Student Name: SOLUTION

Student Number: *mark total 68*

2019 CAB320 Mid Semester Exam Instructions

Do NOT remove the staple from this booklet. Write your name and student number at the top of the first page. Attempt all the questions. All calculators are allowed. This is individual work. **No notes allowed**. If you have any question, call the invigilator. Marks for each question as indicated. **Write your answers in the boxes of the corresponding questions (or on the dots standing for missing code)**. Good luck!

Question 1

Two mobile robots *R1* and *R2* located in an *NxN* grid maze have to meet. It does not matter in which cell of the maze they meet. In each time step, *R1* and *R2* simultaneously move in one of the following directions: {NORTH, SOUTH, EAST, WEST, STOP}. You must devise a plan which positions them together, somewhere in the grid, in as few time steps as possible. Passing each other does not count as meeting; they must occupy the same cell at the same time.

(1.a) Formally state this problem as a single-agent state-space search problem.

States: express the state space in set comprehension form

{(x1,y1),(x2,y2) for x1,x2,y1,y2 in range(N)**4} where (x1,y1) and (x2,y2) are the coords of R1 and R2

Size of the state space:

N**4

Branching factor of the search tree:

5 possible actions for each robot, therefore branching factor is **5**2**

as the action of the agent moves the two robots in a single time step.

Goal test function:

Goal test function is the predicate: x1==x2 and y1==y2

(4 marks)

(1.b) Give a non-trivial (different from zero) admissible heuristic for this problem. Assume that the cost of moving a single robot between two adjacent cells is *1*.

Manhattan distance / 2 : floor(0.5 * (abs(x1-x2)+abs(y1-y2)))

(3 marks)

(1.c) Mark with a tick all of the following **tree search** methods which are guaranteed to output optimal solutions to this problem:

(i) DFS no, could even not terminate

(ii) BFS yes

(iii) UCS yes, same as BFS

(iv) A* with an admissible heuristic yes

(v) A* with heuristic that returns zero for each state yes

(vi) Greedy search with an admissible heuristic no

(6 marks)

(1.d) Assuming that *h1* and *h2* are admissible, mark with a tick all of the following expressions which are also guaranteed to be admissible:

(i) $h1 + h2$ no

(ii) $h1 * h2$ no

(iii) max($h1$, $0.3*h2$) yes

(iv) min($h1$, $3*h2$) yes

(v) $0.94*h1 + 0.08*h2$ no

(5 marks)

Question 2

You are designing a menu for a special event. There are several choices, each represented as a variable: (**A**)ppetizer, (**B**)everage, main (**C**)ourse, and (**D**)essert. The domains of the variables are as follows:

A: (**v**)eggies, (**e**)scargot

B: (**w**)ater, (**s**)oda, (**m**)ilk

C: (**f**)ish, (**b**)eef, (p)asta

D: (**a**)pple pie, (**i**)ce cream, c(**h**)eese

The menu has to obey the following dietary constraints:

(i) Vegetarian options: The appetizer must be veggies or the main course must be either pasta or fish.

(ii) Total budget: If you serve the escargot, you cannot afford any beverage other than water.

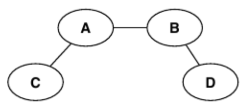
(iii) Calcium requirement: You must serve at least one of milk, ice cream, or cheese.

(2.a) Draw the constraint graph over the variables A, B, C and D.

(i) A=v or C=p or C=f

(ii) A=e => B=w

(iii) B=m or D=i or D=h



(2 marks)

(2.b) Assume we first assign *A=e*. List the domains of the variables after forward checking.

A in { e }

B in { w ~~s m~~ }

C in { f ~~b~~ p }

D in { a i h }

(3 marks)

(2.c) Again assume we first assign *A=e*. List the domains of the variables after arc consistency has been enforced.

A in { e }

B in { w ~~s m~~ }

C in { f ~~b~~ p }

D in { ~~a~~ i h } based on constraint (iii)

(3 marks)

(2.d) Give a solution for this CSP or state that none exists.

Multiple solutions exist.

('e', 'w', 'f', 'i'), ('e', 'w', 'f', 'h'), ('e', 'w', 'p', 'i'), ('e', 'w', 'p', 'h')

(1 mark)

(2.e) What is the running-time of an efficient solver for tree-structured constraint satisfaction problems given that n is the number of variables and d is the maximum size of any variable's domain?

O(n*d*d)

(3 marks)

(2.f) Name and briefly describe a standard technique for turning nearly tree-structured CSPs into tree-structured ones.

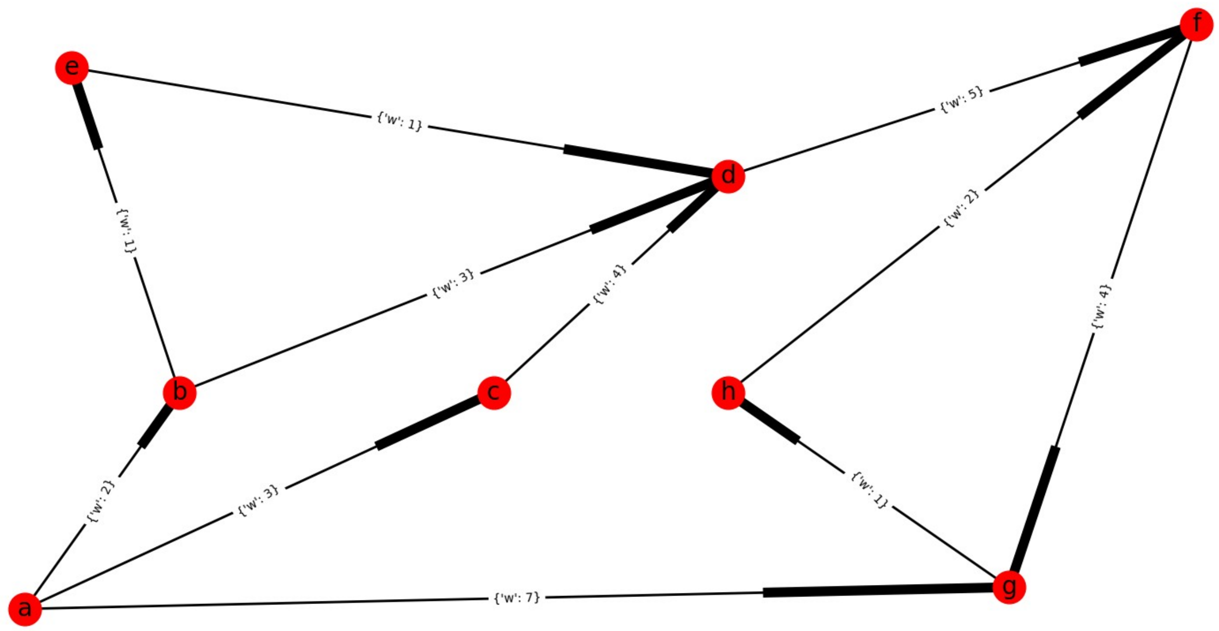
cutset conditioning.

One standard technique is to instantiate cutset, a variable (or set of variables) whose removal turns the problem into a tree structured CSP. To instantiate the cutset you set its values in each possible way, prune neighbors, then solve the reduced tree structured problem (which is fast).

(3 marks)

Question 3

Consider the state graph below whose vertices represent states and whose arcs represent transitions between states. The head of the arc is indicated by the wider segment of the edge. For example, the state 'a' connects to state 'c with an arc directed from 'a' to 'c. The cost of using this arc is the weight 'w=3'. We assume that the **successors of a state** are **generated in alphabetical order**.



(3.a) How many nodes are in the **complete search tree** for the graph above when the initial state is **A**? Note that no goal state has been specified yet.

Infinite search tree because of cycle f-g-h

(2 marks)

(3.b) Recall that a subgraph induced by a subset *S* of vertices of a graph *G* is the graph obtained from *G* by removing all vertices not in *S* and their incident arcs. How many nodes are in the **complete search tree** for the subgraph induced by the subset of vertices {A, B, C, D, E} when the initial state is **A**?

Seven nodes

(2 marks)

(3.c) Consider a depth-first search on the state graph displayed at the beginning of Question 3. Assume that the initial state is **A**, and that **D** and **F** are goal states. Enter in the box below the final path returned by a **depth-first tree search**. Your answer should be 'no solution' or a string with 'A' as the first character and the last character should denote one of the goal states.

A, G,H,F

(4 marks)

(3.d) Consider a breadth-first search on the state graph displayed at the beginning of Question 3. Assume that the initial state is **F**, and that **E** and **D** are goal states. Enter in the box below the final path returned by a **breadth-first tree search**. Your answer should be 'no solution' or a string with 'F' as the first character and the last character should denote one of the goal states.

No solution

(4 marks)

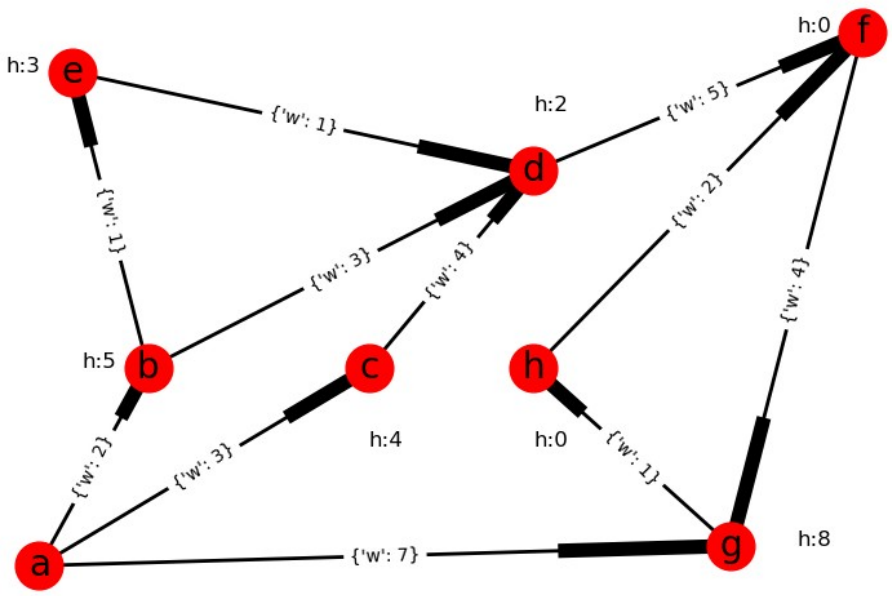
(3.e) Consider a uniform cost search on the state graph displayed at the beginning of Question 3. Assume that the initial state is **A**, and that **F** is the goal state. Enter in the box below the final path returned by a **uniform cost tree search**.

A,B,E,D,F for a cost of 9

(5 marks)

Question 4

Consider the search space below, where **A** is the start state and **F** and **H** are the only states that satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost *h* (heuristic) to a closest goal is indicated next to the vertices.



For each of the following **tree search** strategies, indicate which goal state is reached (if any) and list, in order, all the states popped out of the frontier. Assume that the **successors of a state are generated in alphabetical order**. In case of a tie, pick the element that comes first in the alphabetical order.

Iterative deepening depth first search

Goal state reached: **H**

States popped out of the frontier:
rec function implementation A | A,B,C,G | A,B,D,E,C,D,G,H

non rec implementation **A . A G C B . A G H**
Solution path : A, G, H

A* tree search

Goal state reached: **F** [with a cost of 9, note heuristic is not admissible, so no surprise that the returned path is not optimal]

States popped off of the frontier: **A,B,E,D,C,D,D,F** *(final path ABEDF)*
details next page

(4+5 marks)

Frontier {**A**}

pop **A**, push B(7=2+5), C(7=3+4), G(15=7+8)

Frontier {**B(7=2+5)**, C(7=3+4), G(15=7+8)}

pop **B(7=2+5)**, push D(7=5+2), E(6=3+3)

Frontier {**E(6=3+3)**, C(7=3+4), D(7=5+2), G(15=7+8)}

pop **E(6=3+3)**, push D(6=4+2)

Frontier {**D(6=4+2)**, C(7=3+4), D(7=5+2), G(15=7+8)}

pop **D(6=4+2),** push F(9=9+0)

Frontier {**C(7=3+4)**, D(7=5+2), F(9=9+0), G(15=7+8)}

pop **C(7=3+4)**, push D(9=7+2)

Frontier {**D(7=5+2)**, D(9=7+2), F(9=9+0), G(15=7+8)}

pop **D(7=5+2)**, push F(10=10+0)

Frontier {**D(9=7+2)**, F(9=9+0), F(10=10+0), G(15=7+8)}

pop **D(9=7+2)**, push F(12=12+0)

Frontier {**F(9=9+0)**, F(10=10+0), F(12=12+0), G(15=7+8)}

pop **F(9=9+0), goal!**

Sequence popped ABEDCDDF

Question 5

We write $h^*(s)$ for the true cost from a state s to the closest goal state of a state graph *G1*. We write *g1(n)* for the cost of a path from the root node to node *n* in the search tree associated with *G1*. Let *h1* be an admissible heuristic, and consider *h2 = 7*h1*. Consider the state graph *G2* derived from *G1* by multiplying the weight of each arc by 7.

(5.a) Do the optimal paths in *G2* correspond to the optimal paths in *G1*? Justify your answer. Provide a counter-example if needed.

Yes a path p2 is optimal in G2 if and only if its corresponding path p1 is optimal in G1 as the cost of p2 is 7 times the cost of p1.

(2 marks)

(5.b) Is *h2* an **admissible** heuristic for *G2*? Justify your answer.

Yes, h2 is admissible in G2 because h2*(n) = 7*h1*(n) >= 7*h1(n) = h2(n)

(3 marks)

(5.c) What can you say about the cost of the returned path when running A* tree search on *G1* with *h2* as the heuristic? Relate the cost of the returned path to the cost of an optimal path.

The value of f(n) = g1(n)+h2(n) is less of equal to 7*g1(n)+h2(n) = g2(n)+h2(n)

Let G be first goal node G popped from the frontier.

Let G' an optimal goal node and n' an ancestor of G' in the frontier. Then,

f(G) <= f(n') because G was popped from the frontier before n'.

We have f(n')=g1(n')+h2(n') <= 7* (g1(n')+h1(n'))

We have also g1(n')+h1(n') <= g1(G') because h1 is admissible.

Therefore g1(G)= f(G) <= f(n') <= 7*g1(G')

That is, the returned solution will be in the worst case 7 times more expensive than an optimal solution.

(4 marks)

End of Paper

Scribble Pages [will be ignored by the marker]