1. Върху картата на Румъния приложете посочените по-долу алгоритми, като срещу всеки от тях напишете състоянията/градовете, които ще бъдат обходени. Тръгвате от Arad и целта е да стигнете до Bucharest. Помислете кога бихте предпочели всеки един от алгоритмите.

- DFS: **- BFS:**

- IDS: **- UCS:**

- Best-first Search: **- A*:**

2. Колко вида търсене знаете. Дайте определение/дефиниция за всяко едно. Посочете други проблеми, които могат да бъдат сведени до търсещ проблем.

Информирано и неинформирано търсене. Неинформираното търсене използва само информацията налична при дефинирането на проблема. Информираното търсене използва допълнителна информация от агента, свързана с карта на проблема, цена на действията, апроксимация на решенията и т.н.

3. За всеки вид търсене от въпрос две посочете съответните алгоритмите от въпрос едно.

Неинформирано търсене: Breadth-first search; Uniform-cost search; Depth-first search; Depth-limited search; Iterative deepening search; Bidirectional Search.

Информирано търсене: best-first search: greedy search; A* search.

* Local search: Hill-climbing; Beam Search; Simulated annealing; Genetic algorithms.

4. Посочете всички 4 основни метрики, по които можем да оценим даден алгоритъм и дайте определение за всяка една.

Метрика за оценяване	**Определение**

5. Оценете следните алгоритми на база метриките от въпрос 4. Като „5-та метрика“ отбележете каква структура използва даденият алгоритъм.

Метрика	**DFS**	**BFS**	**Depth-limited**	**IDS**	**UCS**	**Best-first**	**A***
Complete	No	Yes	Yes, *if $l \geq d$*	Yes	Yes	No, *(can get in loops)*	Yes, *(unless there are infinitely many nodes with $f_i \leq f(G)$)*
Optimal	No	Yes, *(if graph is unweghted)*	No	Yes, *(if graph is unweghted)*	Yes	No	Yes
Time	b^m	b^{d+1}	b^l	b^d	b^⌈1 + $C*$ /Ɛ⌉	b^m	Exponential in [relative error in h x length of sol.]
Space	bm	b^{d+1}	bl	bd	b^⌈1 + $C*$ /Ɛ⌉	b^m	Keeps all nodes in memory
Структура	Stack	Queue	Stack	Stack	Priority Queue	Priority Queue	Priority Queue

6. Комбинация от кои алгоритми е „Итеративното търсене в дълбочина“. Изброите предимствата и недостатъците му (какво от кой алгоритъм взима).

Пълен е. Оптимален е, ако теглото на ребрата е еднакво. Комбинация от BFS и DFS като взима от предимствата и на двата. По-бърз е от BFS. По-малко памет. Подходящ е за неинформирано търсене в огромни пространства, където дълбочината на решението не се знае.

7. Каква е връзката между Uniform Cost Search и Dijkstra.

Dijkstra е вариант на Uniform Cost Search, при който няма целево състояние и продължава докато не премахне всички върхове от приоритетната опашка (открива най-краткия път до всеки един връх).

8. Каква е връзката между алчно (Greedy) търсене, Best-first Search и A*.

9. На какво е равна оценяващата функция на A*: *f(n) = ?* Обяснете.
f(n) = d(n) + h(n) -> Реалния път до текущия връх n плюс евристичната функция определяща цената на пътя до най-близкия целеви връх.

10. Какво е евристична функция и какво е най-важното за нея?
Да е равна на 0 в целевите състояния и изчисляването и да не е свързано с голям разход на време и памет.

11. По какво си приличат и различават Greedy Best-First Search и A*?

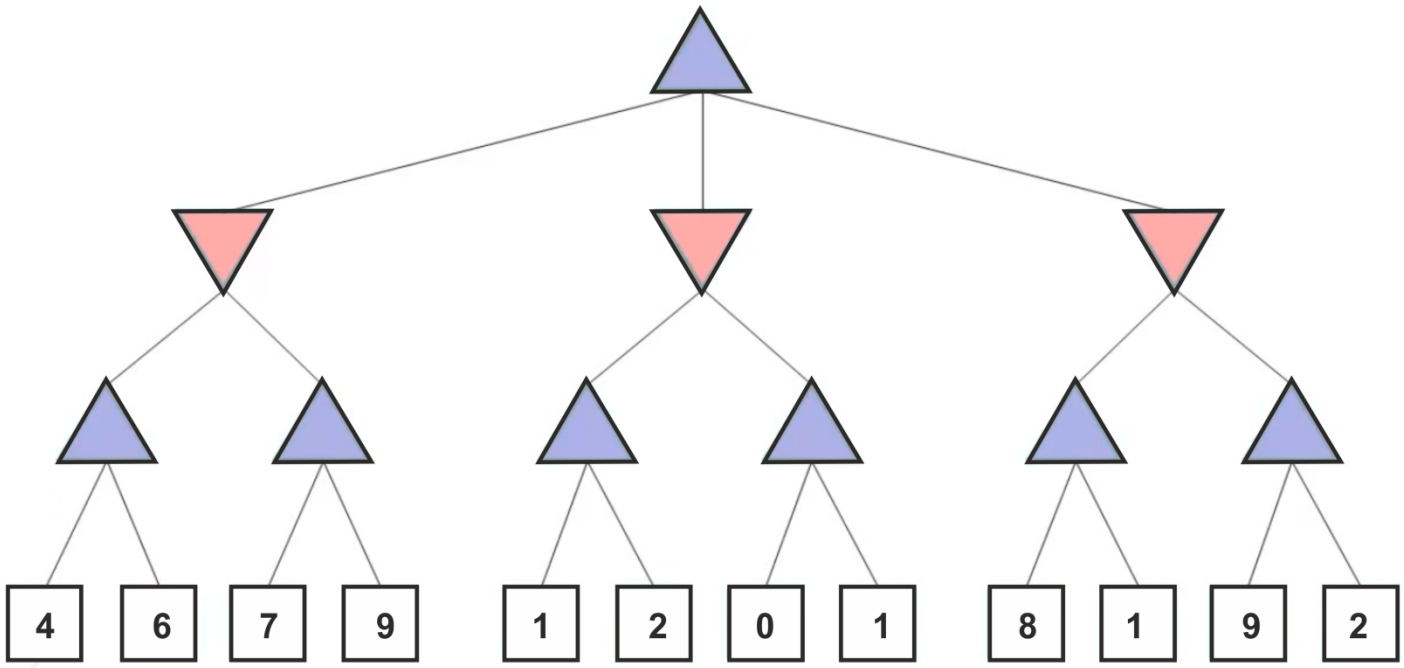
12. Какво представлява задачата за Удовлетворяване на ограниченията (Constraint Satisfaction Problem)?
Математически проблем, определен като набор от обекти, чието състояние трябва да отговаря на редица от ограничения. Често проявяват висока сложност, изискваща комбинация от евристични и комбинаторни методи за търсене, за да бъде решен в рамките на разумен срок.

13. Дайте примери за алгоритми, които се използват за решаването на проблеми от класа задачи CSP и съответно конкретни проблеми, които могат да се сведат до CSP.
Проблеми: Map Coloring Problem; Sudoku; 8 queens.
Обикновено проблеми от тип Удовлетворяване на ограниченията се решават чрез използването на определена форма на търсене. Най-използваните варианти са Backtracking (Използва DFS), Constraint propagation и Local Search (Min-Conflicts).

14. Какво е генетичен алгоритъм?
Генетичен алгоритъм е търсеща евристика, която имитира процеса на естествения подбор. Тази евристика се използва рутинно за генериране на полезни решения за оптимизационни и търсещи проблеми. Генетичните алгоритми принадлежат към големия клас на еволюционните алгоритми, които генерират решения за оптимизационни проблеми използвайки техники, вдъхновени от естествената еволюция като наследяване, мутация, селекция, кръстосване.
15. Какво означава терминът еволюция в генетичните алгоритми?
Еволюция е създаване на поколения по-добри индивиди посредством промяна чрез възпроизводство и избирателно оцеляване на част от наследниците.

16. Изброите трите основни етапа при генетичните алгоритми и дайте определение за всеки един.
Селекция: Селекция е етап от генетичния алгоритъм, в който индивидуалните геноми се избират от популацията за по-късно размножаване (използвайки кръстосване).
Кръстосване: Това е генетичен оператор използван за промяна програмирането на хромозома или хромозоми от едно поколение към друго. Аналогично е на възпроизвеждането при биологичното кръстосване, на което се основават генетичните алгоритми.
Мутация: Мутация е генетичен оператор, използван за поддържане на генетичното разнообразие от едно поколение на популацията от хромозоми на генетичния алгоритъм към друго.
17. Какво е локално търсещ алгоритъм. Дайте пример за такъв алгоритъм.
Локално търсещите алгоритми са непълно задоволяващи. Те може да открият решението на проблема, но може и да се провалят, дори и да съществува решение. Те подобряват итеративно зададеното състоянието на променливите величини. На малки стъпки, малък брой от променливите величини променят стойността си, така че да задоволят посочените ограничения.
Hill-climbing; Simulated annealing; Genetic algorithms; Minimum Conflicts.
18. Какви са основните критерии, по които се определят различните типове игри. Обяснете.

19. Запълнете следното дърво използвайки алгоритъма минимакс и кажете може ли да се приложи и къде алфа-бета отсичане.



20. На какво се базира ефективността на алфа-бета отрязването?
Реда на обхождане на наследниците.