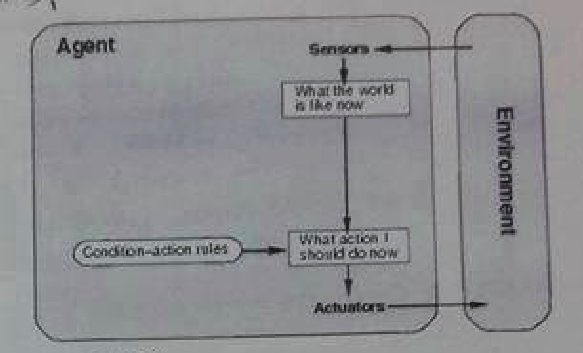
ИС - контролно 1

[Контролно 1](https://drive.google.com/drive/u/0/folders/1KTEZFgzLeztMKEzsRgcA9VwHDJqoRTPH) - папки със снимки + трерия линк

[Въпроси за контролно от Змийчаров](https://docs.google.com/document/d/1pIj2a8TaWdimnhQqaVDFbMwv8Niz2m_E5xX33v0bE-k/edit) - въпроси от Змийчаров, може би може още да се добавят

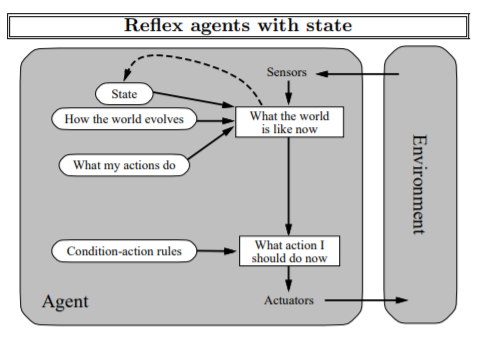
[Подготовка за контролно 1](https://docs.google.com/document/d/1qM0o7TeMsVJBa0HneRAmPb3mWh20KOh6T34v3RZEQT0/edit) - документ с всички въпроси, трябва да се провери в първия линк има ли допълнителни + трябва да се проверят правилно ли са отговорени

1. На кои от основните типове агенти е представената архитектура?
   1. **Прости рефлексни агенти (Simple reflex agents)**
   2. **Рефлексни агенти с памет (Model-based reflex)**
   3. **Целеви агенти (Goal-based)**
   4. **Максимизиращи ползата от своите действия (Utility-based)**

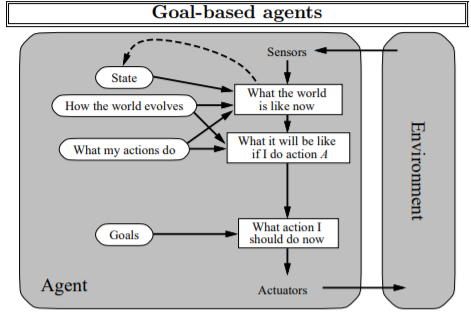
****

**// Другите типове агенти:**

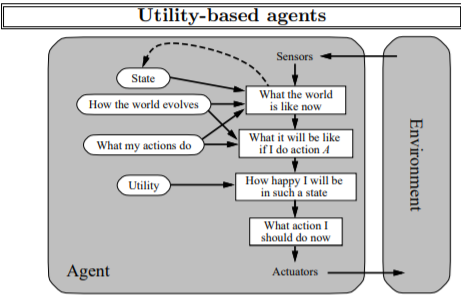
* **агенти на ЦРУ :Д:Д hi hi**
* **Рефлексни агенти с памет (Model-based reflex)**

****

* **Целеви агенти (Goal-based)**

****

* **Максимизиращи ползата от своите действия (Utility-based)**

****

1. Сравнете алгоритмите greedy best first search и A\*. Прилики и разлики?

**Отговор: Алгоритмите си приличат по това, че и двата са вариации на Best-First Search-a. Следователно и двата използват евристична функция върху всеки връх, за да определят кой ще бъде следващия посетен връх. И двата алгоритъма използват опашка, сортирана в намаляващ ред спрямо кой ще е най-скоро посетения връх. Различават се по това, че Greedy Best Search не е пълен и не е оптимален алгоритъм, докато А\* е пълен и оптимален алгоритъм. Също така Greedy Best Search алгоритъма взима върховете с най-ниска евристика(тези, които са най-близо до целта), докато А\* алгоритъма освен евристиката на даден връх, взема предвид и нужната цена, за да се стигне до него.**

1. Използвайте Мини-Макс алгоритъма за намиране на най-добър първи ход за игра на двама играчи без случаен елемент върху частичното дърво на игра по-долу. Стойностите от статичната оценка на позицията са посочени в листа. Напишете оценените стойности на междинните връзки и посочете правилното движение на Макс. Обхождане на възлите на дървото се прави отляво надясно.

**\_ max**

**/ | \**

**/ | \**

**/ | \**

**/ | \**

**/ | \**

**\_ \_ \_ min**

**/ \ / | \ / | \**

**/ \ / | \ / | \**

**/ \ / | \ / | \**

**10 -3 -5 2 -8 1 -4 3 max**

**Решение:**

**-3 max**

**/ | \**

**/ | \**

**/ | \**

**/ | \**

**/ | \**

**-3 -8 -4 min**

**/ \ / | \ / | \**

**/ \ / | \ / | \**

**/ \ / | \ / | \**

**10 -3 -5 2 -8 1 -4 3 max**

1. Посочете възлите в дървото по-горе, чиито резултати не е нужно да бъдат изчислени. Обяснете защо.

**Възелът със стойност 2 и възелът със стойност -8 нямат нужда да бъдат оценявани, защото можем след -5 да сме сигурни, че няма как най-малкият елемент в другите поддървета да е едновременно по-малък от -5 и по-голям от -3.**

**Възелът със стойност 3 не е необходим, защото можем след -4 да сме сигурни, че няма как най-малкият елемент в другите поддървета да е едновременно по-малък от -4 и по-голям от -3.**

1. При цена 1 на всяка дъга, кой от следните алгоритми не е оптимален?
   1. **Търсене в дълбочина**
   2. **Итеративно търсене**
   3. **Търсене с равномерна цена на пътя (uniform cost search)**
   4. **Търсене в ширина**
2. Кое от твърденията е вярно?
   1. **Търсенето в ширина изисква повече памет от търсене в дълбочина**
   2. **Итеративното търсене по нива изисква повече памет от търсенето в ширина**
   3. **Итеративното търсене по нива изисква повече памет от търсенето в дълбочина**
   4. **Търсенето в дълбочина изисква повече памет от търсенето в ширина**
3. Кое от следните твърдения не е вярно?
   1. **При търсене на път в граф, стратегията на търсене се определя от начина, по който разширява фронтът от вече изследваните пътища, и критерия за избор на стойност от фронта**
   2. **При търсенето в дълбочина фронтът се обработва като стек**
   3. **При евристичното търсене фронтът се сортира в намаляващ ред на евристичните оценки на пътищата, като на следващата стъпка се изследва първият елемент от сортираната редица**
   4. **По време на търсенето фронтът се разширява в посока към неизследваните възли, докато се достигне до целеви възел.**
4. От какво зависи ефективността на ɑ - β процедурата за намиране на печеливша стратегия при игри за двама играчи.
   1. **от броя на възлите в дървото**
   2. **от средния брой на наследниците на позициите в дървото**
   3. **от реда на обхождане на наследниците в дървото**
   4. **от броя на листата в дървото**
5. Кое от следните твърдения не е вярно
   1. **Семантичните мрежи за разлика от фреймовете използват само двуместни отношения**
   2. **Семантичните мрежи са графичното представяне на фреймовете**
   3. **Всяко знание представено със семантична мрежа може да бъде представено с фреймове**
   4. **Всяко знание представено с фреймове може да бъде представено със семантична мрежа**
6. Какво е кръстосване (crossover)?

**Описание:**

**Генериране на “наследници” двама или повече “родители”, селектирани от текущата популация на даден генетичен алгоритъм.**

**Значение:**

**Чрез crossover се разглежда естествения дарвински модел на еволюция, при който характеристиките на двама или повече родители се кръстосва с цел да се произведе наследник с по-добър fitness от този на родителите** [**Question 8 Flagged**](https://learn.fmi.uni-sofia.bg/mod/quiz/attempt.php?attempt=154266&cmid=228483&page=7#)**си.**

1. Сравнете алгоритмите търсене в лъч (beam search) и търсене в широчина (breadth-first)

Breadth-first:

* неинформиран
* пълен, ако дълбочината d е крайна, но не е оптимален, освен ако цената на всяка стъпка не е константа

Beam search:

* информиран
* не е пълен и оптимален

Едно предимство на Beam search е че използва по-малко памет от BFS, но не е пълен и е възможно да не разгледа някоя добра възможност заради лоша евристика на даден етап

1. Сравнете следните формализми за представяне на знания: Предикатна логика и Семантични мрежи

от слайдовете:

Why Logic is a Good Representation

Some of many reasons are:

– It’s fairly easy to do the translation when possible

– There are whole tracts of mathematics devoted to it

– It enables us to do logical reasoning

– Programming languages have grown out of logics

Semantic Nets

Adv.

easy to follow hierarchy, easy to trace association, flexible

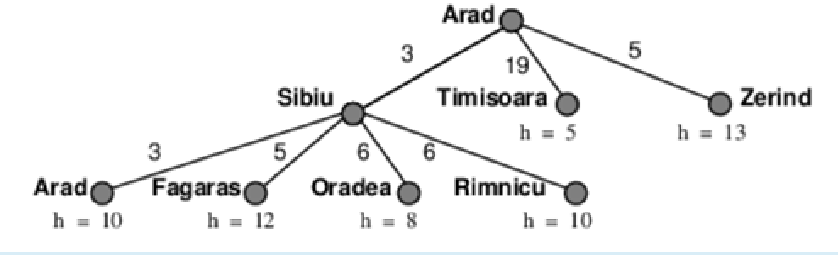
Disadv.

meaning attached to nodes might be ambigious

exception handling is difficult

difficult to program

1. Кои от следните твърдения са верни?
   1. **Експертните системи се използват във високорискови области като медицина, космос и др.**
   2. **Експертни системи предлагат възможност за описание на предложеното от тях решение**
   3. Експертните системи използват машинно самообучение за автоматично извличане на класифициращи правила
   4. Експертни системи имат широка употреба при планиране на действията на роботи
2. Следващата диаграма показва частично разширено дърво на търсене в пространство на състояния. На всяка дъга с етикет е отбелязана съответната цена на пътя, а в листата са обозначени стойностите на евристичната функция, която оценява разстоянието до листата.



Кое ще бъде следващото разширено листо при алчно (greedy) търсене?

* 1. Arad
  2. Fagaras
  3. Oradea
  4. Rimnicu
  5. **Timisoara**
  6. Zerind

1. Алгоритъмът A\* използва оценъчна функция f(n) = g(n) + h(n) - какво означава g(n)?
   1. Цената на пътя от n до целта
   2. Евристична оценка на цената на пътя от началото до n
   3. Евристична оценка на цената на пътя от n до целта
   4. Евристична оценка на цената на пътя до n до целта, която не надвишава реална цена
   5. **Цената на пътя от началото до n**
   6. Евристична оценка на цената на пътя от началото до n, която не надвишава реална цена
2. Казваме, че един алгоритъм за търсене е пълен ако:
   1. Винаги намира най-краткия път до решението
   2. Намира всички възможни решения
   3. Завършва своето изпълнение в полиномиално време по n (броя възли)
   4. Обхожда цялото пространство на състоянията
   5. **Винаги намира решение ако има такова**
3. Сравнете съждителните (пропозиционални) и предикатните логики от първи ред като формализми за представяне на знания. Посочете поне по едно предимство на всеки един.

от слайдовете:

Съждителни (пропозиционални) логики:

Пропозиционалната логика е логика на сложните съждения. Тя разкрива тяхната структура и изяснява логическите операции, чрез които се образуват от прости съждения нови сложни съждения посредством логическите съюзи.

Синтаксисът им включва:

* съждения (пропозиции) като P значещо “нещо е мокро”
* конектори: и, или, не, предполага, еквивалентно
* скоби, T (истина) и F (лъжа)

Семантика

* истината на твърденията може да се открие с таблици на истината

Логики от първи ред:

По-експресивни от съждителните логики

Синтаксисът им позволява:

* константи, променливи, предикати, функции и квантори

Нещо е истина за всички обекти (универсално) или за поне един обект (екзистенциално)

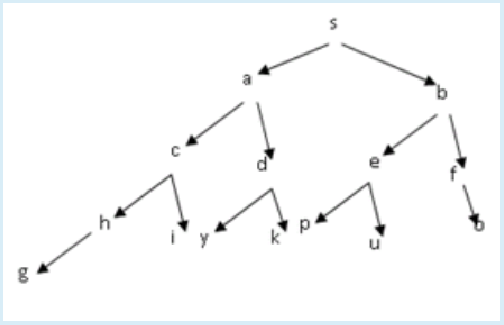
Семантика

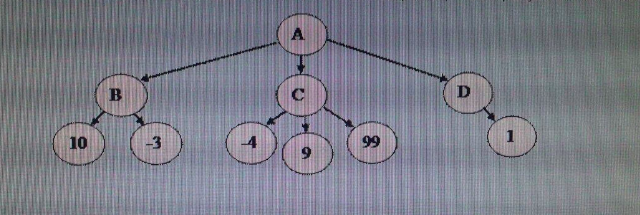
* истината на твърденията може да се открие с дедуктивни правила

**Предимства?**

**за съждителните - по-прости са за програмиране**

**за предикатните - могат да се образуват по-сложни изречения?**

1. Случайно започване от началото (Random restart) е техника, която се използва при:
   1. Алгоритмите за търсене в пространство на състоянията, когато зациклят
   2. Алгоритми за информирано търсене за търсене на по-добро решение
   3. Алгоритми за неинформирано търсене за търсене на по-добро решение
   4. **Локално търсещите алгоритми за търсене на по-добро решение**
2. Кое от следните твърдения е вярно за даденото на фигурата пространство от състояния ако търсенето започва от възела ‘s’, целевото състояние се представя от възела ‘g’, а обхождането се извършва отляво надясно?
   1. Търсенето по метода А\* е по-ефективно от търсенето в дълбочина и търсенето в широчина
   2. Търсенето в широчина е по-ефективно от търсене в дълбочина
   3. **Търсенето в дълбочина е по-ефективно от търсенето в широчина**
   4. Търсенето в дълбочина и търсенето в широчина са по-ефективни от търсенето по метода А\*
3. Използвайте Мини-Макс алгоритъма алфа-бета отсичане върху частичното дърво на игра по-долу, Макс е на ред. Играта не изключва случайност. Стойностите от статичната оценка на позицията са посочени в листата.

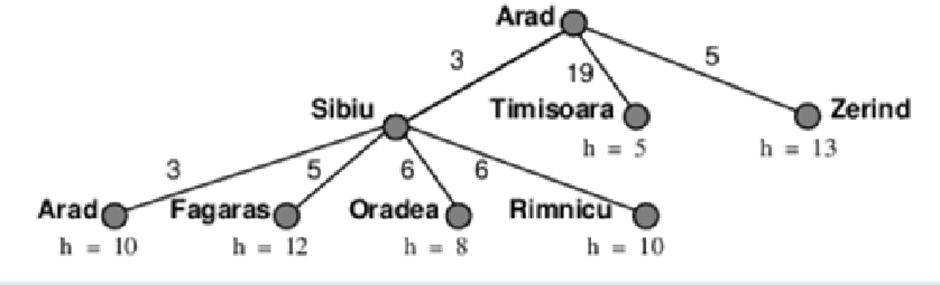


Посочете възлите в дървото, чиито резултати не е необходимо да пресмятате

* 1. -3
  2. **99**
  3. 1
  4. C
  5. **9**

1. Кой от следните алгоритми използва случайност за избягване на локален максимум?
   1. Hill climbing
   2. Best first search
   3. Local beam search
   4. **Simulated annealing**
   5. Gradient descent

1. Следващата диаграма показва частично разширено дърво на търсене в пространство на състояния. На всяка дъга с етикет е отбелязана съответната цена на пътя, а в листата са обозначени стойностите на евристичната функция, която оценява разстоянието до листата.



Кое ще бъде следващото разширено листо при търсене в широчина (BFS)?

* 1. Arad
  2. Fagaras
  3. Oradea
  4. Rimnicu
  5. **Timisoara**
  6. Zerind

3. Какъв основен формализъм за представяне на знания използват традиционните Експертни системи?

* 1. Пропозиционални логики
  2. Семантични мрежи
  3. Фраймово представяне
  4. **Продукционни (правила) системи**
  5. Понятийни графи

4. Сравнете генетичните алгоритми (Genetic algorithms) с алгоритъма за локално т itърсене в лъч (Local Beam search)

**Прилики:**

И двата са локално търсещи

Използват информирано търсене

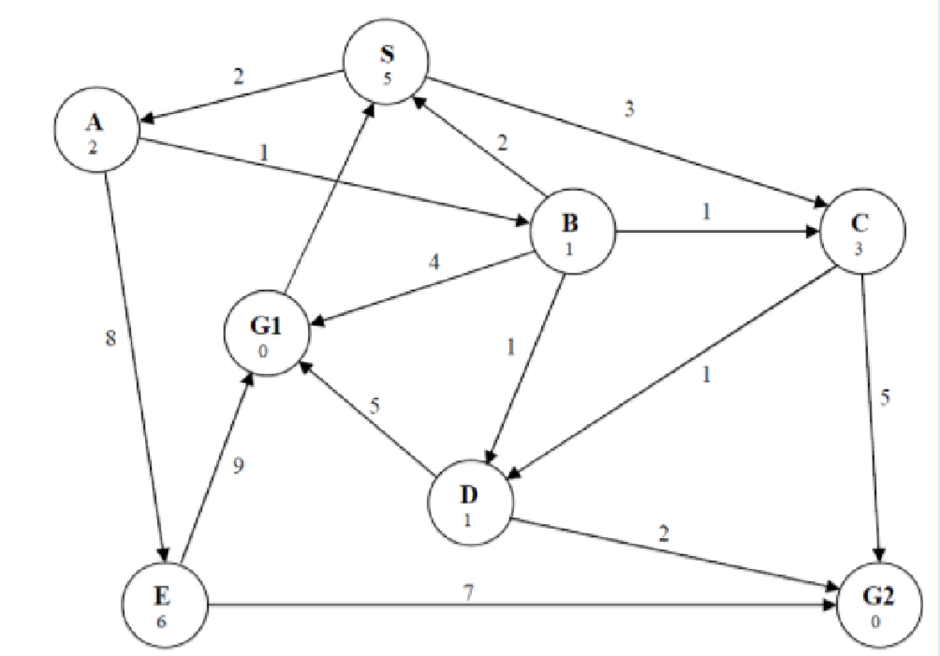
**Разлики:**

Генетичните алгоритми запазват най-добрите си състояния и генерират нови състояния от тях, а beam search избира от текущия връх следващите N на брой най-добри.

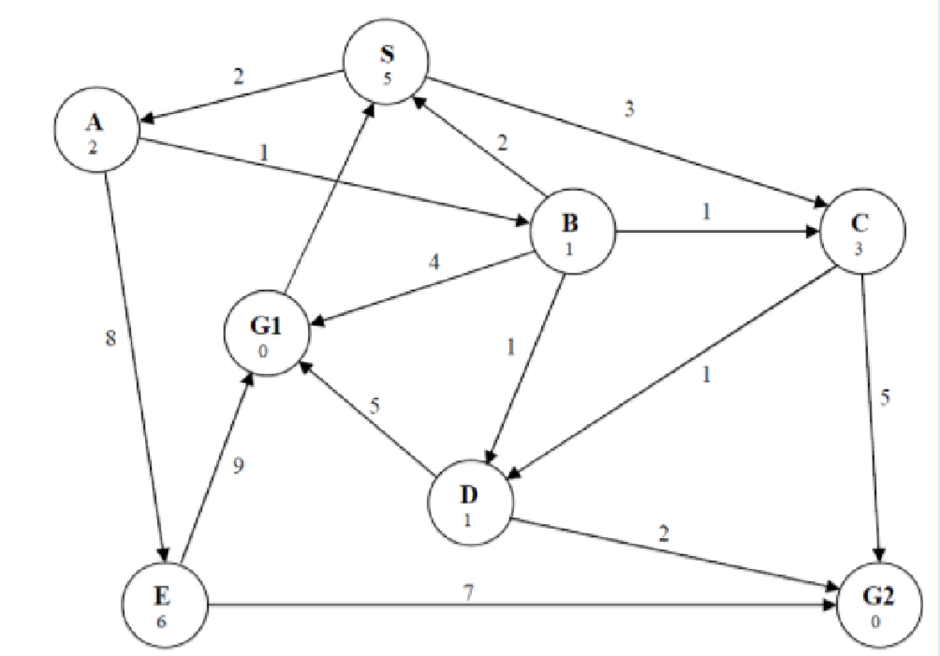
Генетичните алгоритми използват идеята за естествен подбор, докато beam search използва определена евристика

5. Пространство на състоянията е:

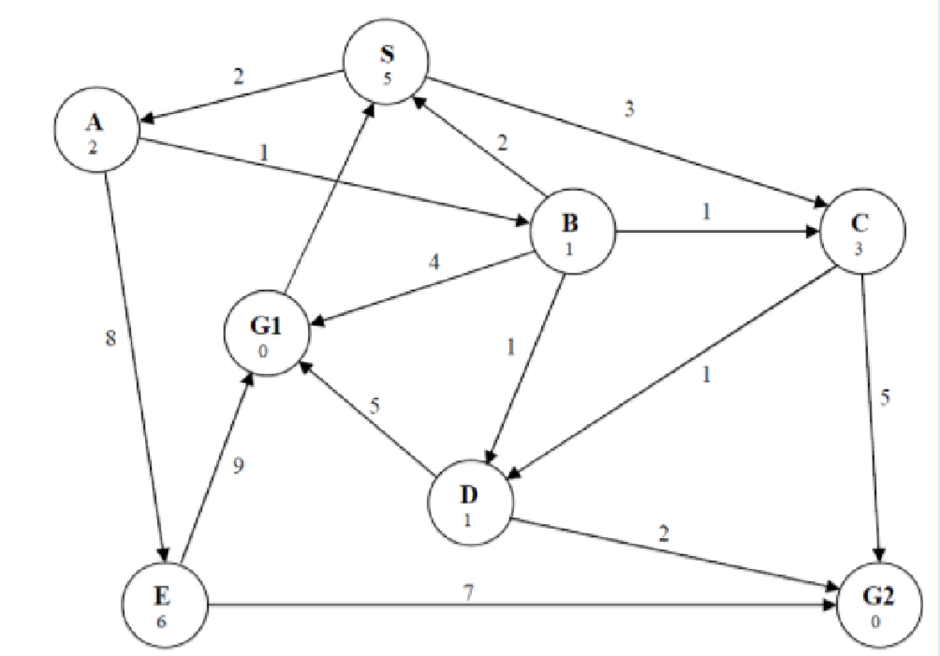
* 1. Съвкупността от всички състояния, през които минава път от началното състояние до някое целево състояние
  2. **Съвкупността от всички възможни състояния, които могат да се получат от дадено начално състояние**
  3. Множество от всички състояния на модела
  4. Едно описание на задачата в процеса на нейното решаване

6. Дадено е графично представяне на пространство на състоянията, където S е началният възел и G1 и G2 са целеви. Разходите за преминаване между възлите са означени по дъгите, а очакваните разходи до целта са вътре във възлите.  
За стратегията за търсене А\* посочете последователността от възлите, които ще бъдат обходени докато достигнем цел.  
Когато всичко друго е равно, възлите трябва да бъдат обработени в азбучен ред.  


* 1. **A B D C G2**
  2. A C B D G1
  3. A B C D G2
  4. A C B G1
  5. A C B E D G2

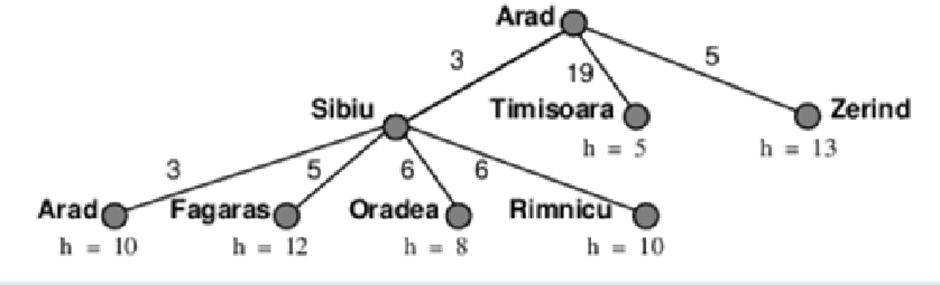
7. Дадено е графично представяне на пространство на състоянията, където S е началният възел и G1 и G2 са целеви. Разходите за преминаване между възлите са означени по дъгите, а очакваните разходи до целта са вътре във възлите.  
За стратегията за търсене Uniform Cost, посочете последователността от възлите, които ще бъдат обходени докато достигнем цел.  
Когато всичко друго е равно, възлите трябва да бъдат обработени в азбучен ред.  


* 1. A B D C G2
  2. A C B D G1
  3. **A B C D G2**
  4. A C B G1
  5. A C B E D G2

8. Дадено е графично представяне на пространство на състоянията, където S е началният възел и G1 и G2 са целеви. Разходите за преминаване между възлите са означени по дъгите, а очакваните разходи до целта са вътре във възлите.  
За стратегията за търсене в широчина (Breadth First search) посочете последователността от възлите, които ще бъдат обходени докато достигнем цел.  
Когато всичко друго е равно, възлите трябва да бъдат обработени в азбучен ред.  


* 1. A B D C G2
  2. A C B D G1
  3. A B C D G2
  4. A C B G1
  5. **A C B E D G2**

9. Следващата диаграма показва частично разширено дърво на търсене в пространство на състояния. На всяка дъга с етикет е отбелязана съответната цена на пътя, а в листата са обозначени стойностите на евристичната функция, която оценява разстоянието до листата.



Кое ще бъде следващото разширено листо при търсене с равномерна цена (uniform cost)?

* + Arad
  + Fagaras
  + Oradea
  + Rimnicu
  + Timidosts
  + **Zerind**

10.Колко вида основни търсещи стратегии и алгоритми знаете? Обяснете с какво се различават:

* Глобално търсене - разглежда се цялото пространство на състоянията. Ако се налага, може да се мине през всички състояния.
* Локално търсене - Разглежда се само локалното пространство, така че се гледат само състоянията, които са там. Ако решението е извън това пространство, няма как да се намери.
* **Неинформирани търсещи стратегии:** Неинформираните търсения използват само информацията, предоставена в самата дефиниция на проблема. Примери за неинформирани търсения са: DFS, BFS, UCS, DLS, IDS
* **Информирани търсещи стратегии**: Информираните стратегии за търсене имат информация за целевото състояние, което помага за по-ефикасно търсене. Тази информация се взема от евристична функция, която изчислява колко близо е сегашното състояние до целевото такова. Примери за информирано търсене са: Greedy Best-First Search, A\*, Beam Search, Hill Climbing

13. Сравнете генетичните алгоритми (Genetic algorithms) с алгоритъма за търсене в лъч (Local beam search):

**Genetic algorithms:**

* метаевристичен
* инспириран от процеса на естествен подбор(отбор?) natural selection
* намират добри решения на оптимизационни задачи и такива за търсене чрез операторите мутация, кросоувър и селекция

**Beam search:**

* локален
* не е пълен
* пази само k най-добри лъчи - O(k) по памет
* O(k\*d) по време
* не е оптимален

**Генетичните алгоритми имат по-голям шанс да намерят решение, тъй като генерират много такива.Те работят върху популация от решения и имат за задача да открият още по-добро такова. Локалното търсене в лъч има опасност да попадне в локални максимуми и плата. То също така може да се лута и да не намери решение или да му отнеме твърде много време преди да намери такова.**

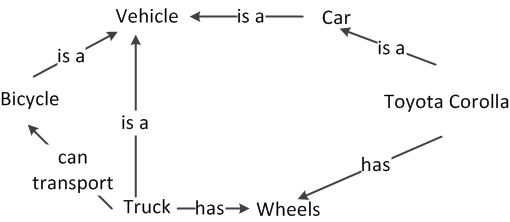
16. Формулирайте теста на Тюринг за Изкуствения интелект:

**Теста на Тюринг е тест, искащ да покаже дали компютърът може да мисли и да се държи интелигентно (като човек). За теста се изисква един човек, който си взаимодейства дистанционно с човек и компютър.Човекът изпраща съобщение и получава отговор от компютъра или другия човек. Първия човек трябва да разпознае дали е получил отговора от компютъра или втория човек от другата страна.**

1. Дайте примери за неинформирано търсещи алгоритми.
   1. **DFS (Depth First Search), BFS (Breadth First Search), IDS (Iterative Deepening Search), DLS (Depth Limited Search), UCS (Uniform Cost Search)**
2. Дайте примери за информирано търсещи алгоритми.
   1. **A\*, Hill Climbing, Best First Search, Beam search**
3. Какво е локално търсещ алгоритъм? Дайте пример за такъв алгоритъм.
   1. **Локално търсещ алгоритъм е този, който търси в ограничено подпространство на цялото пространство на състоянията. По този начин не разглежда всички състояния и може да не намери решение. Примери: Hill Climbing, Beam Search.**
4. Как оценяваме дали даден алгоритъм се представя адекватно върху конкретен проблем (Метрики за оценяване)?
   1. **Дали алгоритъмът е пълен и оптимален? Каква е сложността на алгоритъма по време/памет?**
5. Какво значи даден алгоритъм да е оптимален?
   1. **Винаги да намира най-доброто решение, ако съществува решение.**
6. Попълнете следната таблица:

| Criterion | BFS | Uniform-Cost | DFS | Depth-Limited | IDS |
| --- | --- | --- | --- | --- | --- |
| Complete? | **Yes, if b is finite** | **Yes, if step cost ≥ ε** | **No** | **Yes, if l ≥ d** | **Yes** |
| Time | **O(bd+1)** | **O(b[C\*/e])** | **O(bm)** | **O(bl)** | **O(bd)** |
| Space | **O(bd+1)** | **O(b[C\*/e])** | **O(bm)** | **O(bl)** | **O(bd)** |
| Optimal? | **Yes, if cost = 1 per step; not optimal in general** | **Yes** | **No** | **No** | **No, unless step costs are constant** |

1. Избройте предимства и недостатъци на BFS.
   1. **BFS търси в широчина. Ако съществува решение, BFS със сигурност ще го намери. Ако съществуват няколко решения, BFS ще намери най-оптималното решение. BFS използва памет, пропорционална на броя върхове във всяко ниво, което означава, че използва много памет.**
2. Избройте предимства и недостатъци на DFS.
   1. **DFS търси в дълбочина. Ако търсим в дърво с безкраен брой състояния, DFS може да не приключи. DFS може да не намери решение. DFS използва памет, пропорционална на дълбочината, на която търси, спрямо началното състояние, което означава, че използва малко памет.**
3. Каква е връзката между Uniform Cost Search и Дийкстра?
   1. **UCS е Дийкстра, който прекратява алгоритъма при намиране на най-краткия път до крайната цел. Дийкстра намира най-краткия път до всички състояния.**
4. Избройте предимства и недостатъци на IDS.
   1. **Предимството е че, използва DFS, което използва малко памет. Също така, модифицира DFS да бъде оптимален. Недостатъкът е, че времето, необходимо за достигане на целта е експоненциално.**
5. Кои от следните алгоритми за търсене в пространството от състоянията не са пълни?
   1. **търсене в дълбочина (DFS) // not complete**
   2. търсене в ширина (BFS) // complete, if b is finite
   3. търсене в ограничена дълбочина (DLS) // complete, if l ≥ d
   4. **търсене по метода на най-бързото спускане (best first search) // not complete**
   5. **hill-climbing // not complete**
   6. A\* // complete
   7. **търсене в лъч (beam search) // not complete**
   8. итеративно търсене по нива (IDS) // complete
6. Какви са основните типове дъги в семантична мрежа? Дайте по един пример за всеки един от изброените типове.
   1. **тип подмножество (описват релации от тип клас - суперклас)**
   2. **тип елемент (описват релации от тип обект - клас)**
   3. **тип функция (описват свойствата на обектите и класовете)**



1. От какво зависи ефективността при извличане на информация?
   1. **езика**
   2. **стила, жанра и предметната област на текста**
   3. **вида сценарий, от който се интересува потребителят**
2. Избройте основните компоненти на експертна система.

* **User interface (потребителски интерфейс)**
* **Explanation facility (обяснение)**
* **Knowledge base (база от знания)**
* **Working memory (работна памет)**
* **Inference engine (машина за изводи)**
* **Agenda (бележник)**
* **Knowledge acquisition facility (съоръжение за придобиване на знания)**

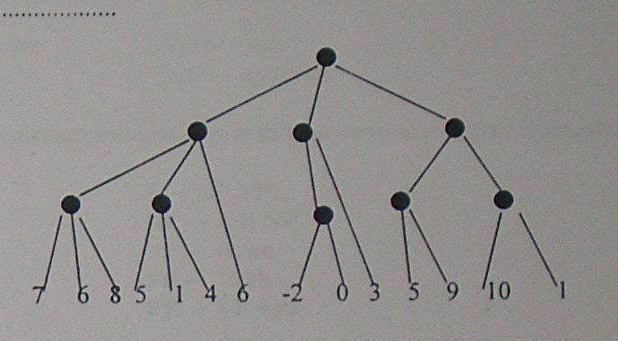
1. Кои са основните етапи на процеса на разпознаване на образи?

* **отделяне на признаци**
* **описание на обектите**
* **обучение на системата за разпознаване на образите, на базата на конкретни критерии и знание**
* **етап на разпознаване**

1. Как се изгражда поколение 0 при генетичните алгоритми?

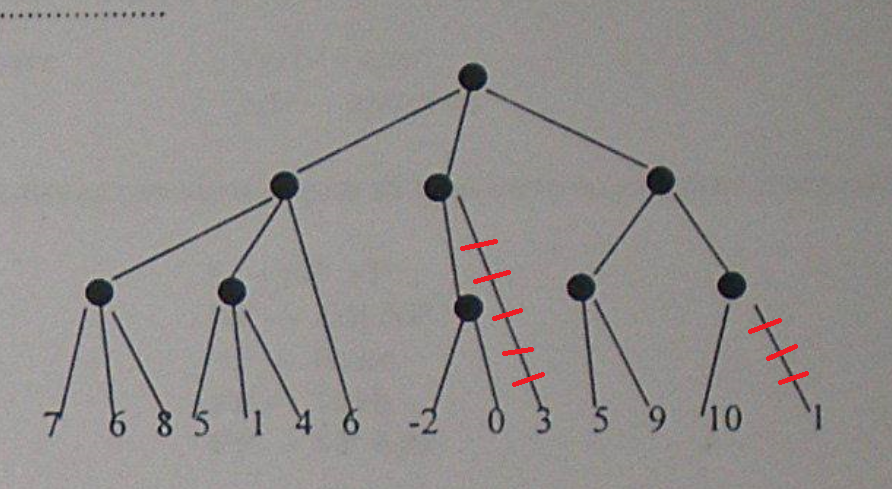
**Генерират се к случайни състояния.**

1. Каква оценка ще получи корена на дървото при алгоритъма mini-max за даденото дърво на играта, ако в него е на ход Max?



* **9**

1. Кои възли от игровото дърво ще бъдат отсечени при прилагане на алфа-бета алгоритъма за даденото игрово дърво?

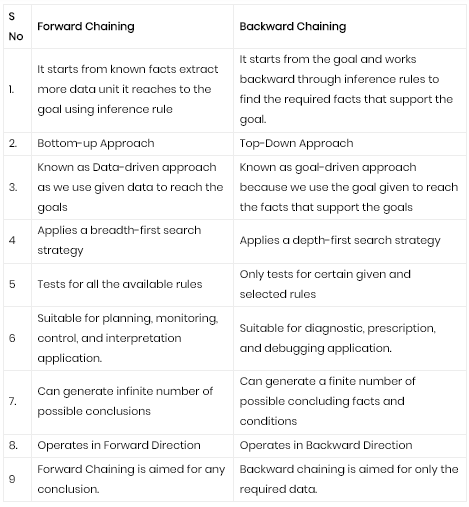


1. Кога правия извод е по-добър от обратния извод при използване на системи от правила и факти? Дайте кратък, но съдържателен отговор.

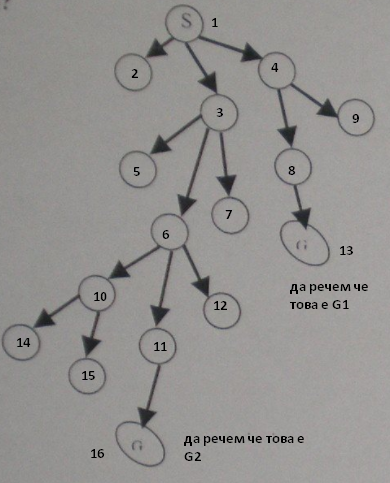
**Правият (forward chaining) извод е познат още като извод, управляван от данните (data-driven inference).**

**Обратният (backward chaining) извод е познат още като извод, управляван от целите (goal-driven inference).**

**Правият извод е по-добър от обратния в началото на изводния процес, когато входната информация води до някакво бързо междинно заключение.**

****

1. В какъв ред ще бъдат обходени възлите на даденото пространство от състояния при търсене в ширина, ако търсенето започва от възела S и целевото състояние е възела G, а обхождането става отляво надясно? Попълнете във възлите поредните номера.



**Обхождането по ширина започва от root възела, т.е. S (затова той е първата стъпка), и после последователно по редове.**

1. Колко възела от даденото пространство от състояния ще бъдат обходени до намиране на първото решение? (горната картинка)

**Тъй като има две целеви състояния G, нека първото е G1, а второто е G2. Да приемем, че първото решение е G1. Броят обходени възли е 13.**

1. Избройте основните видове аксиоми в ситуационното смятане (Situation Calculus).

* **effect axioms**
* **frame axioms**
* **successor-state axioms**

1. Колко типа неврони има в една невронна мрежа (в общия случай)?

**В невронната мрежа обикновено винаги съществуват входен и изходен слой от неврони, във входния се въвежда информацията към мрежата, след това сигналите от входните неврони преминават през един или няколко слоя от междинни (скрити) неврони, според топологията на невронната мрежа, като сигналите накрая стигат до изходния слой, откъдето се чете получената информация.**

1. Какви типове извод се използват при бейсови мрежи?

* **Диагностика - от следствието към причината**
* **Предсказване - от причината към следствието**
* **Междупричинен извод - между причините за дадено следствие**
* **Смесен извод - комбинация от горните три**

1. Избройте основните типове алгоритми при решаване на задачи с удовлетворяване на ограничения.

* **Backtracking**
* **Constraint Propagation**
* **Local Search**
* **Forward chaining**

1. Обяснете как работи алчното (Greedy) търсене
   1. **Greedy търсене работи като избира най-доброто следващо състояние чрез някаква евристична функция. Това обикновено прави алгоритъма бърз, но не оптимален.**
2. Прилики и разлики между Greedy Search и A\*
   1. **И двата алгоритъма разчитат на някаква евристика, но Greedy Search разчита на евристика, която взима в предвид само текущото състояние, докато А\* използва и цената/дистанцията до текущото състояние за евристиката. Това прави A\* оптимален, а Greedy search - неоптимален.**
3. Какво е евристична функция?
   1. **Функция, която оценява колко е добро дадено състояние (колко е близко до финалното). Тази функция трябва да е бърза (по-бърза от колкото точен алгоритъм, който ще намери точно колко далеч сме от крайното състояние). И трябва да ни дава насока кои върхове са по-добри за обхождане.**

**Консистентна евристика е такава, която е по-малка или равна за всеки следващ ноуд от предишния и в goal state-a да е 0.**

**Приемлива евристика е такава, която е по-малка или равна от реалната цена.**

**Доминантна евристика над друга евристика е евристика, която има по-голяма стойност от друга, понеже схожда по-бързо и по-бързо се стига до решение.**

1. На какво е равна оценяващата функция при A\*: f(n) = ?. Обяснете
   1. **Цената на пътя от началното състояние до текущото състояние + евристиката. Това е така, защото А\* оценя състоянията, като събира цената на пътя до състоянието g(n), и цената от състоянието до целевото такова - h(n). Тогава f(n) = g(n) + h(n).**
2. Предимства и недостатъци на Best First Search (Greedy search)
   1. **Предимство е, че работи бързо, но недостатък е че може да не открие оптимално решение.**
   2. **Предимства: може да преминава от BFS към DFS и обратно и да се възползва от предимствата на двете. По-оптимален е в сравнение с DFS.**
   3. **Недостатъци: голям шанс да се заседне в цикъл. Не е оптимален.**
3. Предимства и недостатъци на A\*
   1. **Предимство на А\* е че е оптимален и пълен.**
   2. **Недостатък е, че използва много памет.**
4. Какво представлява алгоритъмът Удовлетворяване на ограниченията (Constraint Satisfaction Problem)?
   1. **Алгоритъм при който се интересуваме дали е възможно да постигнем крайната цел, а не как я постигаме.**
   2. **Проблемът за удовлетворяване на ограниченията (CSP) е проблем, който изисква неговото решение да бъде в рамките на някои ограничения, състоящ се от краен набор от променливи, набор от домейни и краен набор от ограничения.**

**CSP:**

**- състояние се определя от променливи Хi със стойности от Домейн Di**

**- целево състояние: множество от ограничения, върху разрешените променливи**

**CSP = {X, D, C}**

**X = {x1, x2, x3, ..., xn} - всички стойности на променливи**

**D = {d1, ..., dn} - ограничения върху х, домейн от допустими стойности**

**C = {c1, ..., cm} - ораничения където ci се налага върху х' и д', и х' е подмножесто на х.**

1. Посочете алгоритми от класа задачи Удовлетворяване на ограниченията и примерни проблеми, които се решават с тези алгоритми.
   1. **Обикновено използва Backtracking стратегия или Hill climbing + random restart, за да намери крайното решение. Примерни задачи са N-Queens, Съставяне на графици, разписания.**
   2. **Backtracking - DFS**
   3. **Constraint propagation - Arc consistency**
   4. **Local search - min conflicts (hill climbing)**
2. Избройте предимства и недостатъци на Търсене в лъч (Beam Search)
   1. **Предимството е че е Beam Search намира решение бързо и използва малко памет. Това е, защото не трябва да съхранява всички последователни състояния в опашката.**
   2. **Недостатък е, че не е пълен, т.е. може и да не открие решение. Също така не е оптимален - ако върне решение, може да не е най-доброто.**
3. Избройте предимства и недостатъци на Hill Climbing
   1. Предимства:

* линейно време
* константа сложност по памет

Недостатъци

* не е пълен, local search
* не е оптимален
* забива се в локални екстремуми

1. Какво е генетичен алгоритъм?

**Отговор**: Генетичен алгоритъм е търсеща метаевристика, която имитира процеса на естествения подбор.

**От лекцията:** = stochastic local beam search + generate successors from pairs of states (генериране на наследниците чрез кръстосване на състояния)

Имаме текущите състояния, оценяваща функция. От тези състояния избираме най-добрите. Кръстосваме ги - специфичен начин за генериране на наследници. Мутации - понякога в гените се получава някаква случайна промяна. От там -  **стохастика** - рандомизиране на local beam search-a.

Генетичен алгоритъм за задачата за n-те царици.

Генетичните алгоритми са различни от еволюцията!

1. Какво означава терминът еволюция в генетичните алгоритми?

**Отговор**: Еволюция е създаване на поколения по-добри индивиди посредством промяна чрез възпроизводство и избирателно оцеляване на част от наследниците. Индивидите от популацията представляват възможни решения.

1. Какво е селекция?

**Отговор**: Селекция означава естествен подбор. Това е избора на индивидите с най-добри гени от текущото поколение за следващо репродуциране.

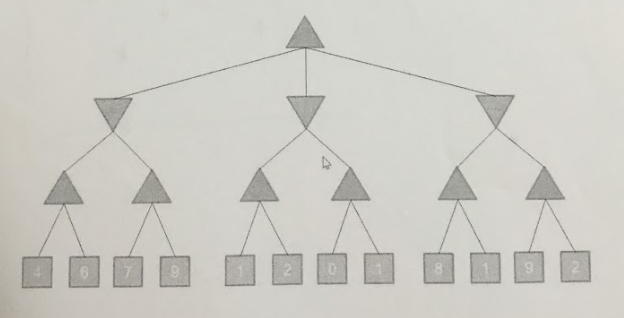
1. Какво е мутация?

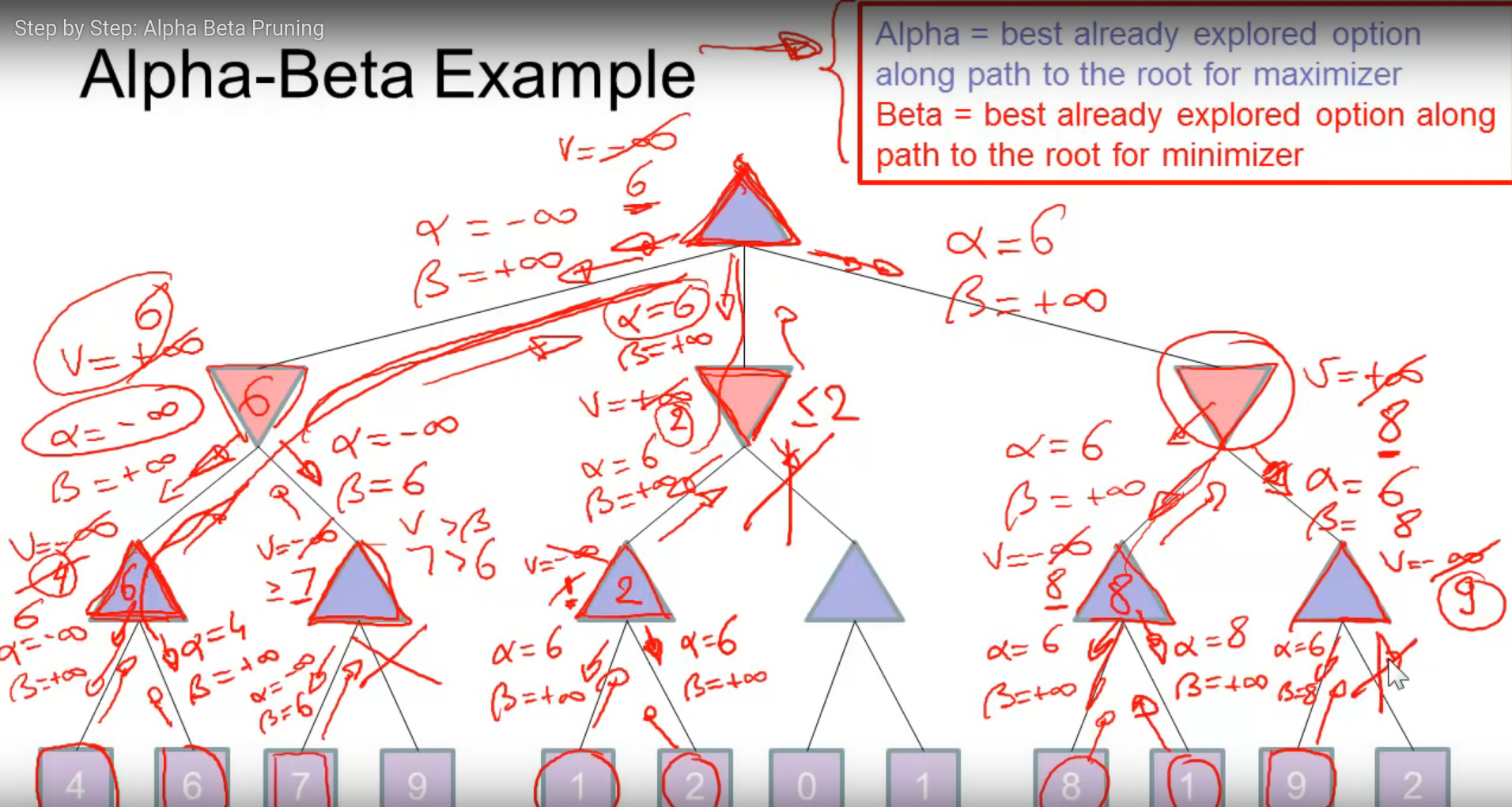
**Отговор**: Повишава биологичното разнообразие. Неговата идея е да промени произволен ген. Примерни функции за мутация са swap, reverse, insert.

Мутация - произволно изменение на гени.

1. Какво е важно за евристичната функция?
   1. **Да се пресмята бързо и лесно. Да връща число, което да оценява даденото състояние и числото да е по-малко ако състоянието е по-близо до крайната цел.**

**// Най-често се използва евристична оценяваща функция h, която връща като резултат приблизителната стойност на определен ресурс, необходим за достигане от оценяваното състояние до целта.**

1. Кой алгоритъм е по-бърз: BFS или DFS?
   1. **Зависи от задачата. Върху крайни графи/дървета имат еднаква сложност по време. Върху безкрайни графи/дървета, ако няма решение също се представят еднакво зле. Но върху безкраен граф/дърво в което съществува крайно състояние BFS ще го намери, докато DFS може и да не го намери, което прави DFS със сложност по време безкрайност.**
2. Какво е Best First Search?
   1. **Алгоритъм, който приоритизира кандидатите за обработване спрямо някаква метрика (евристика). Примерно А\* е такъв.**
3. Какво е Beam Search?
   1. **Алгоритъм, който държи само предефиниран константен брой кандидати. Т.е приоритетната опашка е с ограничен размер. Използва се за да пестим памет. Може да правим Beam Search само когато имаме Best First Search.**
4. Каква структура се използва при DFS?
   1. **Стек**
5. Каква структура се използва при BFS?
   1. **Опашка**
6. Запълнете следното дърво и кажете може ли да се приложи и къде алфа-бета отсичане. Макс е на ход.



<https://www.youtube.com/watch?v=_i-lZcbWkps&ab_channel=CrackConcepts>

**Въпроси от консултацията:**

Каква е разликата между предикатни логики от първи ред и от по-висок ред?

Логиките от по-висок ред са по-скъпи от тези от първи ред и позволяват количествено определяне върху функции и предикати, както и обекти в сравнение с първия ред, който работи само с обект?

**First Order Predicate Logic**

- **Syntax allows – Constants, variables, predicates, functions and quantifiers**

- **So, we say something is true for all objects (universal) – Or something is true for at least one object (existential)**

- **Semantics – Working out the truth of statement**

- **This can be done using rules of deduction**

**Higher Order Predicate Logic:**

· **More expressive than first order predicate logic**

· **Allows quantification over functions and predicates, as well as objects**

· **For example – We can say that all our polynomials have a zero at 17: f (f(17)=0).**

· **Working at the meta-level – Important to AI, but not often used**

Кое от следните неща е вярно за A\*?

**a)** **f(n) = g(n) + h(n)**

b) f(n) = h(n)

c) h(n) е точният път от n до целта

d) g(n) е приближение на пътя от n до целта

Какво е AI?

**Система, която действа като човек (разумно)**

**Други:**

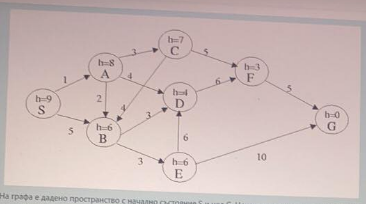
1. Какво е частично нареден план?

**Всеки алгоритъм за планиране, който може да постави две действия в един план без да има наредба кой да е пръв, е частично нареден план.**

1. За да може да премине успешно теста на Тюринг за ИИ, системата:
   1. **трябва да мисли като човек**
   2. трябва да действа като човек
2. Кой от следните алгоритми се използва за решаване на задачата за удовлетворяване на ограниченията?
   1. Uniform cost search
   2. Интерактивно спускане по нива
   3. **търсене в дълбочина с връщане (backtracking)**
   4. търсене в широчина
   5. А\*

* Min-conflicts

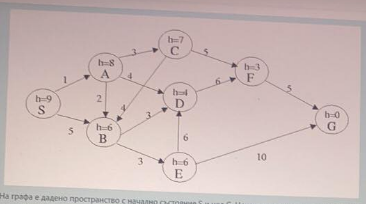
1. Кой от следните алгоритми е оптимален?
   1. **търсене с равномерна цена на пътя (uniform-cost search)**
   2. търсене в ширина
   3. итеративно търсене по ниво (iterative deepening search)
   4. търсене в дълбочина
2. На графа е дадено пространство с начално състояние S и цел G. Цената на преходите са по дъгите, а евристичната оценка h, на разстоянието от състоянието до целта, е… приемем, че конфликти при равни оценки се разрешават, като се избере състоянието, което идва първо по азбучен ред. Какъв е път с най-малко разходи от S до G?



Изберете едно:

* 1. S, A, B, C, D, E, F, G
  2. S, B, D, F, G
  3. S, A, B, D, F, G
  4. **S, A, C, F, G**
  5. S, A, B, D, C, E, F, G

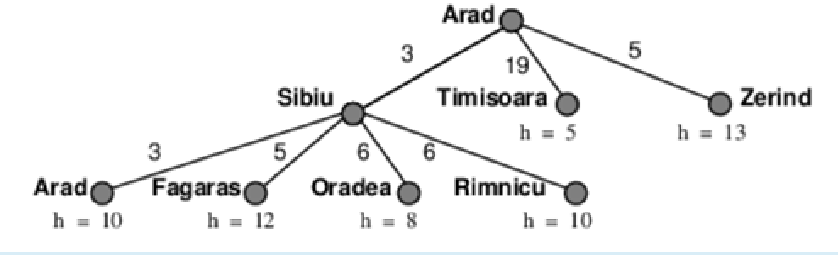
1. На графа е дадено пространство с начално състояние S и цел G. Цената на преходите са по дъгите, а евристичната оценка h, на разстоянието от състоянието до целта, е… приемем, че конфликти при равни оценки се разрешават, като се избере състоянието, което идва първо по азбучен ред. Какъв е редът на състоянията, разширен с помощта на търсене в дълбочина? Да приемем, че DFS се прекратява веднага щом достигне състоянието на целта?



Изберете едно:

* 1. S, A, B, C, D, E, F, G
  2. S, B, D, F, G
  3. **S, A, B, D, F, G**
  4. S, A, C, F, G
  5. S, A, B, D, C, E, F, G

1. Следващата диаграма показва частично разширено дърво на търсене в пространство на състояния. На всяка дъга с етикет е отбелязана съответната цена на пътя, а в листата са обозначени стойностите на евристичната функция, която оценява разстоянието до листата.



Кое ще бъде следващото разширено листо при А\* търсене?

* 1. Arad //то щеше да е ако не беше стартовото състояние
  2. Fagaras
  3. **Oradea**
  4. Rimnicu
  5. Timisoara
  6. Zerind