Machine Learning & Data Mining

Introduction to AI

3 hours : One two-sided handwritten page allowed.

1 Logic (5 pts)

Translate the following sentences into first order logic formula using only the necessary predicates (e.g. roadrunner/1 (or r/1), catch/2, chase/2, saybeep/1, ...).
Put the formulas into clausal form.
Use the formal system Resolution (draw the reasoning graphically) to prove the conclusion.

1. Every coyote chases some roadrunner.
2. Every roadrunner who says "beep-beep" is smart.
3. No coyote catches any smart roadrunner.
4. Any coyote who chases some roadrunner but does not catch it is frustrated.
5. (CONCLUSION) If all roadrunners say "beep-beep", then all coyotes are frustrated.

2 Formalization and Search (6 pts)

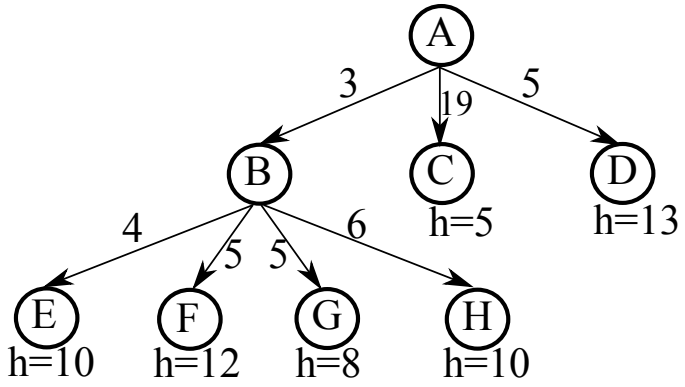
2.1 Problem formalization : Crossing the bridge (2 pts)

Four persons come to a river at night and want to cross it. There is a narrow bridge, but it can only hold two persons at a time. They have one flashlight and, since it is dark, it is impossible to cross the bridge without it. One person can cross the bridge in 1 minute, another one in 2 minutes, another one in 5 minutes, and the last one in 10 minutes. When two people cross the bridge together, they must move at the slowest person's pace.

Formalize the problem. I recall that you have to define
– a suitable representation of the states,
– the initial state and the goal state,
– define the operators (and the preconditions).

2.2 The next node (1,5 pts)

Consider the following search tree where each directed edge is labeled with the cost of the corresponding operator, and the leaves are labeled with the value of a heuristic function h. For uninformed searches, assume the expansion is performed from left to right, and in case of tie in alphabetical order.

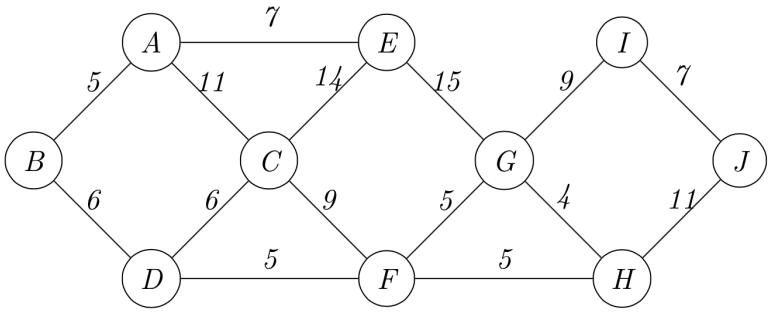


Which node will be expanded next by each of the following search algorithms ? (explain in **one** sentence your answer)

1. Uniform-cost search
2. Depth-first search
3. Greedy best-first Search
4. A* algorithm

2.3 A* (2,5 pts)

Consider the following graph where each edge (x, y) is labeled with the cost of reaching y from x. The objective is to reach J from A.



Moreover, we consider the heuristic function h (defined in the next table) which estimates the cost of reaching J from each vertex.

A	B	C	D	E	F	G	H	I	J
30	26	20	21	25	10	12	8	5	0

– Draw the search tree obtained by applying A* algorithm with h on this graph.
– Detail the steps of the algorithm.
– Give the shortest path from A to J and its cost.

3 Prolog (6pts)

3.1 Maximum length (3pts)

The predicate `max_length(+L,-ML)` computes the length of the longest sub-list of a list of lists. For example, the first answer of prolog to the query `?- max_length([[1,2],[1,4,5]],ML).` is `ML = 3`. For the following questions, you can use the predicate `length` seen during the course.

1. Write two versions of the `max_length` predicate, one **without** an accumulator and one **with** an accumulator. Which one is the most efficient ? Why ?
2. Write a predicate `all_max_length(+L,-R)` which gives in `R` all the sub-lists of `L` with a length equal to the length of the longest sub-list of `L`.

3.2 Call tree (3pts)

Given the following predicates :

```
prop(X,Y):-
    X <= Y.

verify([],[]).
verify([X|L],[1|T]):-
    prop(X,2),
    verify(L,T).
verify([X|L],[0|T]):-
    verify(L,T).
```

1. What is prolog's first answer to the query `?- verify([2,3,1],R).` ?
2. Draw the call tree for the query `?- verify([2,3,1],R).`.
3. Prolog gives multiple answers. Propose two ways to modify the previous predicate to only keep the first answer. Which solution is better ? Why ?

4 Constraint Logic Programming (3 pts)

The aim is to find the digit marked with a "?" in the following operation made in base 10 (no number starts with a 0) :

$$\begin{array}{cccc} & . & . & . \\ \times & & 7 & . \\ \hline . & . & 1 & . \\ 8 & . & . & \\ \hline . & . & . & ? \end{array}$$

Write a GNU prolog program that would solve this problem.