 = 0,$$

$$y = 0$$

$$\hat{y} = 0$$

$$x_1 = 1$$

$$x_2 = 0$$

$$y = 1$$

$$\hat{y} = 1$$

$$x_1 = 2$$

$$x_2 = -1$$

$$y = -1$$

$$\hat{y} = 0$$

$$x_1 = 2$$

$$x_2 = 2$$

$$y = 1$$

$$\hat{y} = 1$$

$$x_1 = 5$$

$$x_2 = 3$$

$$y = 0$$

$$\hat{y} = 1$$

$$x_1 = -2$$

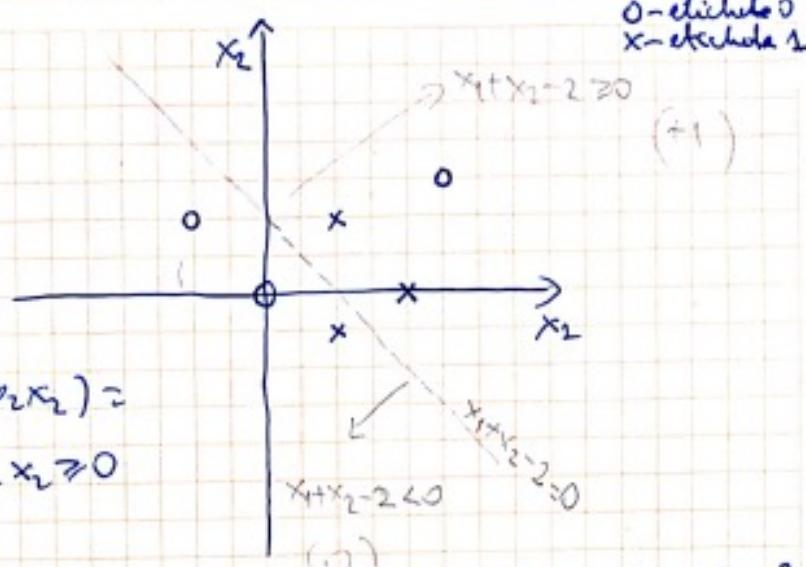
$$x_2 = 2$$

$$y = 0$$

$$\hat{y} = 0$$

elicitate  
proba

elicitate reală	$\hat{y}$	0	1
0	2	1	
1	1	2	



0 - elicitate 0  
x - elicitate 1

$$x_1 + x_2 - 2 \geq 0$$

(+1)

$$x_1 + x_2 - 2 \leq 0$$

(0)

$$S = \{(1, 0)^T, 0\}, \{(4, 0)^T, 1\}, \dots, \{(-2, 2)^T, 0\}\}$$

$$T = \{(1, 1)^T, (5, 5)^T\}$$

Natura de confuzii asociată perceptronului

# Subiectul 1b - enunt

## Subiectul 1. (3,5 puncte)

Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

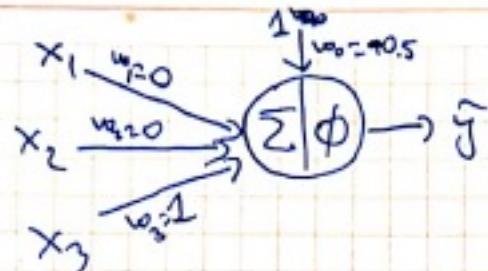
- (b) mulțimea de antrenare  $S$  nu este liniar separabilă. Totuși, dacă adăugăm la fiecare vector a treia coordonată, egală cu eticheta exemplului, obținem o nouă mulțime de antrenare care acum este liniar separabilă. Arătați acest lucru, dând exemplu de un clasificator liniar care separă perfect exemplele de antrenare din noua mulțime. Puteți folosi acest clasificator liniar pentru a eticheta exemple din mulțimea  $T$ ? (1 punct)

# Subiectul 1b - rezolvare

b)  $S_{\text{misi}} = \{( (9,0,0)^T, 0 ), ( (-4,0,1)^T, 1 ), ( (2,-1,1)^T, 1 ), ( (2,2,1)^T, 1 ), ( (5,3,0)^T, 0 ), ( (-2,2,0)^T, 0 ) \}$

Adaugăm practic dimensiunea  $x_3 = \text{înălțime}$ . Trăiești parțialele  
poziții palmei înălțimea 1, restul are înălțimea 0. Multimea multime  
de antrenare este liniștită separabilă, există o funcție de hiperplanuri  
(plane) care separă exemplul pozitiv de cel negativ.

O posibilitate este  $w_0 = 0, w_1 = 2, w_2 = 0, w_3 = 1 ; w_4 = -0.5$  :  $x_3 - 0.5 \geq 0$



$$\hat{y} = \text{hardim}(x_3 - 0.5) = \begin{cases} 1, & x_3 \geq 0.5 \\ 0, & x_3 < 0.5 \end{cases}$$

Ne putem folosi acest clasificator pentru a stabili exemple din  
multimea T între care lipsește o treia componentă (care reprezintă  
etichete pe care vrem să le prevedem).

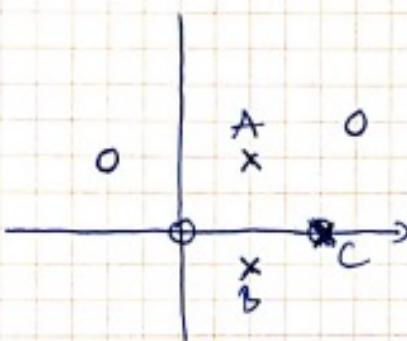
# Subiectul 1c - enunt

## Subiectul 1. (3,5 puncte)

Considerăm o problemă de clasificare binară în care exemplele de antrenare sunt reprezentate de vectori bidimensionali cu etichetele binare 0 și 1. Fie mulțimea de antrenare  $S = \{((0,0)^T, 0), ((4,0)^T, 1), ((2,-1)^T, 1), ((2,2)^T, 1), ((5,3)^T, 0), ((-2,2)^T, 0)\}$ . Fie mulțimea de testare  $T = \{(1,1)^T, (5,5)^T\}$ . Realizați următoarele:

- (c) construiți o rețea neuronală de perceptroni care să învețe perfect mulțimea de antrenare. Etichetați punctele din mulțimea  $T$  folosind rețeaua astfel construită. (1,5 puncte)

c)

 $A(2, 2), B(-2, -2), C(4, 0)$ 

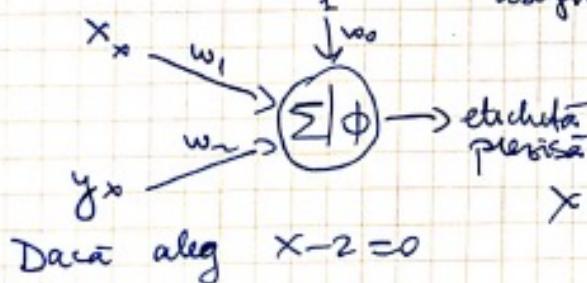
$$AB: \frac{x-x_A}{y-y_A} = \frac{x_B-x_A}{y_B-y_A}$$

$$\frac{x-2}{y-2} = \frac{-2-2}{-1-2} \Leftrightarrow x=2.$$

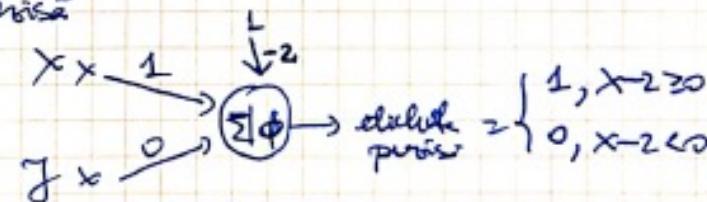
$$x-2=0.$$

Pentru dreapta  $AB$  am două posibile ecuații  $\begin{cases} x-2=0 \\ -x+2=0 \end{cases}$ .

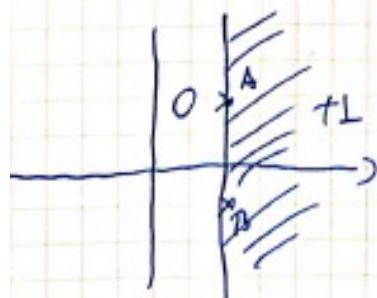
Pe care v-aleg? (soluția mea presupune a constura o relație care asigură 1 punct la tot punctele din triunghiul  $ABC$ ).



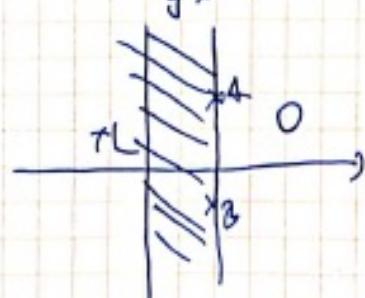
$$\text{Dacă aleg } x-2=0$$



$$\text{Dacă aleg } -x+2=0 \Rightarrow \begin{cases} 1, x-2 \geq 0 \\ 0, x-2 < 0 \end{cases}$$



Cazul  $x-2>0$



Cazul  $-x+2>0$ .

# Subiectul 1c - rezolvare

Vom se prelua punctul C(4,0) și vom se elibera l. Alegem varianta cu  $x+2=0$

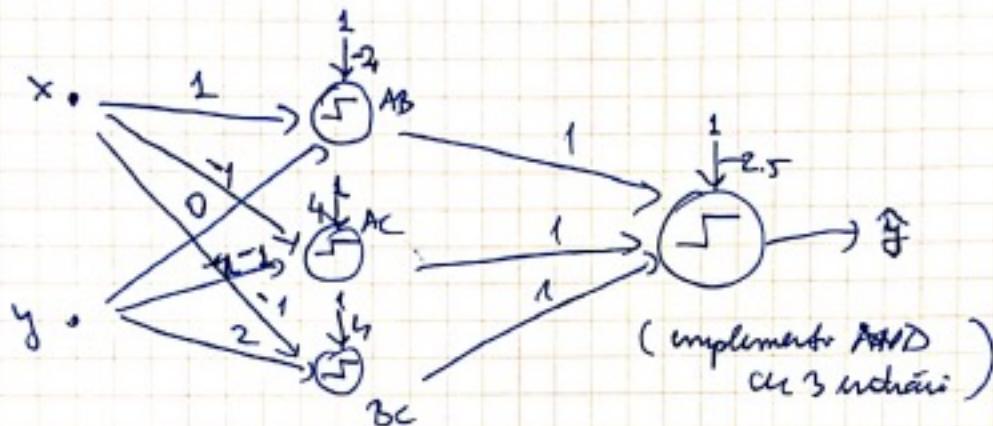
Analog, obținem ecuațiile partii dreptele AC și BC

$$AC: x+y-4=0$$

$$\boxed{-x-y+4=0}$$

$$BC: x-2y-4=0$$

$$\boxed{-x+2y+4=0}$$



$T = \{(1,1)^T, (5,5)^T\} : \text{Ambele puncte vor fi etablate ca } 0.$

# Subiectul 2 - enunt

## Subiectul 2. (1 punct)

Fie  $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$  o mulțime cu  $m$  exemple de antrenare, unde  $x^{(i)}, y^{(i)}$  sunt numere reale pentru  $i = 1, \dots, m$ . Regresia liniară simplă modelează legătura dintre variabila independentă  $x$  și variabila dependentă  $y$  folosind dreapta de ecuație  $h(x) = w_0 + w_1 \cdot x$ . Parametrii  $w_0$  și  $w_1$  se determină folosind tehnica metodei celor mai mici pătrate aplicată pe mulțimea de antrenare  $E$ . La curs a fost arătat că parametri optimi au valorile următoare:

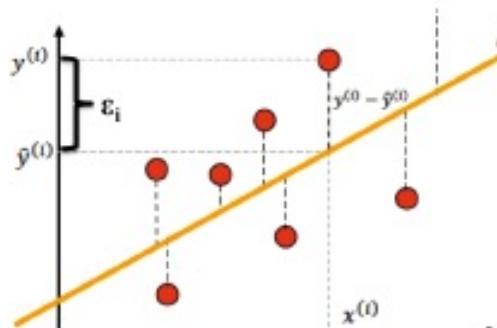
$$w_0 = \frac{1}{m} \sum_{i=0}^m y^{(i)} - w_1 \cdot \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$w_1 = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2},$$

unde  $\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$ , iar  $\bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$ .

Pentru fiecare exemplu de antrenare din  $E$  notăm cu  $\varepsilon_i$  diferența dintre răspunsul corect  $y^{(i)}$  și predicția  $\hat{y}^{(i)}$  (vedeți acest lucru ilustrat și în Figura 1b). Cantitatea  $\varepsilon_i$  se mai numește reziduu. Arătați că suma tuturor reziduurilor este nulă, adică:

$$\sum_{i=1}^m \varepsilon_i = 0.$$



(b)

# Subiectul 2 - rezolvare

Subiectul 2

$$\Sigma = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

Regressia liniara cu conditie dreapta de incerte  $h(x) = w_0 + w_1 x$

$$w_0 = \frac{1}{m} \sum_{i=1}^m y^{(i)} - w_1 \cdot \frac{1}{m} \sum_{i=1}^m x^{(i)} = \bar{y} - w_1 \cdot \bar{x}$$

$$w_1 = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x}) \cdot (y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2}$$

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}, \quad \bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$$

$$\varepsilon_i = y^{(i)} - \hat{y}^{(i)} = y^{(i)} - (w_0 + w_1 x^{(i)})$$

Astăzi că  $\sum_{i=1}^m \varepsilon_i = 0$ .

$$\sum_{i=1}^m \varepsilon_i = \sum_{i=1}^m (y^{(i)} - w_0 - w_1 x^{(i)}) = \sum_{i=1}^m y^{(i)} - m \cdot w_0 - w_1 \cdot \sum_{i=1}^m x^{(i)}$$

$$= m \cdot \bar{y} - m(\bar{y} - w_1 \cdot \bar{x}) - w_1 \cdot m \cdot \bar{x}$$

$$= m \bar{y} - m \bar{y} + m w_1 \cdot \bar{x} - w_1 \cdot m \cdot \bar{x}$$

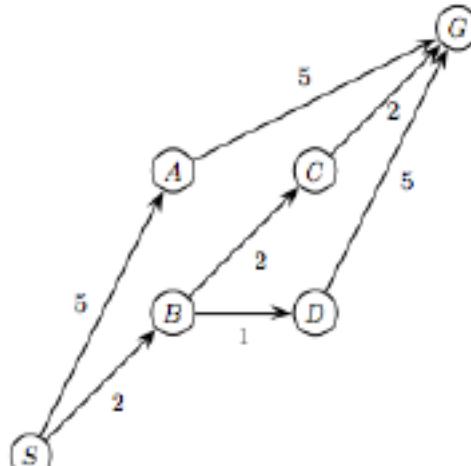
$$= 0.$$

# Subiectul 3a – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (a) care este soluția problemei de căutare folosind strategiile de căutare în lățime și căutare în adâncime? (0,5 puncte)



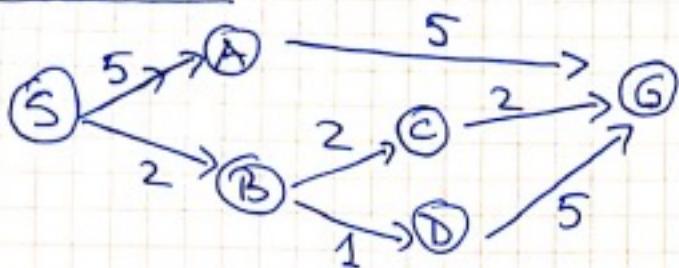
(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

Solutie 3



a)  $F = \{S\} \rightarrow$  frontieră initială  
(contine soluția parțială)

Pas 1:  $F = \{ \underline{S \rightarrow A}, S \rightarrow B \}$

(am expandat nodul S în ordine lexicografică  
pt căutare în labirint)

Pas 2:  $F = \{ \underline{\begin{matrix} S \rightarrow A \rightarrow G \\ \text{solutie} \end{matrix}}, S \rightarrow B \}$

Solutie căutată în labirint este  $S \rightarrow A \rightarrow G$

Căutarea în adâncime:

$F = \{S\}$

Pas 1:  $F = \{ \underline{S \rightarrow A}, S \rightarrow B \}$

$F = \{ \underline{\begin{matrix} S \rightarrow A \rightarrow G \\ \text{solutie} \end{matrix}}, S \rightarrow B \}$

Solutie căutată în adâncime este  $S \rightarrow A \rightarrow G$

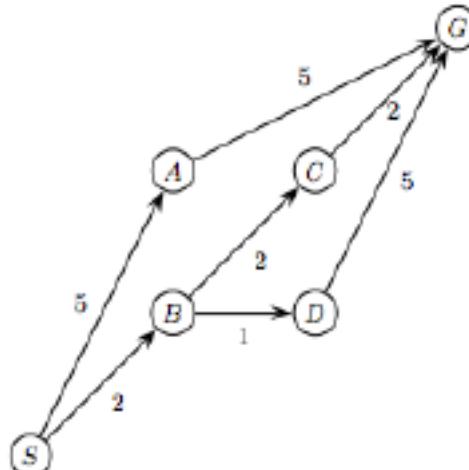
Frontieră	$h_0$	$h_1$	$h_2$	$h^*$
S	0	5	6	6
A	0	3	5	5
B	0	4	2	4
C	0	2	5	2
D	0	5	3	5
G	0	0	0	0

# Subiectul 3b – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (b) care din euristicile  $h_0$ ,  $h_1$ ,  $h_2$  din Figura 2b sunt admisibile? Justificați răspunsul (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3b – rezolvare

b) Curiștici  $h$  este admisibilă dacă ea subestimează peisajul mod în următorul nod al nodului  $n$  către cel mai apropiat nod scop  $0 \leq h(n) \leq h^*(n)$ , unde  $h^*(n)$  este costul real către cel mai apropiat nod scop.

Calculăm  $h^*$  pt fiecare nod:

$$h^*(S)=6, h^*(A)=5, h^*(B)=4, h^*(C)=2, h^*(D)=5, h^*(G)=0$$

Adaugăm coloana  $h^*$  în tabelul de sus.

Observăm că nodurile  $h_2$  sunt curiștici admisibile, dar  $h_2$  nu este admisibilă întrucât  $h_2(C)=5 > h_2^*(C)=2$ .

-5-

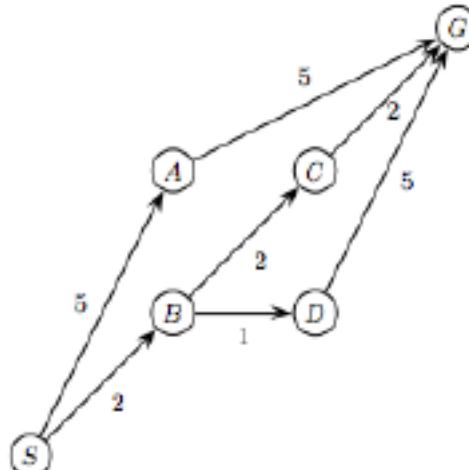
Stare	$h_0$	$h_1$	$h_2$	$h^*$
S	0	5	6	6
A	0	3	5	5
B	0	4	2	4
C	0	2	5	2
D	0	5	3	5
G	0	0	0	0

# Subiectul 3c – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (c) care este soluția returnată de algoritmul  $A^*$  pentru fiecare heuristică  $h_0$ ,  $h_1$ ,  $h_2$  în parte?  
Specificați soluțiile parțiale construite până la obținerea soluției finale. (0,5 puncte)



Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3c – rezolvare

c) Pentru h(x) ( $\Rightarrow$  existăca banală  $A^x$  se comportă ca UCS)

$$f(n) = g(n) + h(n)$$

$\downarrow$  cu exact  $\downarrow$  cu estimare pe baza existenței  
cât am sănătății că rămășăte să fac

$$\begin{aligned} P_{n,0}: F &= \{S\} \\ &\underline{f(S) \geq 0} \end{aligned}$$

$$P_{n,1}: F = \left\{ S \xrightarrow{B}, S \xrightarrow{A} \right\}$$

$f(S \xrightarrow{B}) = 2 + 0 = 2$        $f(S \xrightarrow{A}) = 5 + 0 = 5$

$$P_{n,2}: F = \left\{ S \xrightarrow{B \rightarrow D}, S \xrightarrow{B \rightarrow C}, S \xrightarrow{A} \right\}$$

$f(n) = 3$        $f(n) = 4$        $f(n) = 5$

$$P_{n,3}: F = \left\{ S \xrightarrow{B \rightarrow D \rightarrow G}, S \xrightarrow{B \rightarrow C}, S \xrightarrow{A} \right\}$$

$f(n) = 8$        $f(n) = 6$        $f(n) = 5$

$$P_{n,4}: F = \left\{ S \xrightarrow{B \rightarrow D \rightarrow G}, S \xrightarrow{B \rightarrow C \rightarrow G}, S \xrightarrow{A} \right\}$$

$f(n) = 8$        $f(n) = 6$        $f(n) = 5$

$$P_{n,5}: F = \left\{ S \xrightarrow{B \rightarrow D \rightarrow G}, S \xrightarrow{B \rightarrow C \rightarrow G}, S \xrightarrow{A \rightarrow G} \right\}$$

$f(n) = 8$        $f(n) = 6$        $f(n) = 10$

Soluția este  $S \xrightarrow{B \rightarrow C \rightarrow G}$

# Subiectul 3c – rezolvare

Pentru h<sub>1</sub>

Pas 0:  $F = \{S\}$   
 $f(n) = 2 + 5 = 5$

Pas 1:  $F = \left\{ S \rightarrow A, S \rightarrow B \right\}$   
 $f(n) = 5 + 3 = 8$ ,  $\underline{f(n_1) = 2 + 4 = 6}$

Pas 2:  $F = \left\{ S \rightarrow A, S \rightarrow B \rightarrow C, S \rightarrow B \rightarrow D \right\}$   
 $f(n) = 8$ ,  $\underline{f(n_1) = 4 + 2 = 6}$ ,  $\underline{f(n_1) = 3 + 5 = 8}$

Pas 3:  $F = \left\{ S \rightarrow A, S \rightarrow B \rightarrow C \rightarrow G, S \rightarrow B \rightarrow D \right\}$   
 $f(n) = 8$ ,  $\underline{f(n_1) = 6 + 0 = 6}$ ,  $\underline{f(n_1) = 3 + 5 = 8}$   
Solutie

Solutia este  $S \rightarrow B \rightarrow C \rightarrow G$

# Subiectul 3c – rezolvare

Pentru h<sub>2</sub>

Pass 0:  $F = \{S\}$ ,  
 $f(m) = 0 + 6 = 6$

Pass 1:  $F = \{S \rightarrow A, S \rightarrow B\}$ ,  
 $f(m) = 2 + 5 = 10, f(m) = 2 + 2 = 4$

Pass 2:  $F = \{S \rightarrow A, S \rightarrow B \rightarrow C, S \rightarrow B \rightarrow D\}$ ,  
 $f(m) = 10, f(m) = 4 + 5 = 9, f(m) = 3 + 3 = 6$

Pass 3:  $F = \{S \rightarrow A, S \rightarrow B \rightarrow C, S \rightarrow B \rightarrow D \rightarrow G\}$ ,  
 $f(m) = 10, f(m) = 9, f(m) = 8 + 10 = 18$

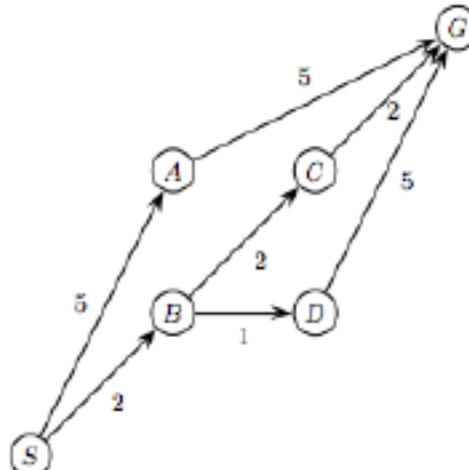
Soluție astă  $S \rightarrow B \rightarrow D \rightarrow G$ .

# Subiectul 3d – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (d) care este soluția returnată de algoritmul de căutare Greedy ce folosește euristica  $h_1$ ? Specificați soluțiile parțiale construite până la obținerea soluției finale. (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3d – rezolvare

d) Greedy cu  $h_1$

Pas 0:  $F = \{S\}$   
 $h_1(S) = 5$

Pas 1:  $F = \left\{ S \xrightarrow{\underline{h_1(w)=3}}, S \xrightarrow{\underline{h_1(w)=4}} A, B \right\}$

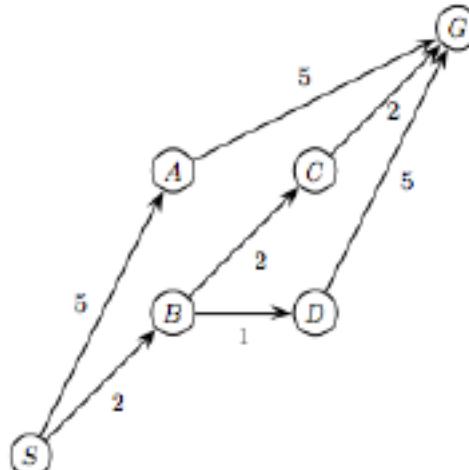
Pas 2:  $F = \left\{ S \xrightarrow{\underline{h_2(w)=0}} A \xrightarrow{\underline{h_2(w)=4}} G, S \xrightarrow{\underline{h_2(w)=4}} B \right\}$   
Solutie

# Subiectul 3e – enunț

## Subiectul 3. (2,5 puncte)

Considerăm problema de căutare dată de graful din Figura 2a.  $S$  este starea inițială iar  $G$  este starea scop. O strategie de căutare care pornește din starea  $S$  și vrea să ajungă la starea  $G$  explorează stările într-o anumită ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (spre exemplu în căutarea în lățime, starea  $A$  va fi explorată înaintea stării  $B$ ). Considerăm euristicile  $h_0$ ,  $h_1$  și  $h_2$  date de tabelul din Figura 2b. Realizați următoarele:

- (e) care este soluția returnată de algoritmul de căutare uniformă după cost (UCS)? Specificați soluțiile parțiale construite până la obținerea soluției finale. (0,5 puncte)



(a)

Stare	$h_0$	$h_1$	$h_2$
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(b)

Figura 2: a. Graful asociat problemei de căutare; b. Euristicile  $h_0$ ,  $h_1$ ,  $h_2$ .

# Subiectul 3e – rezolvare

e)  $UCS = A^*$  cu  $h_0$  = punctul c) prima parte

c) Pentru  $h_0$  ( $\Rightarrow$  există banală  $A^*$  se comportă ca UCS)

$$f(n) = g(n) + h(n)$$

$\downarrow$  cost exact       $\downarrow$  cost estimat pe baza existenței  
cât am sănăt      către care să fac

$$\begin{aligned} P_{A,0}: \quad F = & \{ S \} \\ & \underline{f(S)=0} \end{aligned}$$

$$P_{A,1}: \quad F = \{ S \xrightarrow{B} B, \quad S \xrightarrow{A} A \}$$

$\underline{f(S \xrightarrow{B} B) = 2 + 0 = 2}$        $\underline{f(S \xrightarrow{A} A) = 5 + 0 = 5}$

$$P_{A,2}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{C} C, \quad S \xrightarrow{A} A \}$$

$\underline{f(n)=3}$        $\underline{f(n)=4}$        $\underline{f(n)=5}$

$$P_{A,3}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{D} D \xrightarrow{G} G, \quad S \xrightarrow{B} B \xrightarrow{C} C, \quad S \xrightarrow{A} A \}$$

$\underline{f(n)=8}$        $\underline{f(n)=7}$        $\underline{f(n)=5}$

$$P_{A,4}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{D} D \xrightarrow{G} G, \quad S \xrightarrow{B} B \xrightarrow{C} C \xrightarrow{G} G, \quad S \xrightarrow{A} A \}$$

$\underline{f(n)=8}$        $\underline{f(n)=6}$        $\underline{f(n)=5}$

$$P_{A,5}: \quad F = \{ S \xrightarrow{B} B \xrightarrow{D} D \xrightarrow{G} G, \quad S \xrightarrow{B} B \xrightarrow{C} C \xrightarrow{G} G, \quad S \xrightarrow{A} A \xrightarrow{G} G \}$$

$\underline{f(n)=8}$        $\underline{f(n)=6}$        $\underline{f(n)=10}$

Soluția este  $S \xrightarrow{B} B \xrightarrow{C} C \xrightarrow{G} G$

# Subiectul 4a – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (a) Care este numărul de mutări maxim (= factorul de ramificare) pe care îl are la dispoziție fiecare jucător în orice moment al jocului? Justificați răspunsul. (0,25 puncte)

1	2	3	4
X	0	0	0
0	X	X	0
X	X	0	X

(a)

1	2	3	4
X	0		
0	X		0
X	X	0	X

(b)

Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea -1 (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

# Subiectul 4a – rezolvare

## Subiectul 4

- a) Numărul maxim de mutări (= factorul de ranificare) pe care îl are la dispoziție fiecare jucător = numărul de coloane libere. Inițial, toate coloanele sunt libere, deci  $b = 4$ , spașterea jucătorului să poată întâmpina ca numai o singură coloană să fie liberă, deci  $b = 1$ .

# Subiectul 4b – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (b) Considerăm că după 9 mutări ajungem cu jocul în starea dată de Figura 3b. Jucătorul 0 este acum la mutare. Desenați arborele de joc asociat, considerând drept rădăcină starea actuală a jocului. (0,75 puncte)

1	2	3	4
X	0	0	0
0	X	X	0
X	X	0	X

(a)

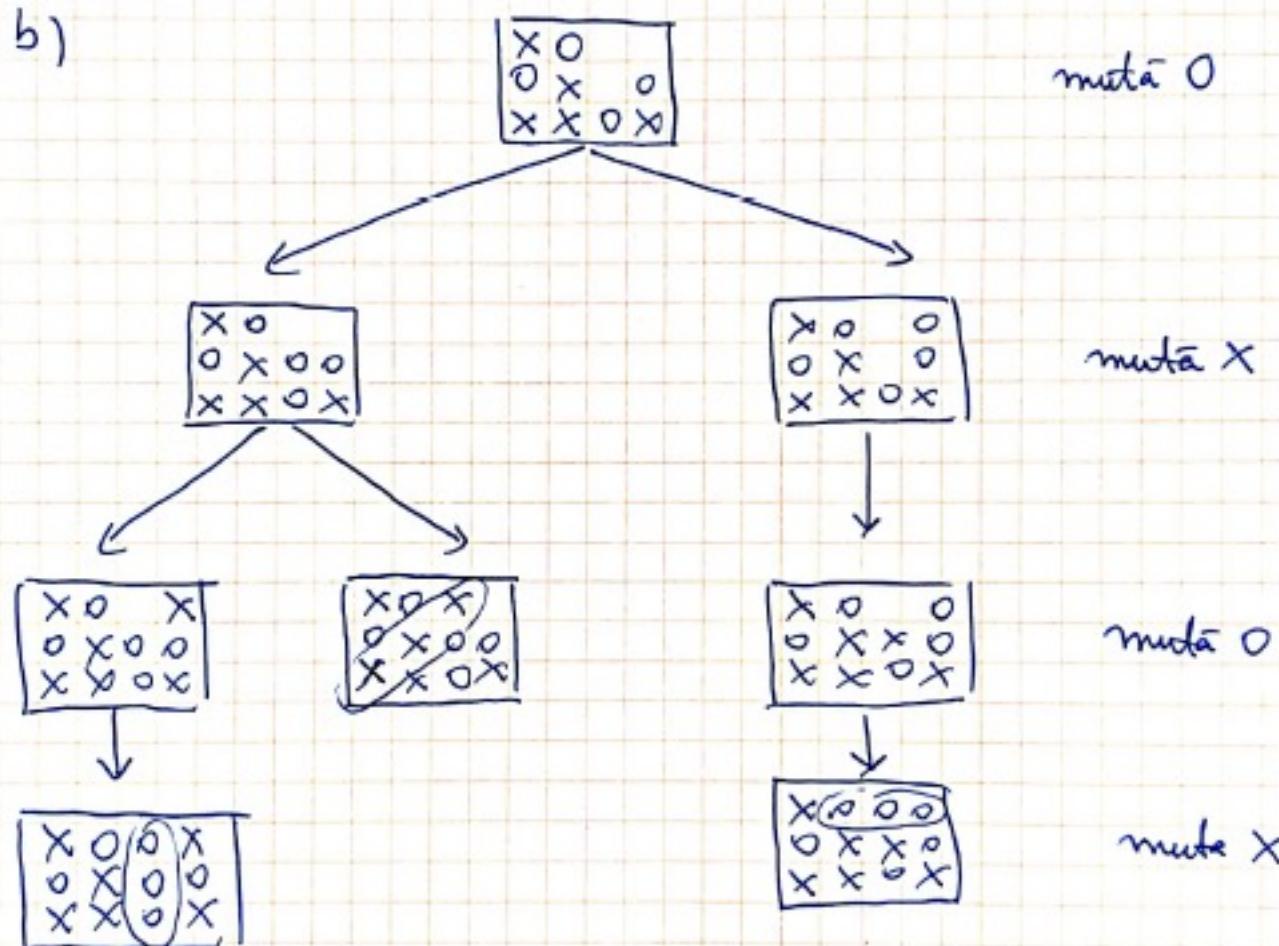
1	2	3	4
X	0		
0	X		0
X	X	0	X

(b)

Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea -1 (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

# Subiectul 4b – rezolvare

b)



# Subiectul 4c – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (c) considerăm jucătorul X ca fiind MAX și jucătorul 0 ca fiind MIN. Funcția de utilitate asociată nodurilor terminale este următoarea: dacă jucătorul X câștigă atunci el primește  $k$  puncte iar jucătorul 0 pierde  $k$  puncte, dacă jocul se termină la egalitate atunci fiecare are 0 puncte. Valoarea  $k$  atunci când jucătorul X câștigă este stabilită astfel:  $k = 3$  puncte pentru o linie realizată din simboluri dispuse pe diagonală,  $k = 2$  pentru o linie realizată din simboluri dispuse pe verticală și  $k = 1$  pentru o linie realizată din simboluri dispuse pe orizontală. Valoarea  $k$  atunci când jucătorul 0 câștigă pe cazurile enumerate anterior este  $-3, -2, -1$ . Folosind algoritmul Minimax etichetați arborele de joc de la punctul anterior cu valorile minimax asociate fiecărui nod. (0,5 puncte)

1	2	3	4
1	X	0	0
2	0	X	X
3	X	X	0
4			X

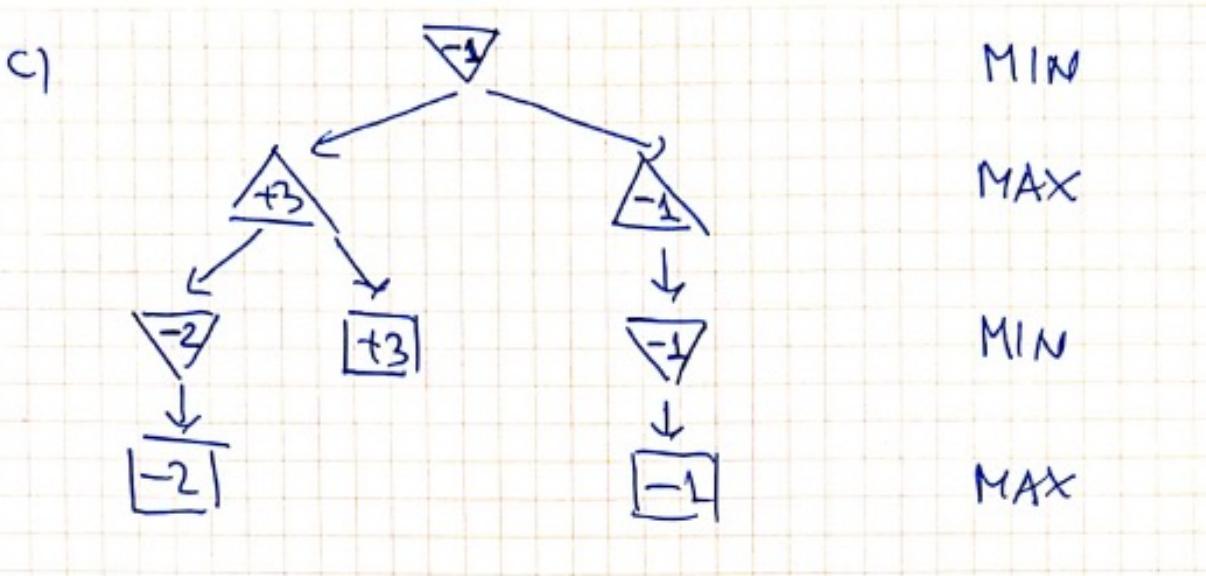
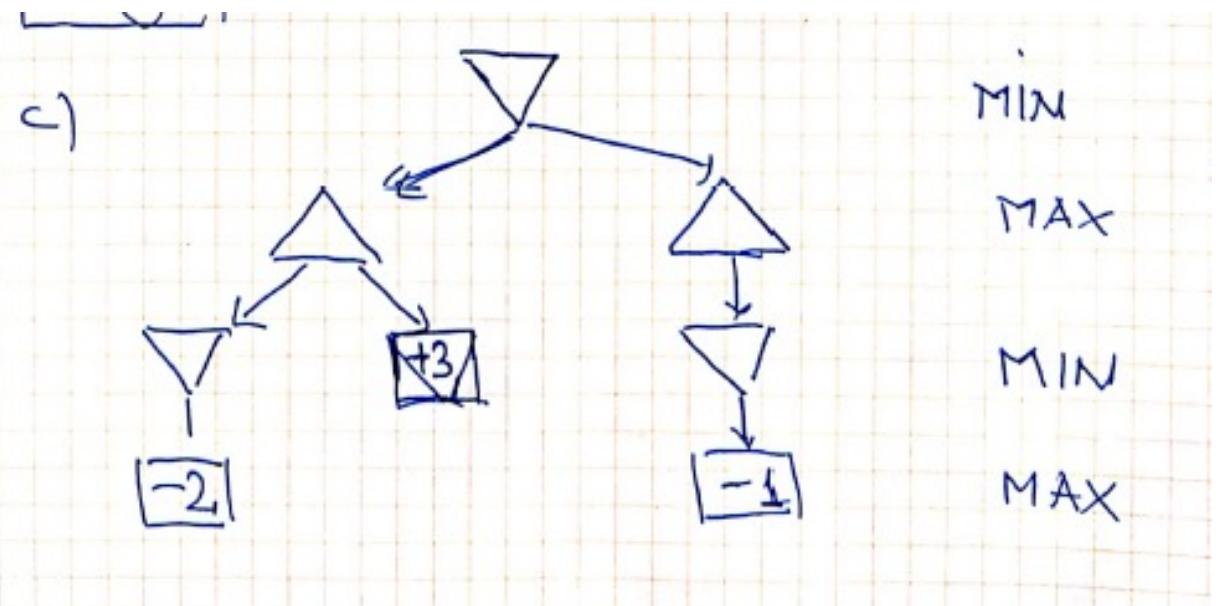
(a)

1	2	3	4
1	X	0	
2	0	X	
3	X	X	0
4			X

(b)

Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea  $-1$  (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

# Subiectul 4c – rezolvare



# Subiectul 4d – enunț

## Subiectul 4. (2 puncte)

În jocul Conectează-3, jucătorii X și 0 mută alternativ plasând simbolurile lor într-una din coloanele 1, 2, 3 sau 4 ale unui tablou cu 3 linii și 4 coloane. Simbolurile se acumulează unul peste altul (Figura 3a). O coloană în care au fost plasate 3 simboluri devine plină și prin urmare jucătorii nu mai pot plasa simboluri în ea. Câștigă jucătorul care realizează primul 3 simboluri dispuse pe orizontală, verticală sau diagonală. Dacă niciun jucător nu realizează acest lucru jocul se termină la egalitate. Jucătorul X mută primul.

- (d) Folosiți algoritmul de retezare Alfa-Beta pentru accelerarea calcului valorilor Minimax. Ce noduri din arbore de joc nu vor fi explorate? Puteți reordona fiile fiecărui subarbore pentru a maximiza numărul de noduri care nu vor fi explorate de algoritmul Alfa-Beta? Justificați răspunsul. (0,5 puncte)

1	2	3	4
X	0	0	0
0	X	X	0
X	X	0	X

(a)

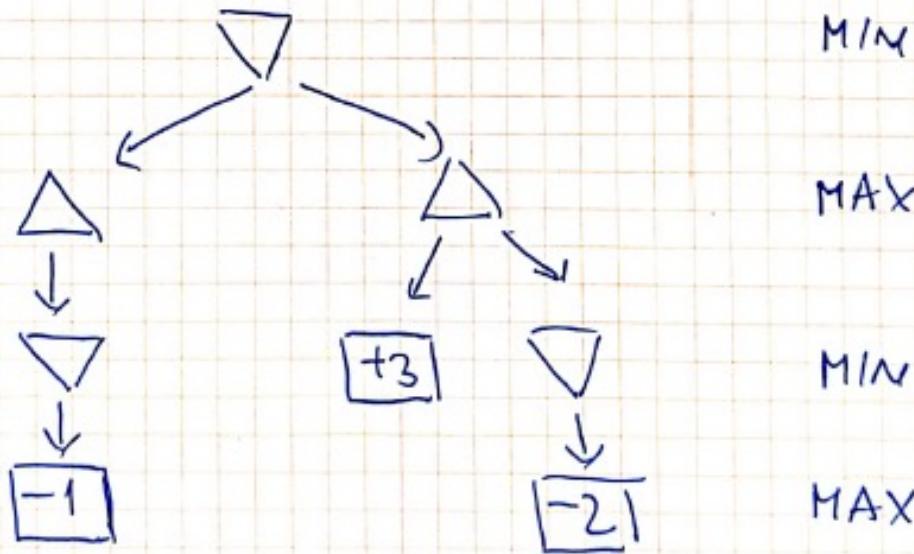
1	2	3	4
X	0		
0	X		0
X	X	0	X

(b)

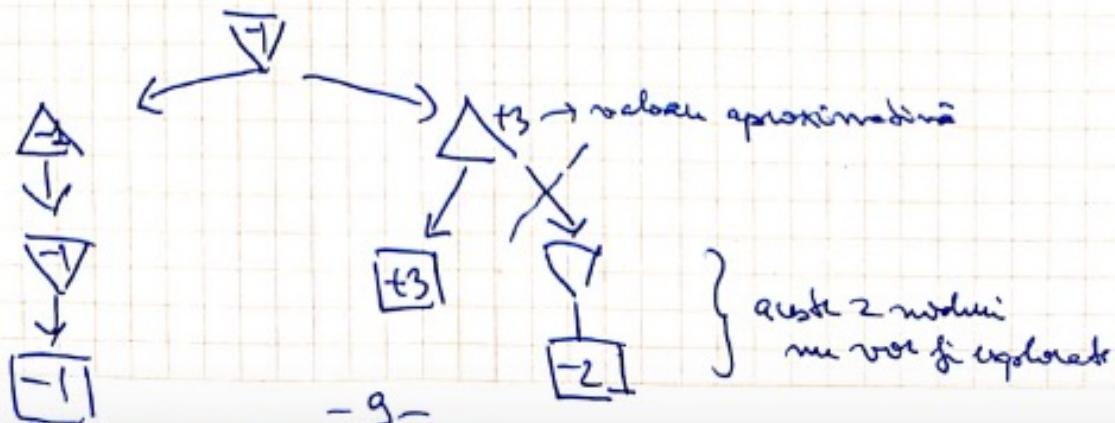
Figura 3: a. Exemplu de stare finală (nod de tip frunză în arborele de joc) cu utilitatea -1 (câștigă jucătorul 0 ce realizează o linie pe orizontală); b. Stare curentă în care se ajunge după 9 mutări, acum jucătorul 0 este la mutare.

d) Cu Alfa-Beta, în configurația actuală nu se poate face nicio reținere.

Rearanjăm arborele astfel:



Aplicam Alfa-Beta



**3** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), 0, (3,-2), 0, (5,-2), 0, (7,-1), 1, (7,2), 1, (7,4), 1\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspusul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

3 intrebare

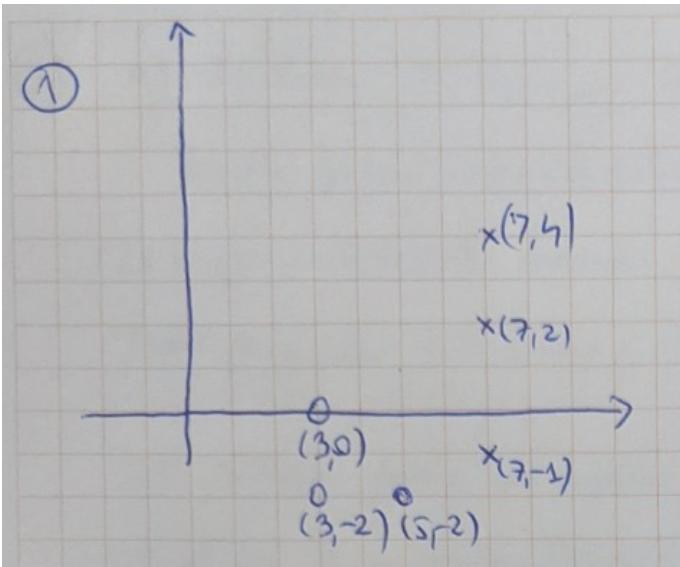
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.



3 intrebare

Nu a primit  
răspuns încă

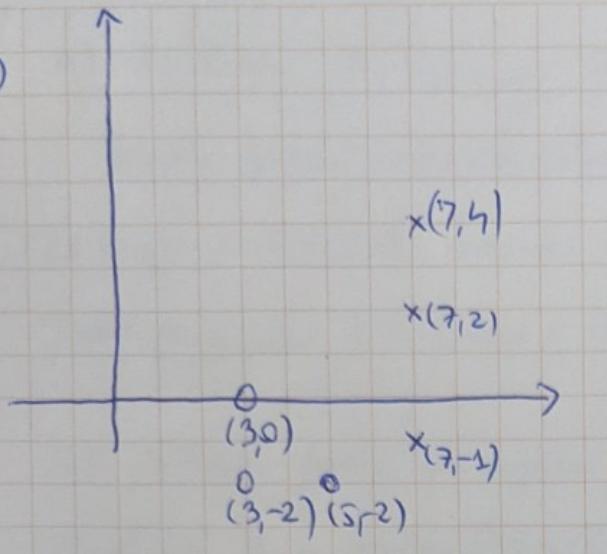
Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspusul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

①



1. Considerăm punctul  $(7,0)$ .

Cei mai apropiati 3 vecini sunt :

- $(7,-1)$  distanta 1 (distanta euclidiana)
- $(7,2)$  distanta 1
- $(5,-2)$  distanta  $\sqrt{8}$  ( $\sqrt{(7-5)^2 + (0-(-2))^2}$ )

3 intrebare

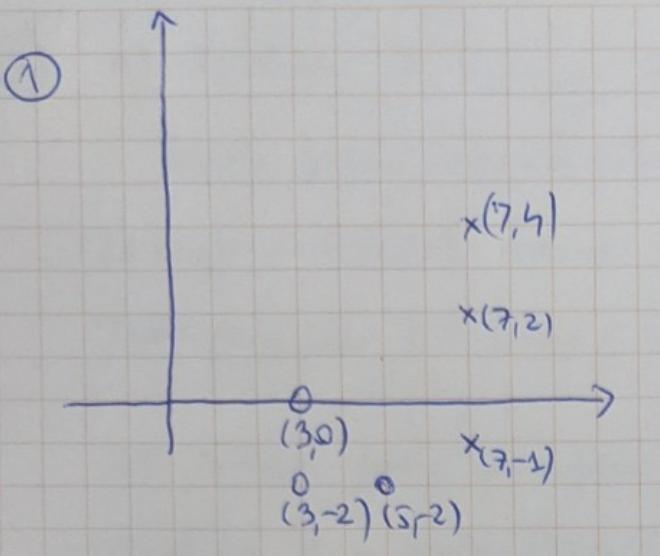
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu k=3. Fie multimea S de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), (5,-2), (7,-1), (7,2), (7,4)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.



2. Considerem punctul  $(3,0)$

Cei mai apropiati 3 vecini în lăsuță Manhattan sunt:

-  $(3,0)$  distanță Manhattan 0

-  $(3,-2)$  distanță Manhattan 2  $(|3-3| + |0-(-2)|)$

-  $(5,-2)$  distanță Manhattan 4  $(|3-5| + |0-(-2)|)$

3 intrebare

Nu a primit  
răspuns încă

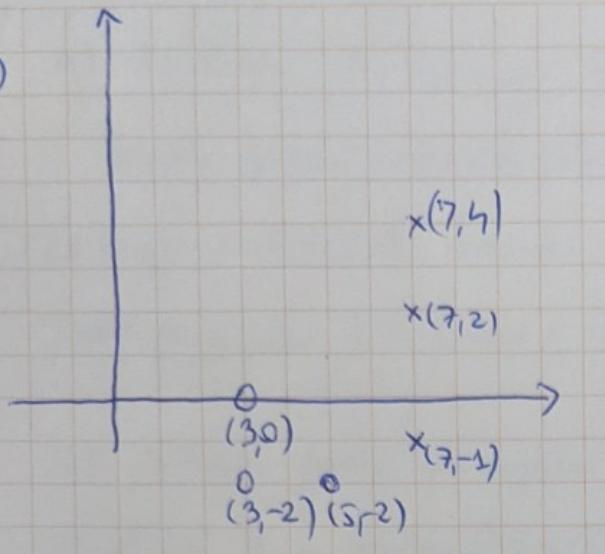
Marcat din 1,00

Întrebare cu  
flag

Consideram clasificatorul binar bazat pe metoda celor mai apropiati k-vecini cu  $k=3$ . Fie multimea  $S$  de exemple de antrenare etichetate cu 6 puncte din plan unde  $S = \{(3,0), (3,-2), ((5,-2),0), ((7,-1),1), ((7,2),1), ((7,4),1)\}$ , unde fiecare exemplu de antrenare e reprezentat de vectorul dat de coordonatele sale = (abscisa, ordonata) si de eticheta corespunzatoare (0 sau 1).

1. Dati exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare.
2. Dati exemplu de un punct care va fi clasificat cu eticheta 0 folosind distanta Manhattan. Justificati raspunsul scriind care sunt cei mai apropiati 3 vecini si scriind distantele corespunzatoare
3. Puteti da exemplu de un punct care va fi clasificat cu eticheta 1 folosind distanta euclidiana si cu eticheta 0 folosind distanta Manhattan? Daca răspunsul e pozitiv dati exemplul, altfel justificati de ce nu puteti da exemplul.

①



3. Punctul  $(5,1)$ :

- folosind distanta Euclidiana cei 3 vecini sunt  $(3,0)$  - etichetă 0  $(\sqrt{5})$   
 $(7,2)$  - etichetă 1  $(\sqrt{5})$   
 $(7,-1)$  - etichetă 1  $(\sqrt{8})$

$\Rightarrow$  va fi clasificat cu etichetă 1

- folosind distanta Manhattan cei 3 vecini sunt:  $(3,0)$  - etichetă 0  $(3)$   
 $(7,2)$  - etichetă 1  $(3)$   
 $(5,-2)$  - etichetă 0  $(3)$

$\Rightarrow$  va fi clasificat cu etichetă 0

**2** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

¶ Întrebare cu  
flag

Consideram problema clasificarii literelor mici (clasele ‘a’, ‘b’, …, ‘z’) scrise de mana din alfabetul englez (in total 26 de clase) pe baza modelului Naïve Bayes. Datele de antrenare constau in 100 de exemple de antrenare cu imagini 28 x 28 pixeli pentru fiecare litera in parte. In total aveti 2600 exemple de antrenare. Consideram ca stim probabilitatea a-priori de aparitie a fiecarei litere pe baza studiului frecventei de aparitie a literelor in limba engleza. Pentru estimarea probabilitatii likelihood folosim modelul Naïve Bayes care considera ca fiecare pixel din imagine are o distributie normala specifica pozitiei pixelului din imagine si clasei literei.

1. cati parametri trebuie sa invete modelul vostru pentru a putea clasifica o litera 28 x 28?
2. cum ati invata parametri corespunzatori pixelului de pe linia 14, coloana 14 (centrul imaginii) din clasa ‘a’?

2 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram problema clasificarii literelor mici (clasele 'a', 'b', ..., 'z') scrise de mana din alfabetul englez (in total 26 de clase) pe baza modelului Naïve Bayes. Datele de antrenare constau in 100 de exemple de antrenare cu imagini 28 x 28 pixeli pentru fiecare litera in parte. In total aveti 2600 exemple de antrenare. Consideram ca stim probabilitatea a-priori de aparitie a fiecarei litere pe baza studiului frecventei de aparitie a literelor in limba engleza. Pentru estimarea probabilitatii likelihood folosim modelul Naïve Bayes care considera ca fiecare pixel din imagine are o distributie normala specifica pozitiei pixelului din imagine si clasei literei.

1. cati parametri trebuie sa invete modelul vostru pentru a putea clasifica o litera 28 x 28?
2. cum ati invata parametri corespunzatori pixelului de pe linia 14, coloana 14 (centrul imaginii) din clasa 'a'?

② a) 26 litere 'a', 'b', 'c', ... 'z'

Pentru fiecare litera (in total 26) invatați puncte fizice sau pozitii din cel 784

( $28 \times 28$ ) și distribuția normală  $N(\mu, \sigma^2)$

Total parametri: 26 litere  $\times$  784 pozitii  $\times$  2 parametri

b) Selectați toate exemplul (în număr de 100) cu litera 'a'. Selectați valoarele pixelelor de pe poziție (14,14) în total 100 de valori. Calculați apoi media  $\mu$  și deviație standard  $\sigma$ .

**1** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

1 intrebare

Nu a primit  
răspuns încă

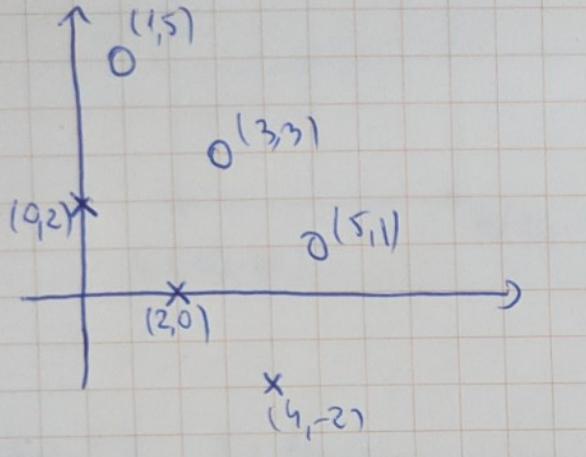
Marcat din 1,00

Întrebare cu  
flag

Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

③



1 intrebare

Nu a primit  
răspuns încă

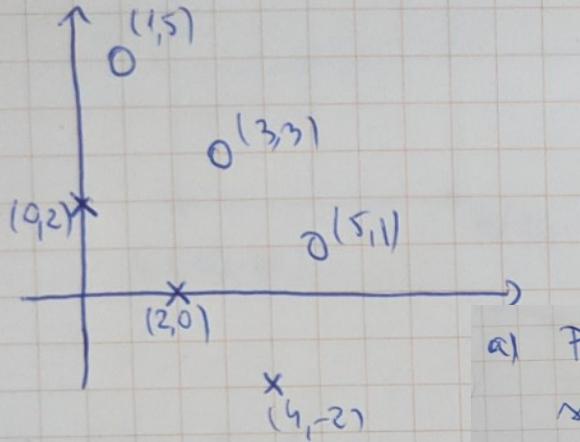
Marcat din 1,00

Întrebare cu  
flag

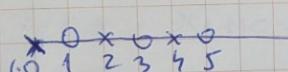
Fie  $P = \{((0,2),1), ((2,0),1), ((4,-2),1), ((1,5),0), ((3,3),0), ((5,1),0)\}$  o multime de 6 exemple de antrenare de puncte din plan. Fiecare punct are o abscisa si o ordonata si o eticheta (0 sau 1).

1. Este multimea  $P$  liniar separabila? Justificati raspunsul.
2. Daca proiectam fiecare punct pe prima coordonata (abscisa) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Ox$ ) liniar separabila? Justificati raspunsul.
3. Daca proiectam fiecare punct pe a doua coordonata (ordonata) este multimea de exemple de antrenare nou obtinuta (cu puncte de pe axa  $Oy$ ) liniar separabila? Justificati raspunsul.

③

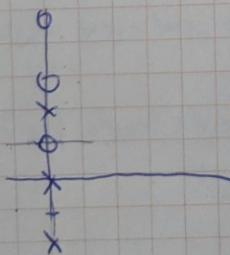


a)  $P$  este liniar separabila, spre exemplu dreapta de ecuatie  $y+x-4=0$  separa perfect clasele

b) Obtemem   $P' = \{(0,1), (2,1), (4,1), (1,0), (3,0), (5,0)\}$

Multimea  $P'$  nu mai este liniar separabila, nu putem avea un hiperplan (-punkt) care sa separe perfect cele 2 clase

c) Obtemem



$P'' = \{(2,1), (0,1), (-2,1), (5,0), (3,0), (1,0)\}$

Multimea  $P''$  nu este liniar separabila, nu putem avea un hiperplan (-punkt) care sa separe perfect cele 2 clase

**10** întrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Consideram o problema de clasificare binara cu puncte din  $\mathbf{R}$  (dreapta reala). Fie multimea de exemple de antrenare  $P = \{(-1,1), (0,0), (1,0), (2,0), (3,1), (4,1)\}$  unde fiecare exemplu de antrenare e reprezentat de coordonata sa = abscisa si de eticheta corespunzatoare (0 sau 1). Scrieti ponderile  $w_0$  (biasul),  $w_1$  (ponderea pentru abscisa) a unui perceptron cu functia de activare hardlim care obtine acuratetea cea mai mare pe multimea P.

10 intrebare

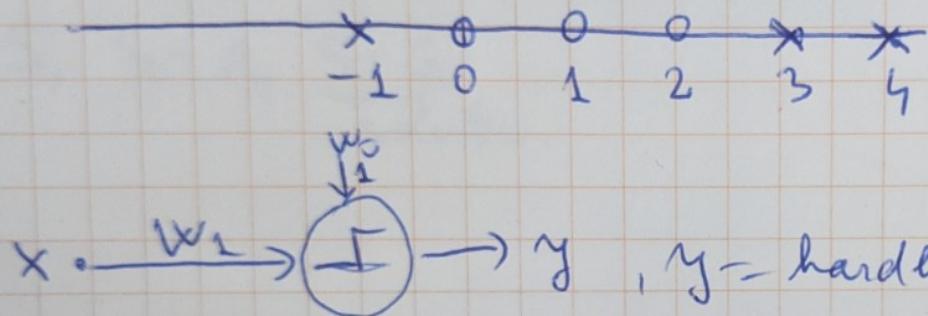
Nu a primit  
răspuns încă

Marcat din 1,00

Întrebare cu  
flag

Consideram o problema de clasificare binara cu puncte din  $\mathbf{R}$  (dreapta reala). Fie multimea de exemple de antrenare  $P = \{(-1,1), (0,0), (1,0), (2,0), (3,1), (4,1)\}$  unde fiecare exemplu de antrenare e reprezentat de coordonata sa = abscisa si de eticheta corespunzatoare (0 sau 1). Scrieti ponderile  $w_0$  (biasul),  $w_1$  (ponderea pentru abscisa) a unui perceptron cu functia de activare hardlim care obtine acuratetea cea mai mare pe multimea P.

7



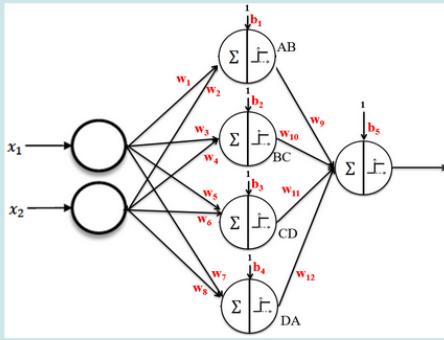
$$y = \text{hardlim}(w_1 \cdot x + w_0) = \begin{cases} 1, & w_1 \cdot x + w_0 \geq 0 \\ 0, & \text{altele} \end{cases}$$

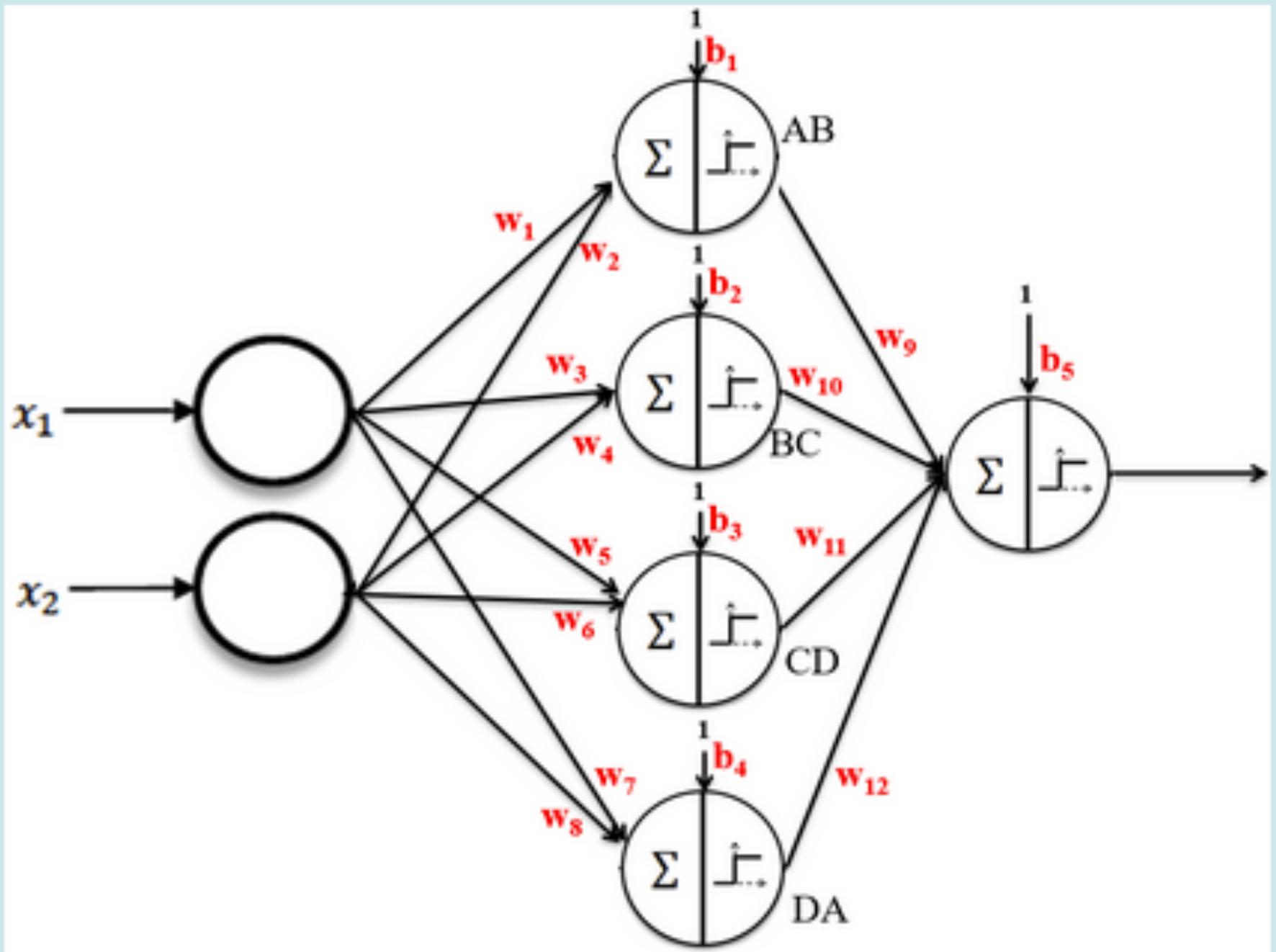
$$w_1 = 1$$

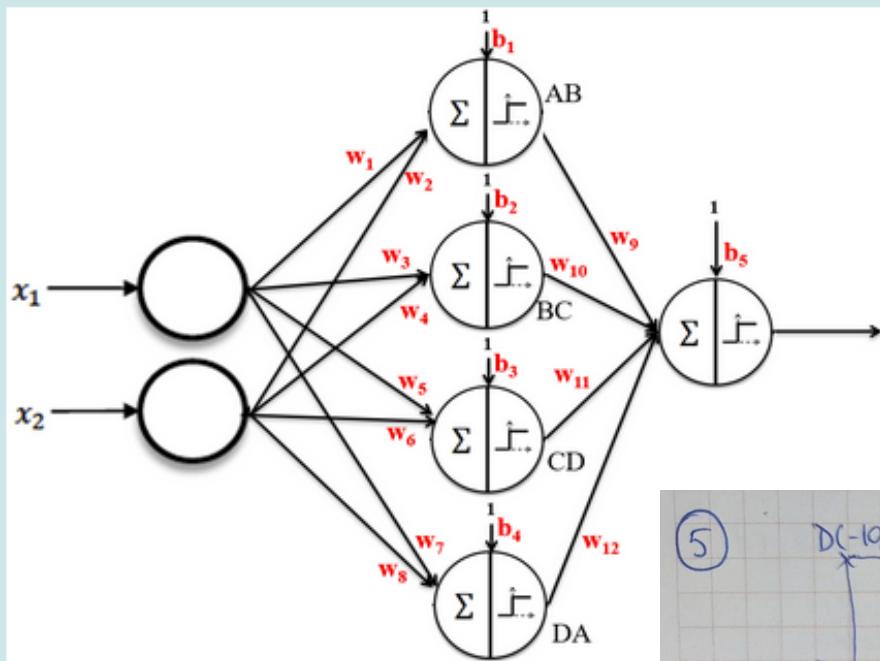
$$w_0 = 2.5$$

$$y = \begin{cases} 1, & x \geq 2.5 \\ 0, & \text{altele} \end{cases}$$

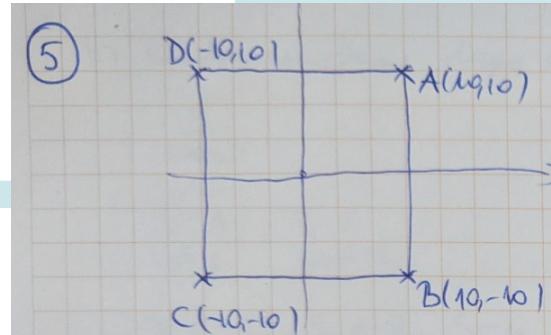
Consideram patratul de varfuri A(10,10), B(10,-10), C(-10,-10), D(-10,10) si reteaua neuronală din figura de mai jos care dorește să implementeze funcția indicator a patratului ABCD: pentru fiecare punct din interiorul sau de pe frontieră patratului rețea trebuie să aibă ieșirea 1, în rest ieșirea rețelei ar trebui să fie 0. Rețea neuronală de perceptri sunt două straturi ascunse. Pe primul strat ascuns sunt patru perceptri care implementează ecuațiile dreptelor AB, BC, CD, DA astfel încât punctele din patrat primesc eticheta 1 din partea tuturor celor 4 perceptri. Al doilea strat ascuns are un singur perceptor care implementează funcția AND de 4 variabile (ieșirile perceptronilor de pe primul strat ascuns). Un punct este în patrat dacă toți cei 4 perceptri de pe primul strat au ieșirea 1. Un punct din exteriorul patratului este caracterizat de faptul că cel puțin un perceptor de pe primul strat ascuns va avea ieșirea 0. Funcția de activare a tuturor perceptronilor din desen este hardlim. Scrieți un set de valori pentru ponderile  $w_1, w_2, \dots, w_{12}$  și bias-urile  $b_1, b_2, \dots, b_5$  astfel încât rețea neuronală obținută să implementeze funcția indicator a patratului ABCD după logica descrisă.







treste a implementa functia indicator a patratului ABCD: pentru fiecare punct din interiorul sau de pe frontieră patratului reteaua ar trebui sa aiba iesirea unui patru perceptroni care implementeaza ecuatia dreptelor AB, BC, CD, DA astfel incat punctele din patrat primesc eticheta 1 din partea tuturor celor patru perceptroniilor de pe primul strat ascuns). Un punct este in patrat daca toti cei 4 perceptroni de pe primul strat au iesirea 1. Un punct din exteriorul patratului de pe primul strat ascuns. Scrieti un set de valori pentru ponderile w<sub>1</sub>, w<sub>2</sub>, ..., w<sub>12</sub> si bias-urile b<sub>1</sub>, b<sub>2</sub>, ..., b<sub>5</sub> astfel incat reteaua



$$AB: x_1=10, \text{ dan } -x_1=-10$$

Perceptual van implementatie heeft AB  
trekking die output- en 1 pt tot pot  
den van huidige particuler. Toch parende le

Strange in 10 thebra se pomeasai 1 ( $x \leq 10$ ). Deicastic wecto est  
 $-x_1 + 10 = 0$  ( $w_1 = -1, w_2 = 0, b_1 = 10$ ).

Somit erhalten wir:  $x_2 + 10 = 0$  ( $w_3 = 0, w_4 = 4, b_2 = 10$ )

$$CD: x_1 + 10 = 0 \quad (w_5 = 1, w_6 = 0, b_3 = 10)$$

$$AD : -x_2 + 10 = 0 \quad (w_2 = 0, v_2 = -1, b_2 = 10)$$

Perceptionul de pe strelul 2 implementeaza AND, deci putem considera

$$w_9 = w_{10} = w_{11} = w_{12} = 1, b_5 = -3.5 \text{ (and } -3.6, -3.7, -3.8\text{, etc.)}$$

**8** intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

Consideram problema regresiei liniare simple pe multimea de antrenare  $S = \{(3,0), (4,2), (6,0), (7,2)\}$ . Notam cu  $w = (w_0, w_1)$  astfel incat  $y = w_0 + w_1 * x$  este dreapta solutie a regresiei liniare. Cat este  $w$ ?

8 intrebare

Nu a primit  
răspuns încă

Marcat din 1,00

▼ Întrebare cu  
flag

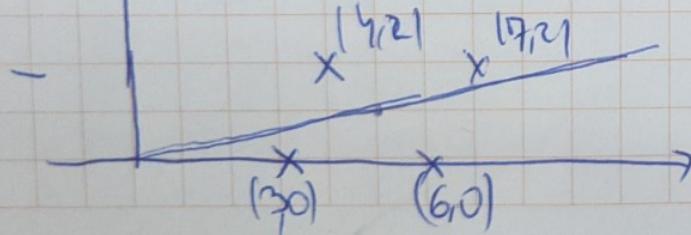
Consideram problema regresiei liniare simple pe multimea de antrenare  $S = \{(3,0), (4,2), (6,0), (7,2)\}$ . Notam cu  $w = (w_0, w_1)$  astfel incat  $y = w_0 + w_1 \cdot x$  este dreapta solutie a regresiei liniare. Cat este  $w$ ?

⑥

$$S = \{(3,0), (4,2), (6,0), (7,2)\} \quad w = (w_0, w_1) \quad \bar{x} = 5, \bar{y} = 1$$
$$w_1 = \frac{\sum (x^i - \bar{x})(y^i - \bar{y})}{\sum (x^i - \bar{x})^2} = \frac{(3-5)(0-1) + \dots + (7-5)(2-1)}{(3-5)^2 + \dots + (7-5)^2}$$

$$= \frac{2}{10} = \frac{1}{5}$$

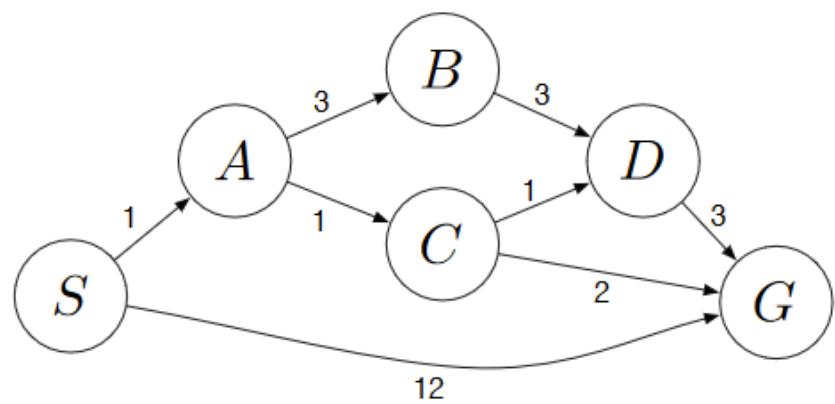
$$w_0 = \frac{1}{m} [\sum y^i - w_1 \sum x^i] = \bar{y} - w_1 \bar{x} =$$
$$= 1 - \frac{1}{5} \cdot 5 = 0$$



Consideram problema de cautare data de graful din figura de mai jos. S este starea initiala, G este starea scop. O strategie de cautare care porneste din starea S si vrea sa ajunga la starea G exploreaza starile intr-o anumita ordine. Implicit, se prefera explorarea starilor in ordine lexicografica. Realizati urmatoarele:

1. Care este solutia problemei de cautare folosind strategia de cautare in latime? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.
2. Care este solutia problemei de cautare folosind strategia de cautare uniforma dupa cost? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.

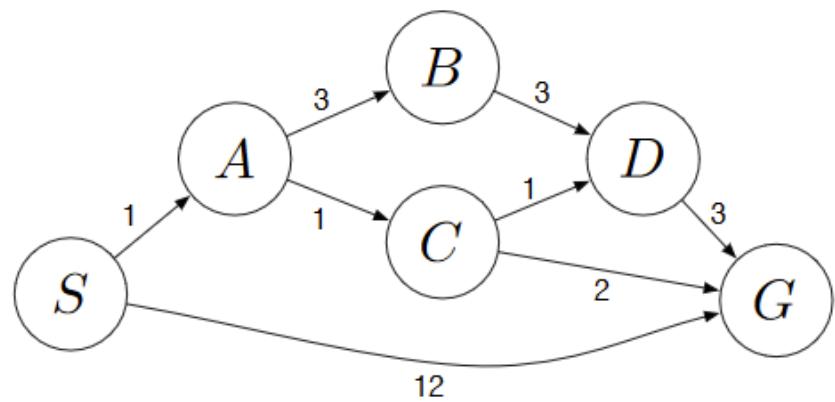
(Cand raspundeti la intrebari folositi ca notatie pentru o solutie parciala urmatoare notatie: S-A-D-G - solutie parciala invalida pentru cazul de fata)



Consideram problema de cautare data de graful din figura de mai jos. S este starea initiala, G este starea scop. O strategie de cautare care porneste din starea S si vrea sa ajunga la starea G exploreaza starile intr-o anumita ordine. Implicit, se prefera explorarea starilor in ordine lexicografica. Realizati urmatoarele:

1. Care este solutia problemei de cautare folosind strategia de cautare in latime? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.
2. Care este solutia problemei de cautare folosind strategia de cautare uniforma dupa cost? Specificati solutiile partiale construite in frontiera pana la obtinerea solutiei finale.

(Cand raspundeti la intrebari folositi ca notatie pentru o solutie parciala urmatoare notatie: S-A-D-G - solutie parciala invalida pentru cazul de fata)



⑦ a) Cautare in latime:

S,

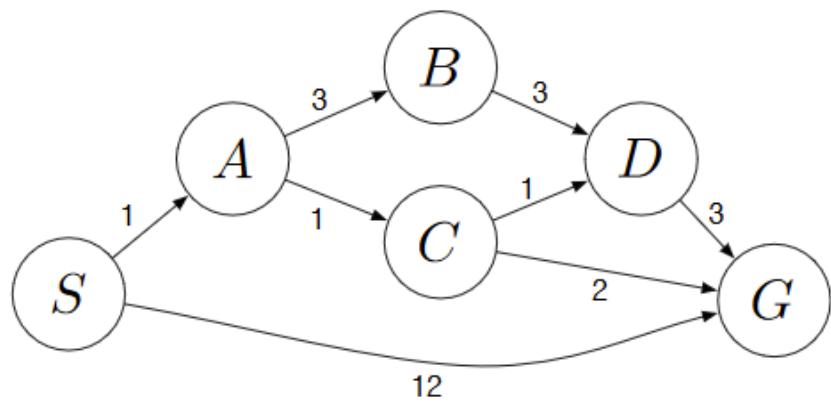
$S \rightarrow S-G$

solutie

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează starile într-o anumita ordine. Implicit, se preferă explorarea starilor în ordine lexicografică. Realizați urmatoarele:

1. Care este soluția problemei de căutare folosind strategia de căutare în latime? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.
2. Care este soluția problemei de căutare folosind strategia de căutare uniformă după cost? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție parțială invalidă pentru cazul de față)



⑦ a) Căutare în latime:

$$\begin{array}{l} S, \\ S-A, \frac{S-G}{\text{solutie}} \end{array}$$

b) căutare uniformă după cost

$$\begin{array}{l} S \\ \text{cost} = 0 \end{array}$$

$$\begin{array}{l} S-A \\ \text{cost } 1 \end{array}, \begin{array}{l} S-G \\ \text{cost } 12 \end{array}$$

$$\begin{array}{l} S-A-C \\ \text{cost } 2 \end{array}, \begin{array}{l} S-A-B \\ \text{cost } 1 \end{array}, \begin{array}{l} S-G \\ \text{cost } 12 \end{array}$$

$$\begin{array}{l} S-A-C-D \\ \text{cost } 3 \end{array}, \begin{array}{l} S-A-C-G \\ \text{cost } 4 \end{array}, \begin{array}{l} S-A-B \\ \text{cost } 1 \end{array}, \begin{array}{l} S-G \\ \text{cost } 12 \end{array}$$

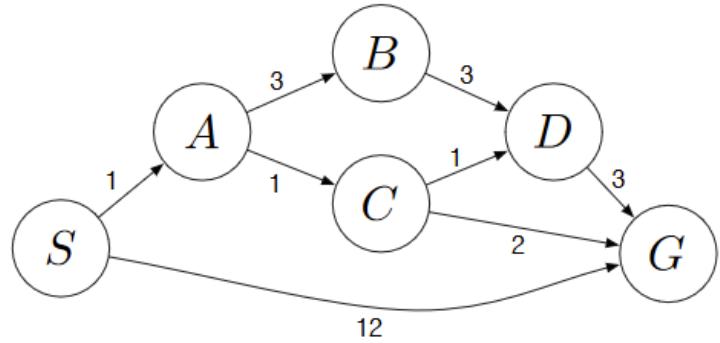
$$\begin{array}{l} S-A-C-D-G \\ \text{cost } 6 \end{array}, \begin{array}{l} S-A-C-G \\ \text{cost } 4 \end{array}, \begin{array}{l} S-A-B \\ \text{cost } 1 \end{array}, \begin{array}{l} S-G \\ \text{cost } 12 \end{array}$$

$$\begin{array}{l} S-A-C-D-G, \begin{array}{l} S-A-C-G \\ \text{cost } 4 \end{array}, \begin{array}{l} S-A-B-D \\ \text{cost } 7 \end{array}, \begin{array}{l} S-G \\ \text{cost } 12 \end{array} \\ \text{solutie} \end{array}$$

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează stările într-o anumita ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (la costuri egale). Considerăm euristicile  $h_1$  și  $h_2$  din tabelul de mai jos. Realizați următoarele:

1. Care din euristicile  $h_1$  și  $h_2$  este admisibilă? Justificați răspunsul.
2. Care este soluția problemei de căutare folosind algoritmul A\* cu heuristică  $h_1$ ? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție invalidă pentru cazul de față)

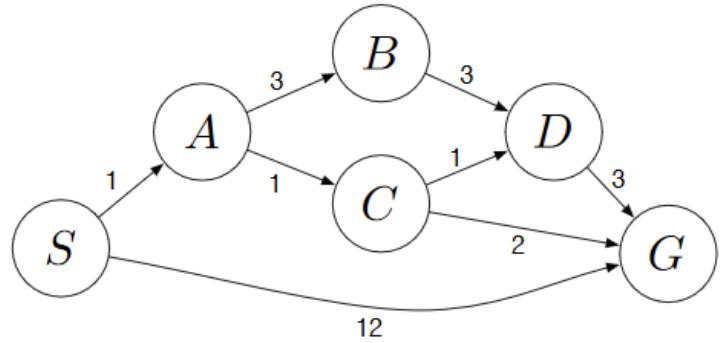


State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează stările într-o anumita ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (la costuri egale). Considerăm euristicile  $h_1$  și  $h_2$  din tabelul de mai jos. Realizați următoarele:

1. Care din euristicile  $h_1$  și  $h_2$  este admisibilă? Justificați răspunsul.
2. Care este soluția problemei de căutare folosind algoritmul A\* cu heuristică  $h_1$ ? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție invalidă pentru cazul de față)



State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

⑧ a)  $h$  este admisibilă dacă substanțnează costul optim al găsirii mod.

$$h^*(S) = 4$$

$$h_1(S) = 5 \Rightarrow h_1 \text{ nu este admisibilă}$$

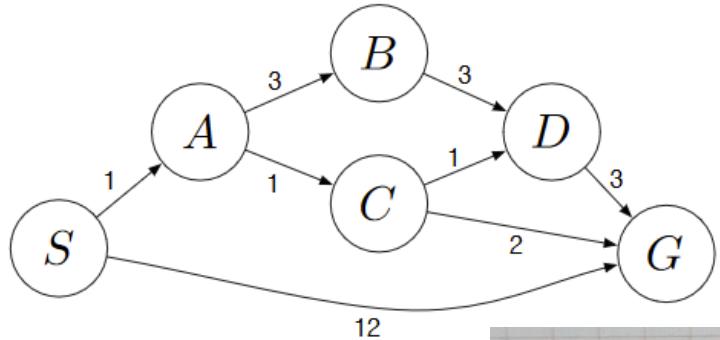
$h_2$  verifică condiția  $h_2(N) \leq h^*(N)$  pentru orice nod N

$\Rightarrow h_2$  este admisibila

Consideram problema de cautare data de graful din figura de mai jos. S este starea initială, G este starea scop. O strategie de căutare care porneste din starea S și vrea să ajunga la starea G explorează stările într-o anumita ordine. Implicit, se preferă explorarea stărilor în ordine lexicografică (la costuri egale). Considerăm euristicile  $h_1$  și  $h_2$  din tabelul de mai jos. Realizați următoarele:

1. Care din euristicile  $h_1$  și  $h_2$  este admisibilă? Justificați răspunsul.
2. Care este soluția problemei de căutare folosind algoritmul A\* cu heuristică  $h_1$ ? Specificați soluțiile parțiale construite în frontieră până la obținerea soluției finale.

(Când răspundeti la întrebări folosiți ca notație pentru o soluție parțială următoare notație: S-A-D-G - soluție invalidă pentru cazul de față)



State	$h_1$	$h_2$
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

b)  $f(n) = g(n) + h(n)$   
 tractat (exact) vechi (șters)

$S(5)$   
 $g=9, h=5$

$S-A(4), S-G(12)$   
 $g=1, h=3 \quad g=12, h=0$

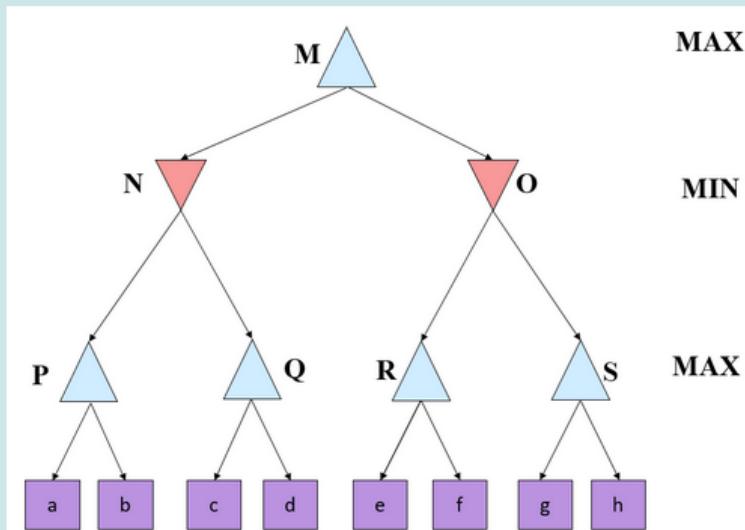
$S-A-C(4), S-A-B(10), S-G(12)$   
 $g=2, h=2 \quad g=4, h=6 \quad g=12, h=0$

$S-A-C-D(6), S-A-C-G(4), S-A-B(10), S-C(12)$   
 $g=3, h=3 \quad g=4, h=\infty \quad g=4, h=6 \quad g=12, h=0$

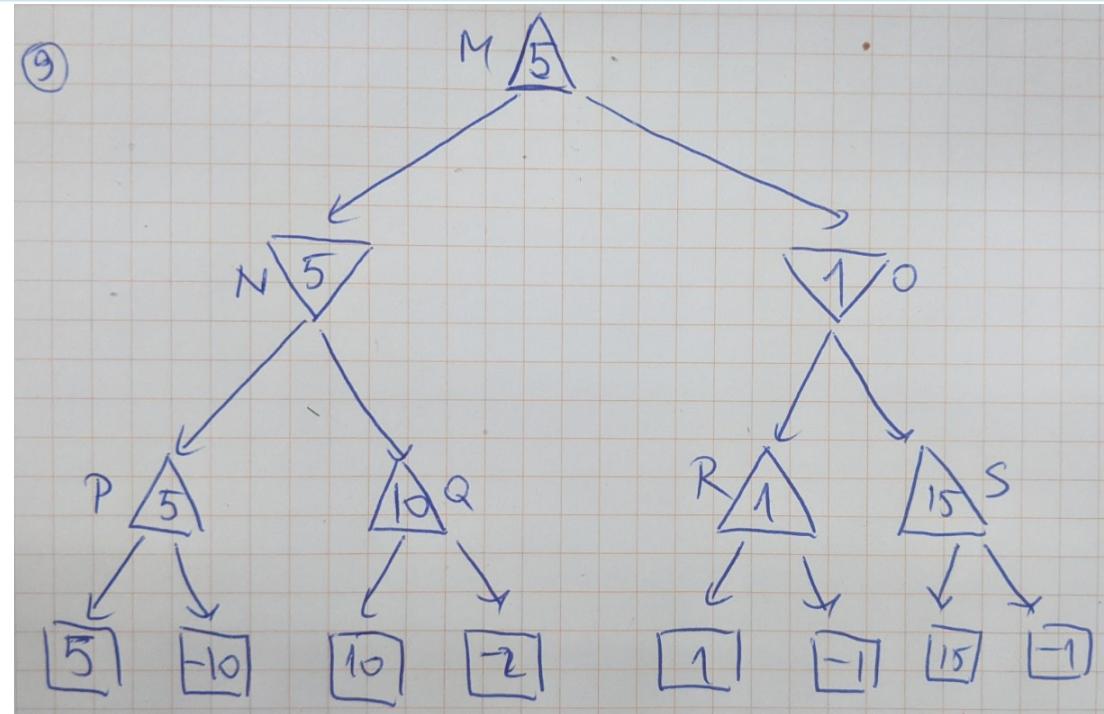
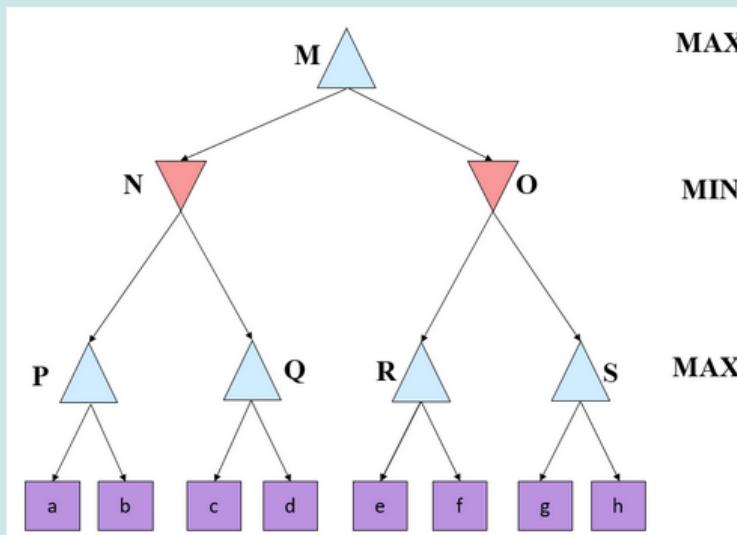
soluție

Stare	$h_1$
S	5
A	3
B	6
C	2
D	3
G	0

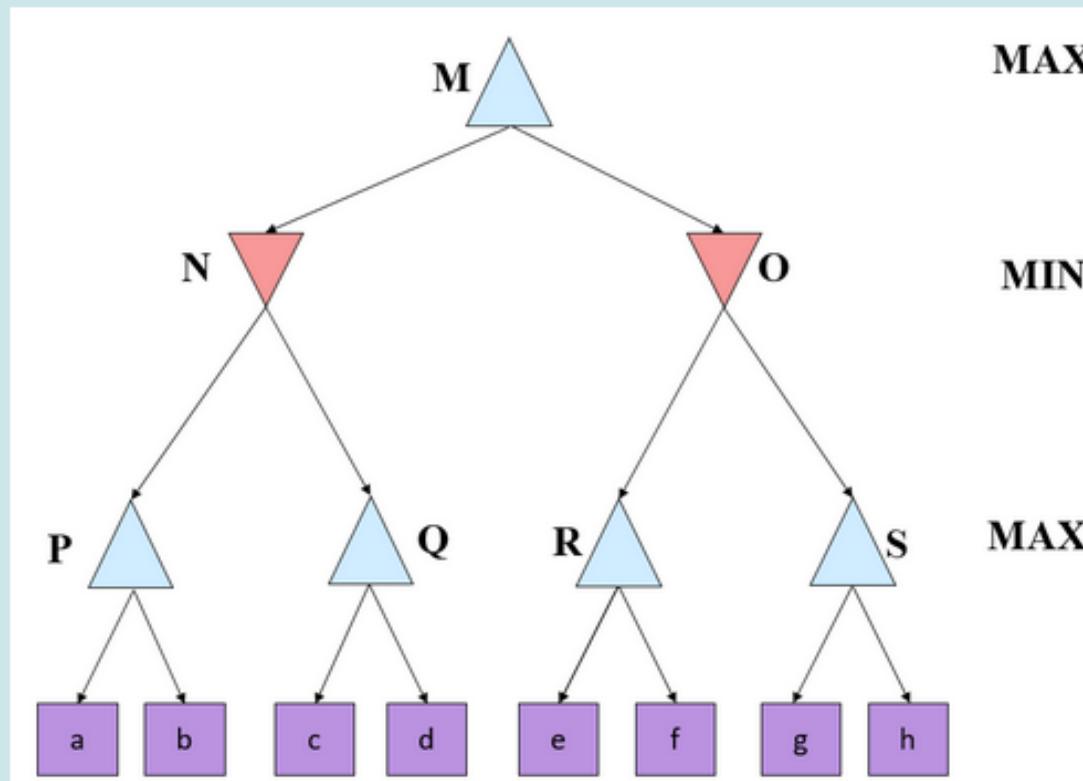
Consideram arborele de cautare din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN vrea sa minimizeze castigul lui MAX. Nodurile terminale au valorile a = 5, b = -10, c = 10, d = -2, e = 1, f = -1, g = 15, h = -1. Care sunt valorile Minimax ale nodurilor M, N, O, P, Q, R, S?



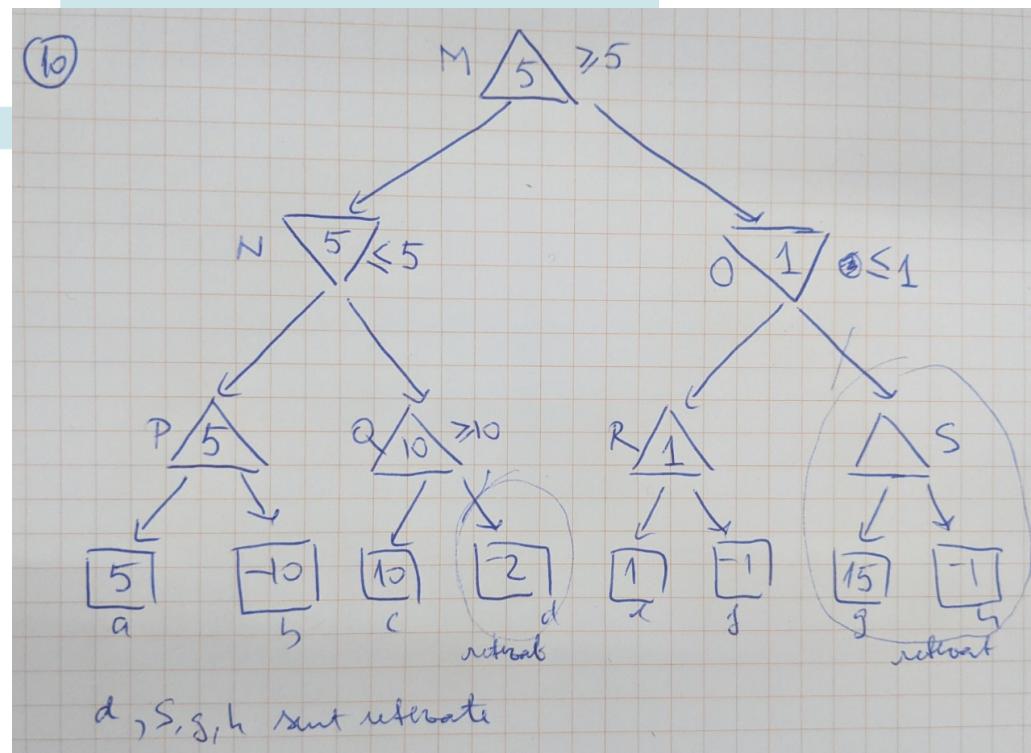
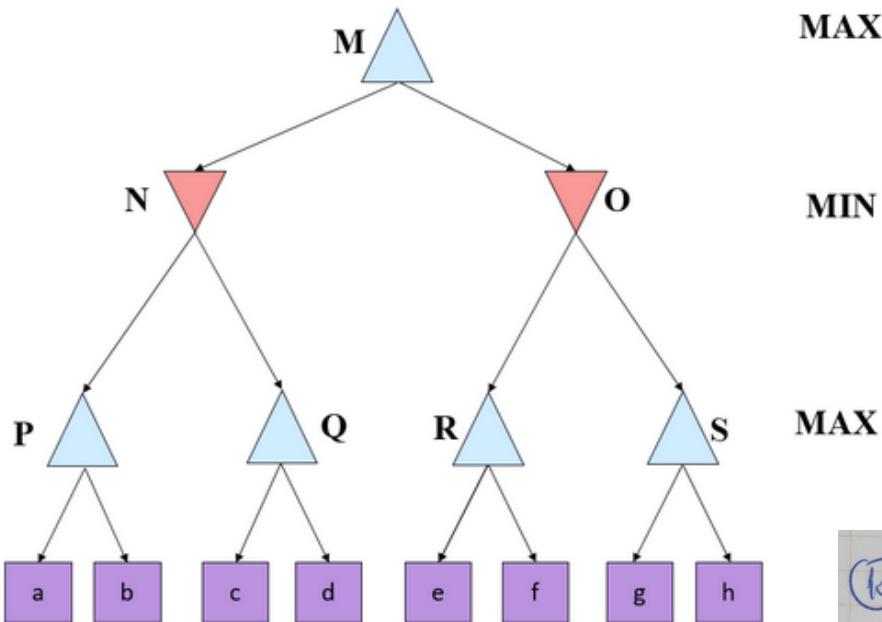
Consideram arborele de cautare din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN vrea sa minimizeze castigul lui MAX. Nodurile terminale au valorile a = 5, b = -10, c = 10, d = -2, e = 1, f = -1, g = 15, h = -1. Care sunt valorile Minimax ale nodurilor M, N, O, P, Q, R, S?



Consideram arborele de joc din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN aplică algoritmul alfa-beta în calcul valorii Minimax a radacinii M. Ce noduri sunt retezate (nu sunt vizitate)? Justificati raspunsul.



Consideram arborele de joc din figura de mai jos, in care jucatorul la mutare este jucatorul MAX (vrea sa isi maximizeze castigul). Jucatorul MIN va calcula valoarea minimă a radacinii M. Ce noduri sunt retezate (nu sunt vizitate)? Justificati raspunsul.



# Cursuri optionale CTI anul 4

CTI, anul 4, semestrul 1		CTI, anul 4, semestrul 2
<b>DD1</b>	Metode moderne de calcul si simulare	<b>DD1</b> Sistemul de operare Linux
<b>DD2</b>	Prelucrarea semnalelor	<b>DD2</b> Algoritmi paraleli
<b>DC1</b>	Comunicare si relatii publice	<b>DC1</b> Managementul operatiunilor
<b>DC2</b>	Management	<b>DC2</b> Marketing
<b>DS1</b>	Cloud Computing	<b>DS1</b> Criptografie si securitate
<b>DS2</b>	Tehnici Web	<b>DS2</b> Programarea pe dispozitive mobile
<b>DS3</b>	Scalarea retelelor de calculatoare	<b>DS3</b> Conectarea retelelor la nivel WAN
<b>DS4</b>	Calcul Numeric în Informatică	<b>DS4</b> Blockchain
<b>DS5</b>	Concepțe și aplicații în Vederea Artificială	<b>DS5</b> Elemente de securitate și logica aplicată
<b>DS6</b>	Introducere în Robotica	<b>DS6</b> Introducere în prelucrarea limbajului natural
<b>DS7</b>	Tehnici de optimizare	<b>DS7</b> Învățare automată în Vedere Artificială
<b>DS8</b>	Robotic Process Automation (RPA) folosind platforma UiPath	<b>DS8</b> Metode Formale în Ingineria Software
		<b>DS9</b> Production engineering
		<b>DS10</b> Strategii de planificare a unei echipe de roboti (SPER)
		<b>DS11</b> Inițiere în cercetare și bioinformatică
		<b>DS12</b> Algoritmica avansata a grafurilor

# Cursuri optionale CTI anul 4

CTI, anul 4, semestrul 1		CTI, anul 4, semestrul 2	
DD1	Metode moderne de calcul si simulare	DD1	Sistemul de operare Linux
DD2	Prelucrarea semnalelor	DD2	Algoritmi paraleli
DC1	Comunicare si relatii publice	DC1	Managementul operatiunilor
DC2	Management	DC2	Marketing
DS1	Cloud Computing	DS1	Criptografie si securitate
DS2	Tehnici Web	DS2	Programarea pe dispozitive mobile
DS3	Scalarea retelelor de calculatoare	DS3	Conectarea retelelor la nivel WAN
DS4	Cloud Computing I	DS4	Blockchain
DS5	Concepțe și aplicații în Vederea Artificială	DS5	Elemente de securitate și logica aplicată
DS6	Introducere în Robotica	DS6	Introducere în prelucrarea limbajului natural
DS7	Tehnici de optimizare	DS7	Învățare automată în Vedere Artificială
DS8	Robotic Process Automation (RPA) folosind platforma UiPath	DS8	Metode Formale în Ingineria Software
		DS9	Production engineering
		DS10	Strategii de planificare a unei echipe de roboti (SPER)
		DS11	Inițiere în cercetare și bioinformatică
		DS12	Algoritmica avansata a grafurilor

# Prezentare curs optional: Concepte și Aplicații în Vedere Artificială



**Bogdan**

Curs (50%)



**Radu**

Curs (50%)



**Alexandra**

Laborator (100%)

-cursul optional se adresează studenților de anul 3 Mate-Info, anul 3 Info, anul 4 CTI

# Ce este vederea artificială?



- Înzestrarea computerelor cu un sistem vizual asemănător cu sistemul vizual uman
- Scrierea de programe pentru calculator care pot interpreta imagini/video-uri

# Structura cursului

## 0. Formarea imaginilor

- generalități despre lucrul cu imagini

Laborator: realizarea de mozaicuri

Imagine  
de  
referință



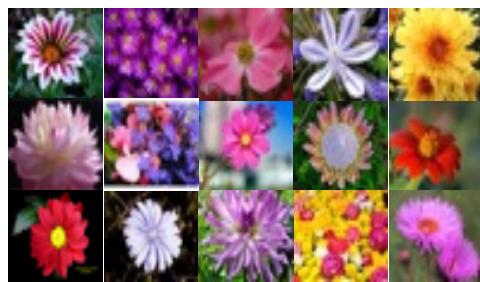
colecție de  
imagini (piese)  
de dimensiuni  
reduse

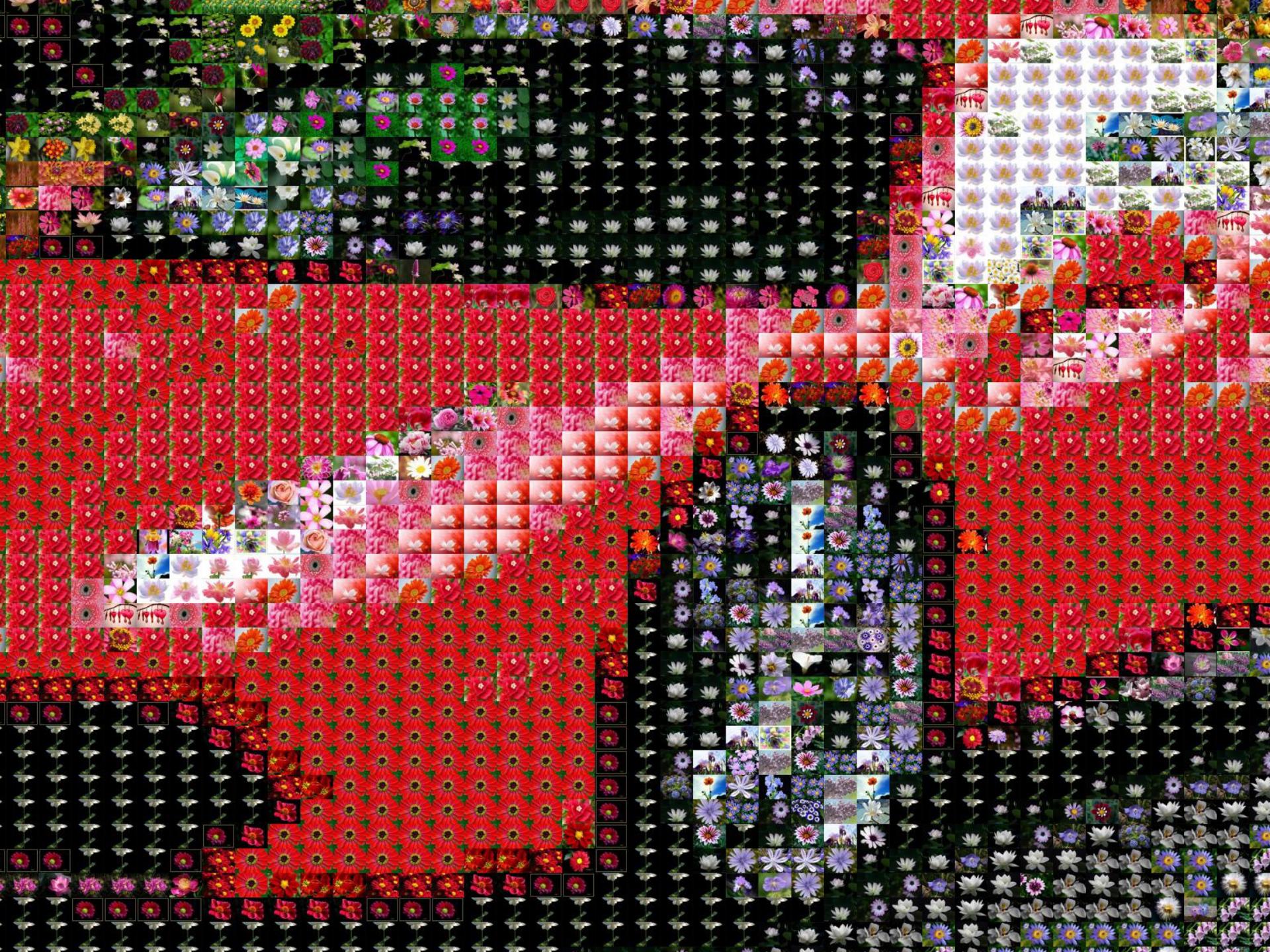


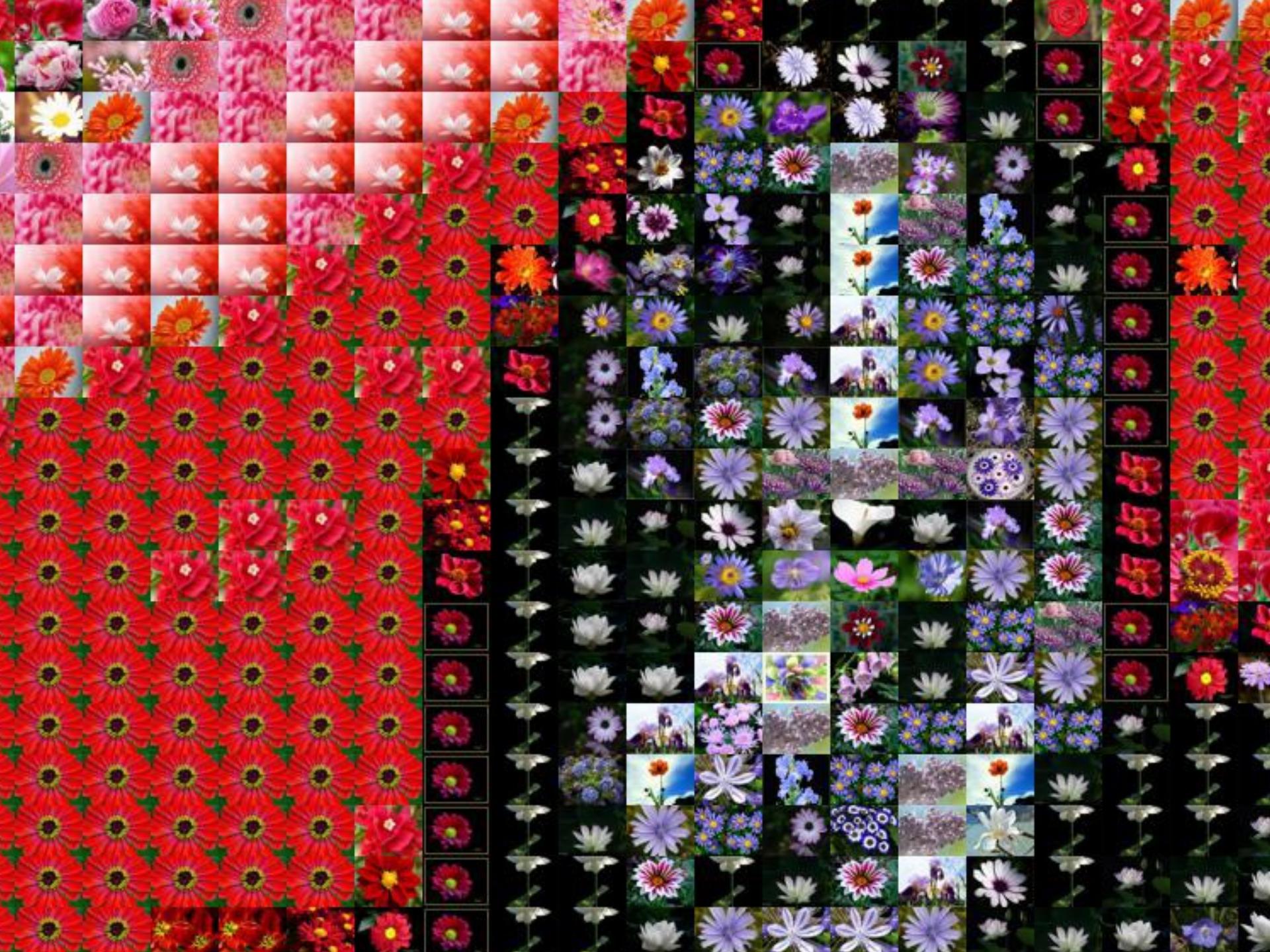
algoritm  
codat de  
voi



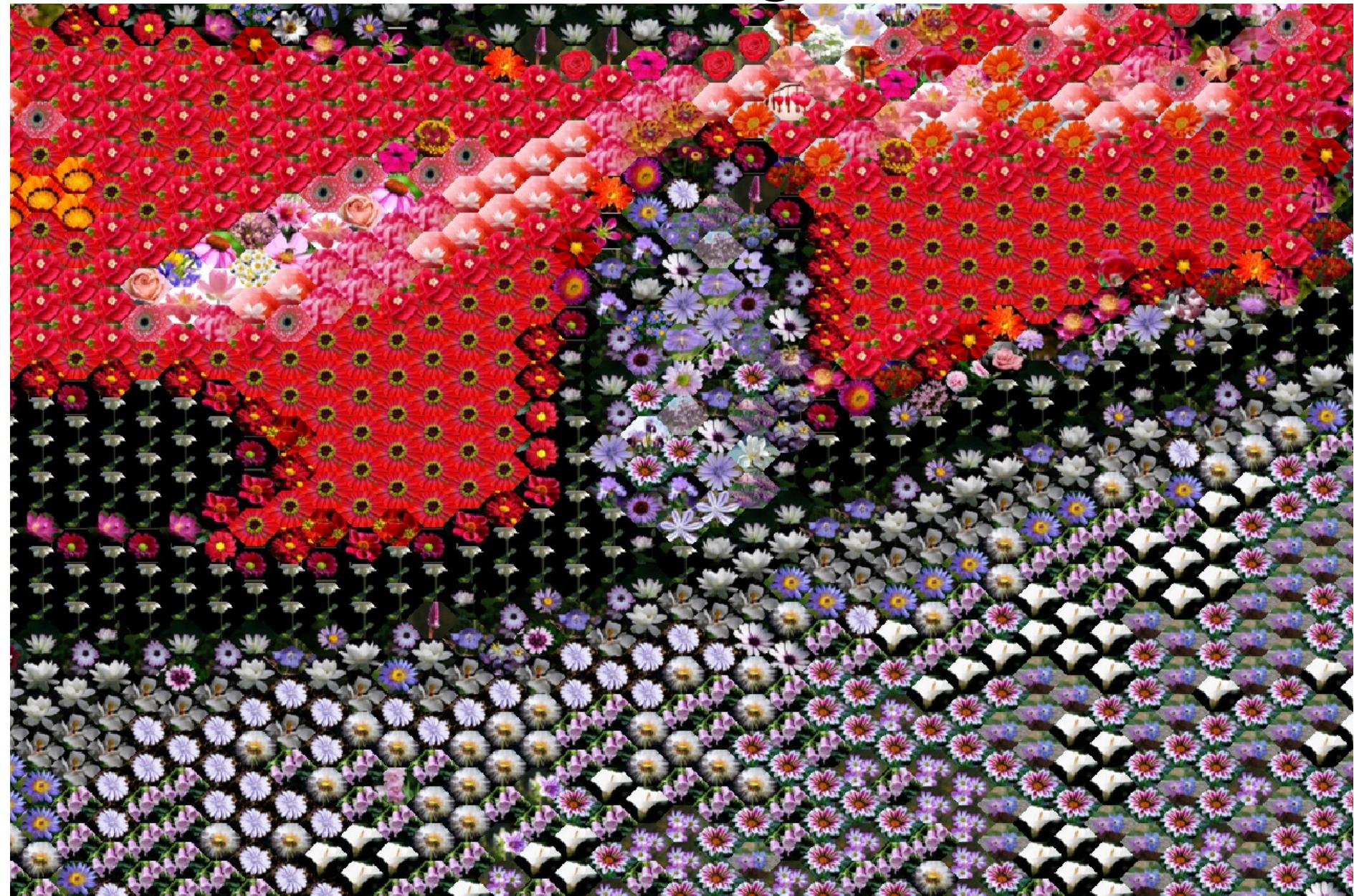
Imagine  
mozaic







# Piese hexagonale



# Piese hexagonale



# Structura cursului

## 1. Caracteristici ale imaginilor

- filtre, gradienti, muchii, textură

Laborator: redimensionarea imaginilor cu păstrarea conținutului



Redimensionare  
uzuală (imresize)



Redimensionare cu  
păstrarea conținutului

# Structura cursului optional

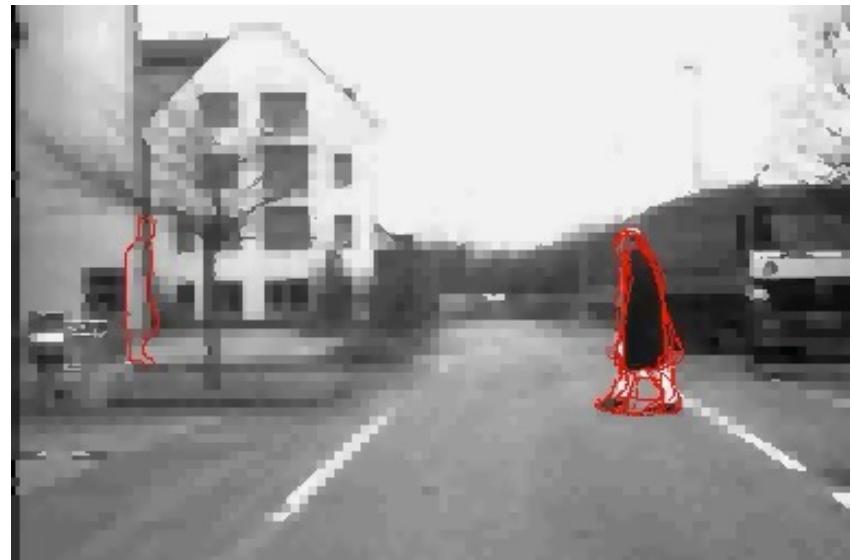
## 1. Caracteristici ale imaginilor

- filtre, gradienti, muchii, textură

Laborator: detectare de obiecte



Detectare semne de circulație



Detectare pietoni

# Structura cursului optional

## 2. Descriptori vizuali

- puncte de interes, descriptori SIFT, descriptori HOG

Laborator: detectare facială



# Structura cursului optional

## 3. Recunoaștere de obiecte

- modelul K-nearest neighbours, metode kernel, rețele neuronale conveționale, modelul bag of visual words

Laborator: clasificarea imaginilor, colorarea imaginilor în tonuri de gri (grayscale)

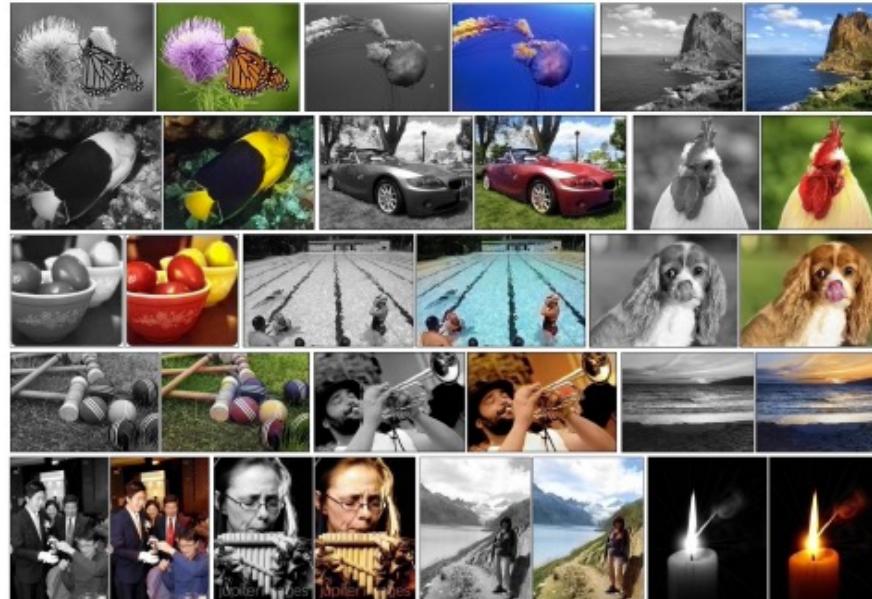


Figura 1: Rezultatele colorării imaginilor în tonuri de gri.

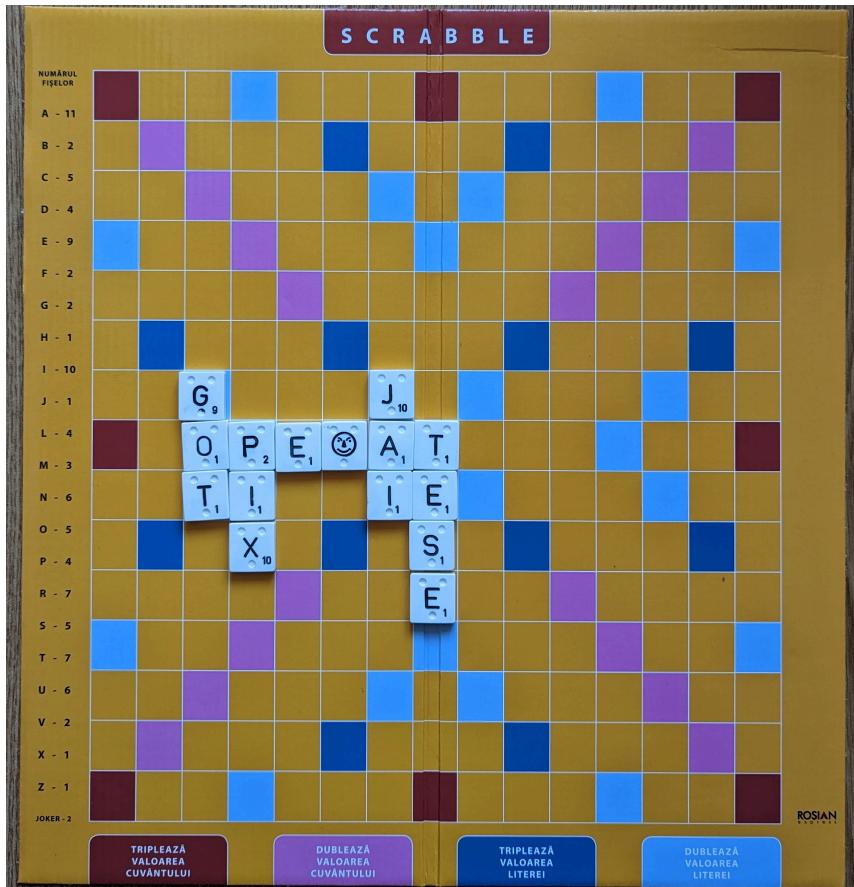
# Tema 1 din acest an:

# Calculator automat de scor

Scrabble este un joc în care participanții formează cuvinte prin plasarea de litere pe o tablă având dimensiunile de  $15 \times 15$  pătrate (Figura 1). Cuvintele se pot forma pe orizontală sau pe verticală, precum la cuvinte încrucișate, iar punctajul obținut este mai mare atunci când literele folosite sunt mai rare (mai valoroase) sau când sunt plasate pe pătrate divers colorate care dau diferite bonificări ale tablei de joc. Cuvintele sunt valide doar dacă corespund dicționarelor acceptate oficial. Fiecare jucător concurează pentru a obține un scor cât mai mare folosind diferite combinații de litere plasându-le cât mai bine pe tablă astfel încât să profite de pătratele premium.



# Exemplu de formare a cuvintelor + calcul scor



COOPERAT:  $(1(C) + 1(O) + 1(O) + 2(P) + 1(E) + 0(?) + 1(A) + 1(T)) * 3(STC) = 24$

**TOTAL: 24**

SDL = SCOR DUBLU LITERA  
STL = SCOR TRIPLU LITERA

SDC = SCOR DUBLU CUVANT  
STC = SCOR TRIPLU CUVANT

# Tema 2 din acest an:

## Detectarea și recunoașterea facială a personajelor din serialul de desene animate **Viața cu Louie**

*Viața cu Louie* este un serial de desene animate lansat în anul 1994 și care are 3 sezoane, a căte 13 episoade. Serialul prezintă întâmplări din viața lui Louie Anderson, un copil de 8 ani care trăiește în Cedar Knoll, Wisconsin (SUA). Multe episoade sunt inspirate din viața adevăratului Louie Anderson, comedian american care apare la începutul episoadelor și care le vorbește telespectatorilor, acesta regizând, de altfel, și toate episoadele.



# Task 1 – detectare facială

Prima problemă pe care o aveți de rezolvat constă în detectarea facială a *tuturor* fețelor personajelor care apar în imagini. Pentru fiecare imagine de intrare algoritmul vostru trebuie să returneze o mulțime de detecții asociate (fereastră dreptunghiulară și scor) ce localizează *toate* fețele dintr-o imagine. Figura 1 arată câteva exemple din mulțimea de antrenare și adnotările corespunzătoare, ce constau în ferestre dreptunghiulare de culoare roșie ce încadrează perfect *fiecare față*.

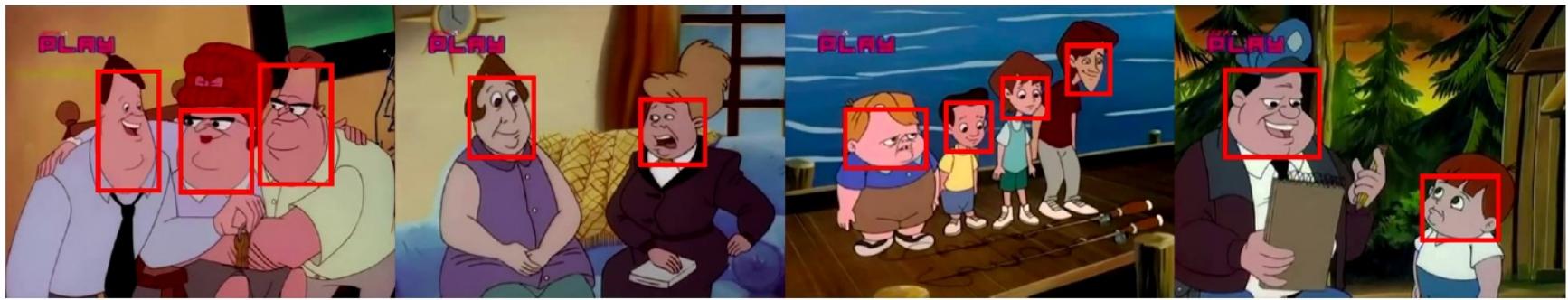
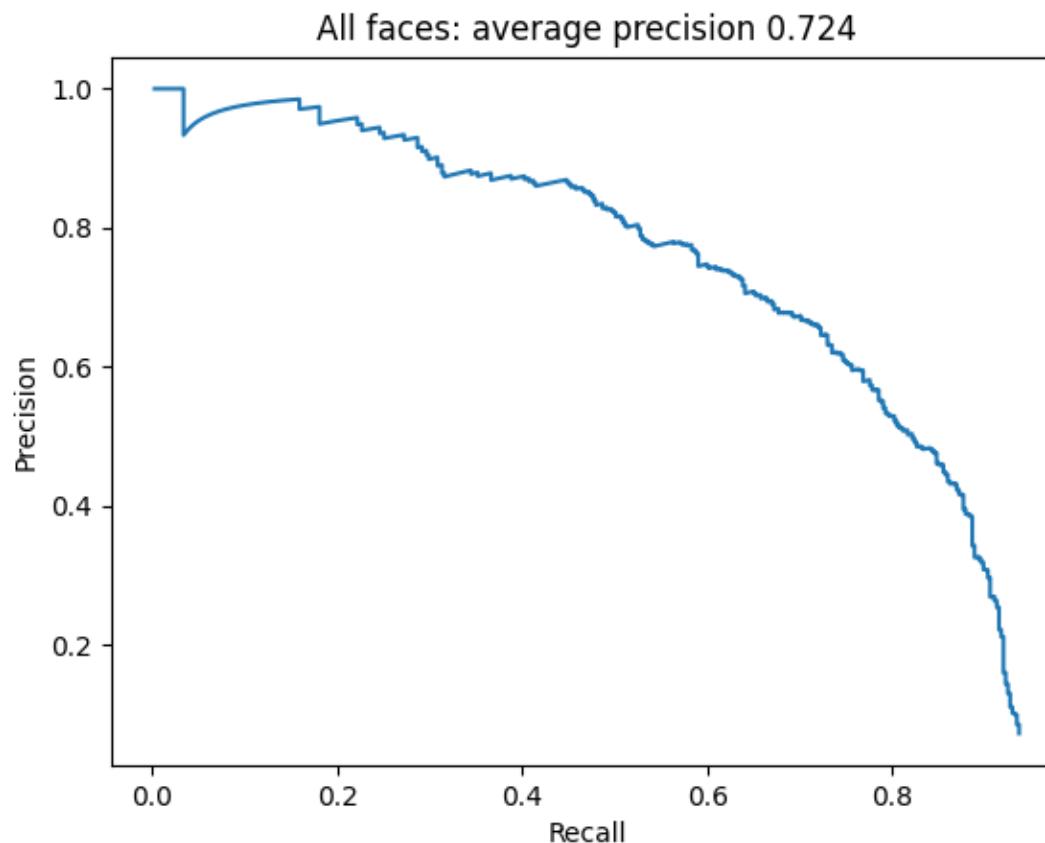


Figura 1: Detectare facială a personajelor din Viața cu Louie: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare roșie.

# Evaluare task 1 – AP pe datele de test



(performanța soluției noastre pe datele de validare = 200 de imagini)

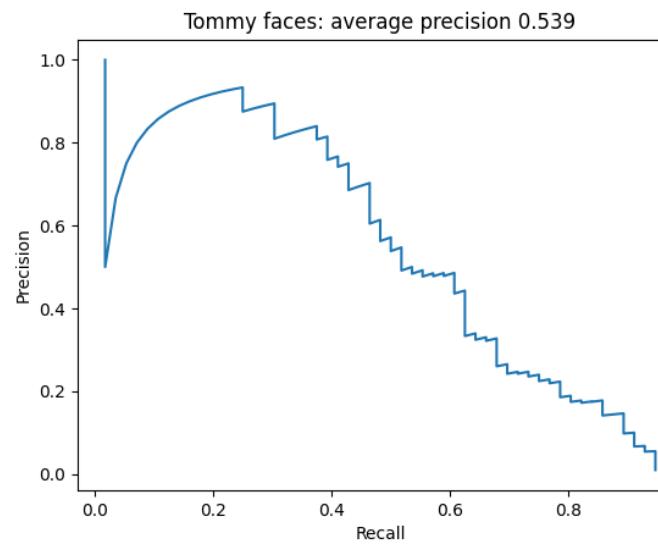
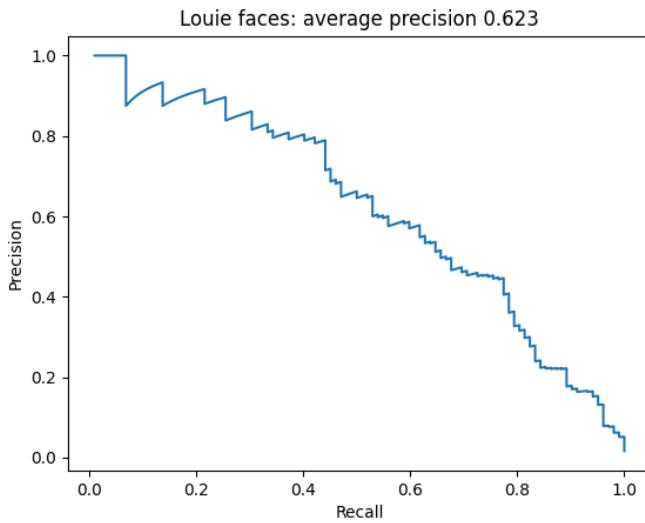
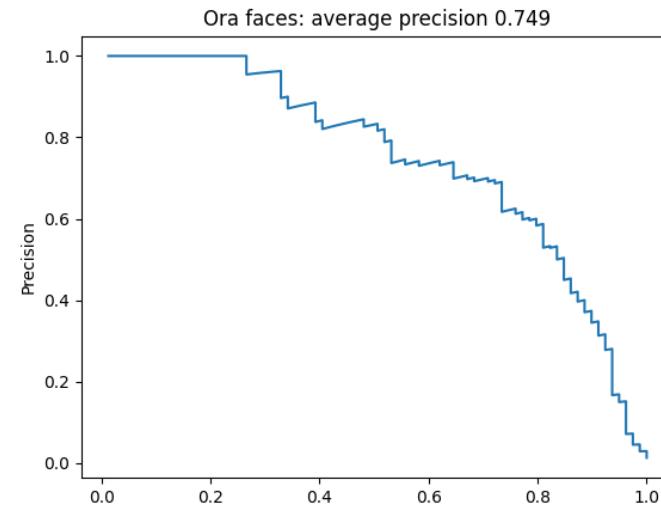
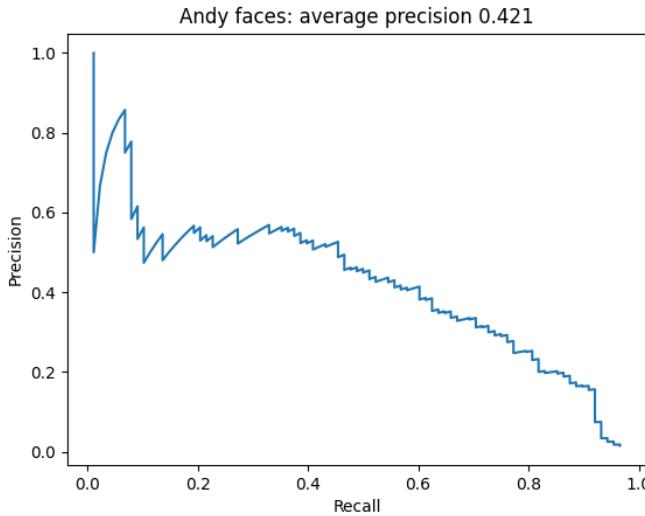
# Task 2 – recunoaștere facială

A doua problemă pe care o aveți de rezolvat constă în recunoașterea facială a numai anumitor personaje. Alături de Louie, personajul principal din serial, apar cu preponderență alte trei personaje: Andy - tatăl său, Ora - mama sa, Tommy - fratele său. Vom considera problema recunoașterii faciale numai pentru aceste patru personaje, Andy, Ora, Louie și Tommy. Pentru fiecare imagine de intrare algoritmul vostru trebuie să returneze o mulțime de detecții asociate (numele personajului, fereastră dreptunghiulară și scor) ce localizează fețele celor patru personaje de interes (Andy, Ora, Louie, Tommy) din imagine. Figura 2 arată câteva exemple din mulțimea de antrenare și adnotările corespunzătoare, ce constau în ferestre dreptunghiulare ce încadrează perfect *fețele de interes* (pentru Andy, Ora, Louie și Tommy). Fiecare detecție are o culoare specifică clasei personajului (albastru - Andy, galben - Ora, verde - Louie, violet - Tommy).



Figura 2: Recunoașterea facială a personajelor din *Viața cu Louie*: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare specifică clasei personajului (albastru - Andy, galben - Ora, verde - Louie, violet - Tommy).

# Evaluare task 2 – mAP pe datele de test



(performanța soluției noastre pe datele de validare = 200 de imagini)