

Acest document se adreseaza studentilor de la **INFO grupele 231-234**, an universitar 2019-2020.

In urma votului studentilor, majoritatea au ales varianta de evaluare finala bazata pe **proiect** (2 pb., 10-14 zile).

Nota de laborator IA "Reprezentarea Cunostintelor" contine:

- **2p** activitatea din timpul celor 7 saptamani (temele rezolvate pt problemele obligatorii, activitatea de la ore, participarea la discutii, etc)
- **4p** proiect_1 (o problema la alegere cu Alg. A-star, dintre enunturile din pdf-ul "LAB_6" de pe Google Drive)
- **4p** (+1p bonus) proiect_2 (o problema la alegere cu Alg. Minimax si Alg. Alpha-Beta, dintre enunturile din pdf-ul "LAB_9" de pe Google Drive).

Punctul bonus se acorda pentru adaugarea unei interfete, folosind Pygame, pentru a putea alege mutarile jocului cu click de mouse in loc de introducerea lor de la tastatura.

Termenul de trimitere a proiectelor este 26 aprilie 2020, seara, la linkul:

(<https://www.dropbox.com/request/X9Zs83PRTp37Pp6yRnm4>)

Toate fisierele unui student (atat cele 2 cu codul, cat si fisierele de input si output) trebuie sa fie denumite incepand cu "**Grupa_Nume_Prenume_LabNr_PbNr**" apoi continuand cu "**cod.py**" sau "**input1.txt**" etc., unde:

- o Grupa in {231, 232, 233, 234}
- o LabNr in {Lab6, Lab9} pentru proiect_1, respectiv proiect_2
- o PbNr in {Pb4, Pb5, Pb6, ...} conform numarului problemei alese din pdf-ul cu enunturi.

Pe Dropbox, dupa ce selectati fisierele si le adaugati (separate, NU zip):

- La "first name" completati de fapt **grupa** (pt ca vor fi sortate dupa acel camp)
- La "last name" puneti **numele si prenumele**

Tema nu se puncteaza fara prezentare. Se va da o nota pe prezentare de la 1 la 10 in functie de cat de bine a stiut studentul sa explice ce a facut. Punctajul temei se va inmulti cu nota_prezentare/10. Astfel, daca cineva stie sa explice doar jumatate din ce a facut, primeste jumatate din punctaj; daca nu stie nimic primeste 0, etc.

Temele copiate duc la anularea notei atat pentru cel care a dat tema cat si pentru cel care a copiat, iar numele studentilor cu aceasta problema vor fi comunicate profesorului titular de curs.

Aveti mai jos **barem**ul detaliat pentru fiecare problema (baremul este acelasi, indiferent de enuntul ales din lista).

Barem proiect_1: (punctajul e dat in procentaje din punctajul maxim al temei; procentajul maxim este 100%):

- **(10%)** Numele tuturor fisierele de input vor fi scrise intr-o lista, in program, realizand rularile pe baza acelei liste (pentru fiecare fisier „input_X.txt” sa se obtina cate un fisier „output_X.txt”, cu X in {1,2,3,4}). Citirea din fisier + memorarea starii initiale a problemei.
- **(20%)** Functia de generare a succesorilor.
- **(10%)** Functia de testare a scopului.
- **(25% =10+10+5)** 3 euristici. Gasirea a doua euristici admisibile posibile (se va justifica la prezentare si in comentarii daca sunt si consistente), diferite de euristica banala (1 pentru nod scop, 0 in rest); si gasirea unei euristici care nu indeplineste conditia de admisibilitate.

[**Obs:** Euristica e admisibila daca $fct \hat{h}$ nu supraestimeaza niciodata valoarea efectiva h . Aveti in curs detalii despre conditia de consistenta. Pe scurt, trebuie verificat daca $\hat{h}(tata) \leq cost_muchie(tata, fiu) + \hat{h}(fiu)$, adica daca \hat{f} descreste monoton pe masura ce ne indepartam de nodul de start.]

- **(10%)** Crearea a 4 fisiere de input cu urmatoarele proprietati:
 - o un fisier de input care nu are solutii,
 - o un fisier de input care da o stare initiala care este si finala
 - o un fisier de input cu un drum de cost minim de lungime 3-5
 - o un fisier de input cu un drum de cost minim de lungime mai mare decat 5
 - o dintre ultimele doua fisiere, cel putin un fisier sa dea drumul de cost minim pentru euristicile admisibile si un drum care nu e de cost minim pentru acea euristica neadmisibila.
- **(5%)** Afisarea timpilor pentru rulara cu fiecare dintre cele doua euristici, pe fisierele de input.
- **(10%)** Afisarea in fisierele de output in formatul cerut.
- **(5%)** Optimizari: gasirea unui mod de reprezentare cat mai eficient.
- **(5%)** Comentarii. Explicarea algoritmului de generare a succesorilor, explicarea fiecarei euristici si dovedirea faptului ca are mereu valoarea mai mica decat costul real al drumului (nu trebuie demonstratie matematica, doar explicat clar). Explicarea pe scurt a fiecarei functii si a parametrilor.

Barem proiect_2: (punctajul e dat in procentaje din punctajul maxim al temei; procentajul maxim este 100%):

- **(10%)** La inceputul programului utilizatorul va fi intebat ce algoritm doreste sa foloseasca (minimax sau alpha-beta), cu ce simbol sa joace (la jocurile unde are sens aceasta intrebare) si care sa fie nivelul de dificultate a jocului (incepator, mediu, avansat). In functie de nivelul ales se va seta adancimea arborelui de mutari (cu cat nivelul ales e mai mare, cu atat adancimea trebuie sa fie mai mare ca sa fie mai precisa predictia jocului). Daca utilizatorul ofera un input gresit, i se repeta "intrebarea".
- **(5%)** Generarea starii initiale.
- **(20%)** Functia de generare a mutarilor (succesorilor).
- **(10%)** Realizarea mutarii utilizatorului. Utilizatorul va introduce de la tastatura optiunile sale de mutare. Va trebui verificata corectitudinea mutarilor utilizatorului: nu a introdus alt tip de date, a dat toate informatiile cerute, nu a facut o mutare invalida. Daca utilizatorul ofera un input gresit, i se repeta "intrebarea".
- **(10%)** Functia de testare a starii finale si stabilirea castigatorului.
- **(20%)** Doua moduri de estimarea scorului (pentru stari care nu sunt inca finale).
- **(15%)** Afisari. Afisarea timpului de gandire, dupa fiecare mutare, atat pentru calculator cat si pentru utilizator. Afisarea al cui este randul. Afisarea castigatorului sau a remizei daca este cazul, impreuna cu scorurile, atat pentru jucator cat si pentru calculator (in jocurile unde nu avem un scor clar specificat, poate fi dat chiar scorul calculat prin euristica). Afisarea timpului final de joc (cat a rulat programul) si a numarului total de mutari atat pentru jucator cat si pentru calculator (la unele jocuri se mai poate sari peste un rand).
- **(5%)** La fiecare mutare utilizatorul sa poata si sa opreasca jocul daca vrea (o optiune de exit - de exemplu raspunde cu cuvantul "exit" cand i se cere noua mutare), caz in care se vor afisa scorul lui si al calculatorului.
- **(5%)** Comentarii. Explicarea algoritmului de generare a mutarilor, explicarea estimarii scopului si dovedirea faptului ca ordoneaza starile cu adevarat in functie de cat de prielnice ii sunt lui MAX (nu trebuie demonstratie matematica, doar explicat clar). Explicarea pe scurt a fiecarei functii si a parametrilor.
- **Bonus (25%).** Posibilitatea de a juca jocul folosind o interfata grafica facuta in Python. Jocul trebuie sa poata fi rulat si in consola, chiar daca are interfata grafica. Pornirea in consola sau in interfata grafica se va face conform optiunii introduse de la tastatura (se afiseaza intrebarea la inceputul rularii).