CS60045 Artificial Intelligence, Autumn 2023–2024

Class Test 1

31–August–2023 06:30pm–07:30pm Maximum marks: 20

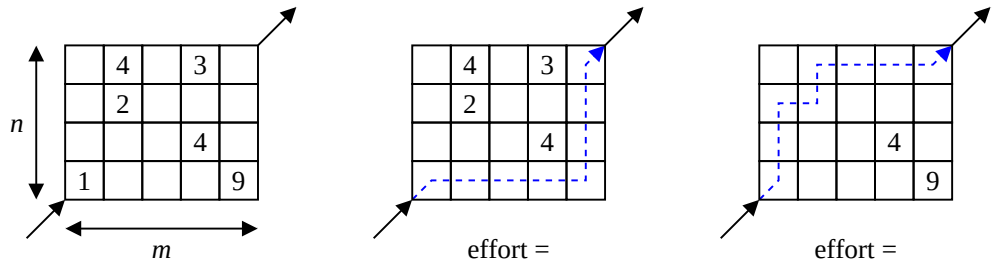
Roll no: _____ **Name:** _____

[*Write your answers in the question paper itself. Be brief and precise. Answer all questions.*]

1. [State-Space Formulation]

A robot carrying an empty container enters an $m \times n$ mesh. Some cells in the mesh contain integral amounts of mercury. The robot enters the mesh at the bottom-left corner and leaves the mesh at the top-right corner. The robot is allowed to move only in the right or the up direction. Its main objective is to collect as much mercury as possible. If it enters a cell, it collects the entire mercury (if any) stored in that cell. If it carries W amount of mercury while moving from one cell to an adjacent one, its effort for that step is W. The secondary objective of the robot is to minimize its total effort (the sum of the efforts of all the steps it takes).

The following figure gives an example on a 5×4 mesh. A cell with a number indicates the amount of mercury stored in that cell. Empty cells contain no mercury. Two paths of collecting the maximum amount of mercury are also shown in the figure. After the mercury is taken from a cell, the cell is shown as blank. These two paths give the robot the same amount of mercury, but the first leads to a total effort of 34, whereas the second to a total effort of 30. So the second path is preferable. Assume that cell numbering is 1-based. Also assume for simplicity that the (m,n)-th cell does not contain any mercury.



effort = 1 + 1 + 1 + 1 + 10 + 10 + 10 = 34

effort = 1 + 1 + 1 + 3 + 7 + 7 + 10 = 30

In this exercise, you are not asked to solve the search problem. Your task is instead to formulate the problem as a state-space search problem. You are also not needed to assign costs to the transitions or the states. Answer only the following parts.

(a) What should each state consist of? Mention all the necessary items that each state should store. Do not include any irrelevant items. **(2)**

Solution (1) The position (x,y) of the robot with $1 \leqslant x \leqslant m$ and $1 \leqslant y \leqslant n$.
(2) The amount of mercury still existing in each cell (0 or the initial amount) (an $m \times n$ array of integers).
(3) The amount of mercury the robot is carrying.

Note that maintaining both (2) and (3) seems unnecessary. The current best path to a node can be obtained by following the parent pointers. However, maintaining both saves some computation during node expansion.

But did we ever mention that the search is going to maintain a tree with parent pointers?

(b) What is the start state? **(2)**

Solution (1) The robot's initial position $(1,1)$.
(2) The original mercury contents of all the cells.
(3) Initial amount of zero.

(c) What are the goal states? **(2)**

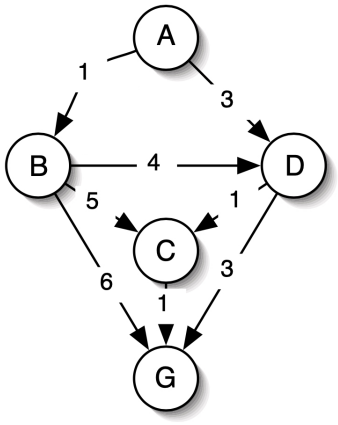
Solution All configurations consisting of
(1) The robot's final position (m,n).
(2) The $m \times n$ mesh with each cell containing either 0 or the initial amount of mercury.
(3) Any non-negative integer indicating the total amount of mercury collected.

(d) How will a transition look like? **(2)**

Solution Take a state $((x,y),M,W)$. Let μ be the amount of mercury at the (i,j)-th cell of M (we may have $\mu = 0$). If $x < m$, then we have a transition to $((x+1,y),M',W+\mu)$, where M' is obtained from M by making its (i,j)-th cell 0 (if it was not that already). In addition, if $y < n$, then we have a transition to $((x,y+1),M',W+\mu)$.

2. [Heuristic Search]

Consider the state-space graph shown in the adjacent figure. Here, A is the start node, and G is the only goal node. We can search from the start node to the goal node by using a variety of different algorithms, resulting in different search/exploration trees. In this exercise, you explore the use of the *A* algorithm* under the following two heuristics.



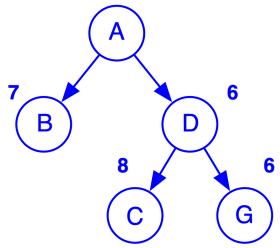
[Heuristic 1] $H_1 : \{\ h(A) = 3,\ h(B) = 6,\ h(C) = 4,\ h(D) = 3\ \}$

[Heuristic 2] $H_2 : \{\ h(A) = 3,\ h(B) = 3,\ h(C) = 0,\ h(D) = 2\ \}$

In connection with this search, answer the following parts.

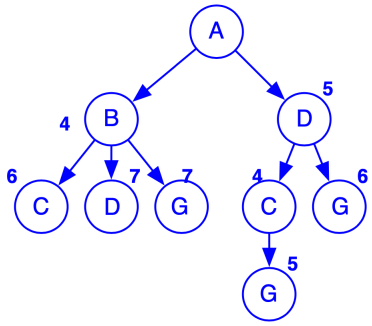
(a) Draw the exploration tree generated by the A* algorithm under the H_1 heuristic. If a node is generated multiple times, show all the instances. Against each instance n, write the the value of $f(n) = g(n) + h(n)$ at the time of the generation of that instance. **(2)**

Solution



(b) Repeat Part (a) (draw the exploration tree along with the $f(n)$ values), but using the H_2 heuristic. **(2)**

Solution



(c) Did the A* algorithm under the H_1 heuristic find the least-cost path? Give proper justifications. **(2)**

Solution **No.** A* is guaranteed to find an optimal path when the heuristic is admissible. In H_1, the heuristic value for C (that is, $h(C)$) is not an underestimate of the optimal cost to the goal.

(d) Did the A* algorithm under the H_2 heuristic find the least-cost path? Give proper justifications. **(2)**

Solution **Yes.** A* is guaranteed to find an optimal path when the heuristic is admissible. In H_2, the heuristic values of all nodes are admissible, so the optimal path was found,

Note that H_2 is not a consistent heuristic, since the link from D to C decreases the heuristic cost by 2, which is greater than the link cost of 1. Anyway, consistency is not required for optimal paths. Admissibility suffices.

3. [AND-OR Graphs]

Your task is to represent a positive integer n as a sum of $5, 7, 11$ (each may appear multiple times). For example, $21 = 11+5+5 = 7+7+7$ are two representations of 21. You pose this as an AND-OR search problem. Let m be a number that we want to represent as a $5, 7, 11$ sum ($m = n$ at the start state). We decompose this problem as follows. We write $m = a+b$ for positive integers a and b, and search whether both a and b can be represented as $5, 7, 11$ sums. In order to avoid duplication of effort, we force $a \geqslant b$. Moreover, since 5 is the smallest of the given summands, we should have $b \geqslant 5$. A number may have many such decompositions, like $100 = 95+5 = 94+6 = 93+7 = 92+8 = \cdots = 50+50$. The decompositions like $5+95$, $49+51$, $96+4$ and $99+1$ are not to be explored. Each state in the AND-OR graph is specified by a positive integer m (that we want to decompose as a $5, 7, 11$ sum). Note that:

(1) 5, 7, and 11 are the only **terminal nodes**.

(2) Any state $m \neq 5, 7, 11$, for which any decomposition $m = a+b$ with $a \geqslant b$ implies $b < 5$, is a **dead end** (Example: 8). In other words, your graph must not contain any node $m < 5$.

Draw the *complete* AND-OR search graph for $n = 17$. Mark all the terminal nodes and the dead ends in your graph. Moreover, for each $m \leqslant n$ (if it appears in the search graph), there must be a unique node for m. Indicate all the AND arcs appropriately. You do not have to specify the cost against any transition or node. You do not have to run the AO* algorithm on this graph. **(4)**

Solution