CPT_S 540 HW2

Yang Zhang

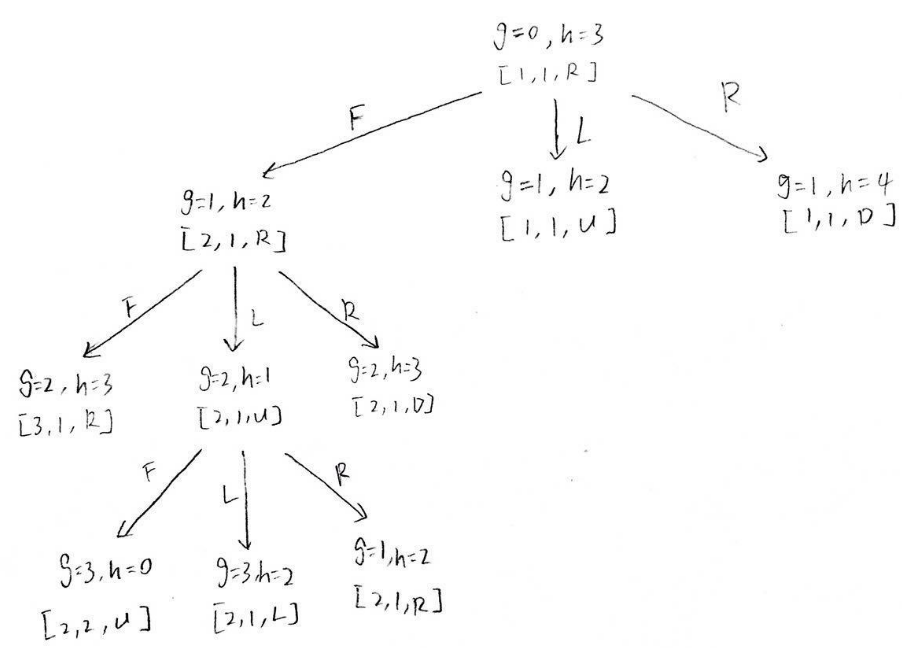
11529139 (graduate)

1. Recall the Wumpus World search task. Consider a simplified search-based approach for moving our Wumpus World agent using the state representation [*x, y, d*], where *x, y* is the location of the agent, and *d* is the direction {U=up, D=down, L=left, R=right} that the agent is facing. The agent has three actions: F=GoForward, L=TurnLeft, and R=TurnRight. The initial state is [1, 1, R].

a. Show the search tree generated by A* search, where the goal test is [2, 2, U], and actions are always tried in the order F, L, R. For each node, show the value of the evaluation function *f(n)=g(n)+h(n)*. The heuristic function is given by

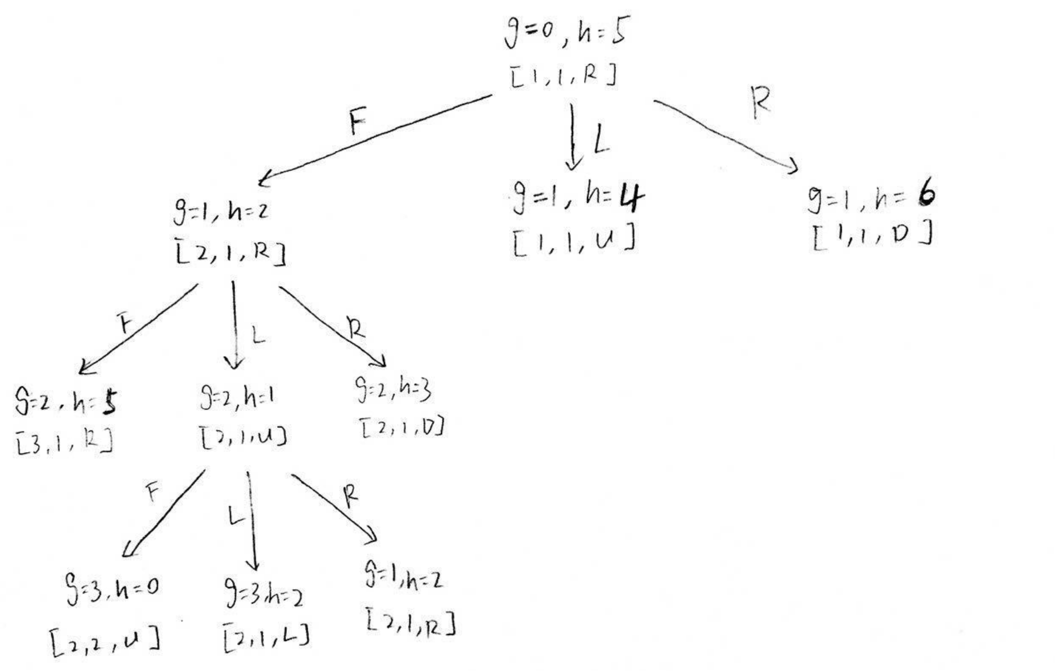
h(n) = (city-block-distance + discrete-angular-distance)

For example, the initial state has a city-block-distance of 2 from the goal state, and a discrete-angular-distance of 1 (requires 1 TurnLeft to be in the up orientation); thus, the heuristic value of the initial state is 2 + 1 = 3. Note that discrete-angular-distance is always one of {0, 1, 2}.



b. Repeat part (a), but with a different heuristic function

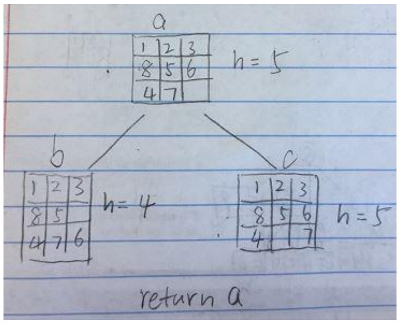
$h(n)$ = (city-block-distance)^2 + (discrete-angular-distance)

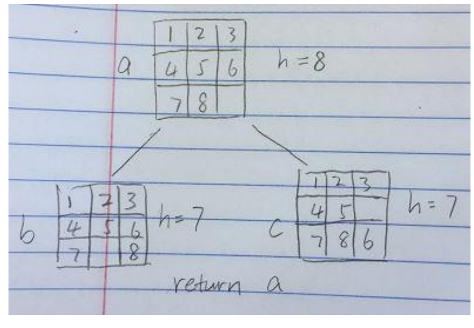


c. **Is the heuristic function from part (b) admissible? Justify your answer.**
It is not admissible, because it over-estimates the cost of reaching goal state from current state.

2. Suppose we want to use the Hill-Climbing search algorithm to solve the 8-puzzle problem using the "Number of Tiles Correct" heuristic (which equals 8 – "Number of Tiles Out").

a. **Draw the search tree generated for the initial state and goal state in Figure 1a. Show the value of the heuristic next to each node. Indicate which state the algorithm will return.**





1	2	3
8	5	6
4	7	

Initial State

1	2	3
4	5	6
7	8	

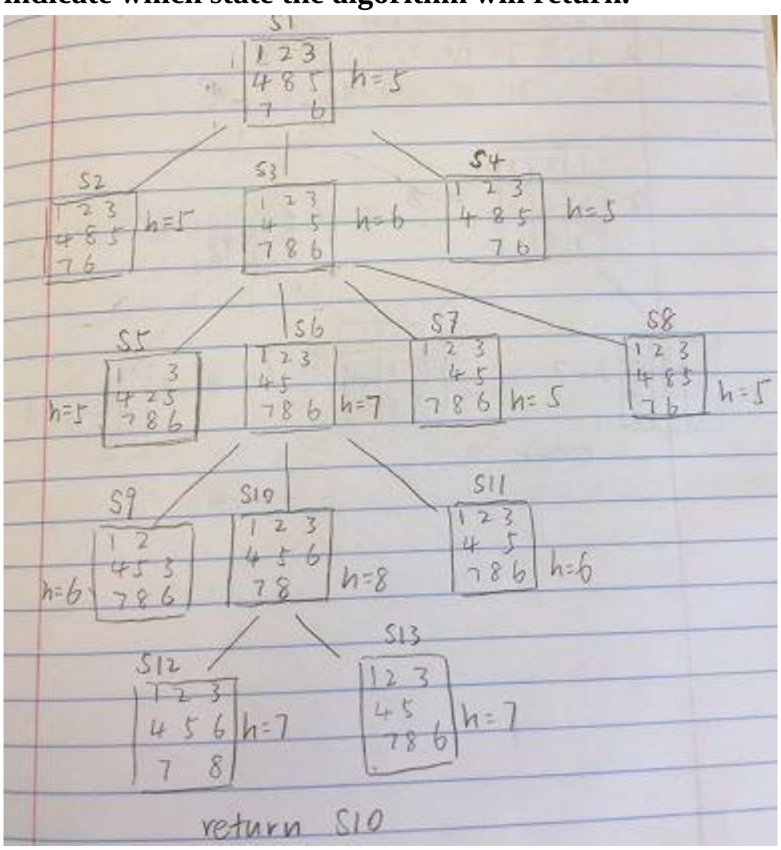
Goal State

Figure (1a)

1	2	3
4	8	5
7		6

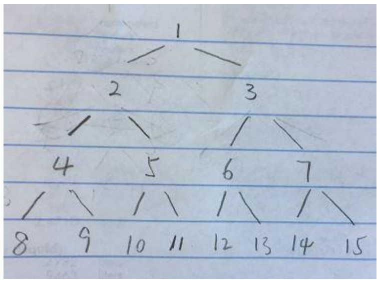
Figure (1b)

b. Consider a different initial state (shown below in Figure 1b). Repeat part (a) for this new problem. Again, show the value of the heuristic next to each node and indicate which state the algorithm will return.



3. Consider a state space where the states are described by positive integers *i*. The initial state is i = 1 and the successor function for state *i* returns two states *2i* and *2i + 1*.

a. Draw the portion of the search tree that contains the initial state and all states that can be reached from the initial state down to depth 3.



b. Let the goal state *Goal* = 11. List the order of nodes that are expanded for breadth-first search, depth-limited search to depth 3, and iterative deepening DFS.

BFS: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}
DLS: {1, 2, 3, 6, 7, 14, 15, 12, 13, 4, 5, 10, 11}
Id-DFS: {1, 2, 3, 6, 7, 4, 5, 14, 15, 12, 13, 10, 11}

c. Suppose we call the action of going from state *i* to state *2i* LEFT while the action of going from state *i* to state *2i + 1* RIGHT. Devise an algorithm that outputs the solution to this problem without any search at all; write your algorithm either in crisp, brief English prose or as a legible pseudo-code with brief explanations. (Here, you are given the goal state in the form *Goal* = *n*, where *n* is a positive integer.)

```
Func : input integer n > 1
    Path_stack := {}
    While n > 1
            If(n%2 == 0)
                    Path_stack.push(LEFT)
            Else
                    Path_stack.push(RIGHT)
            EndIf
            n /= 2;
    EndWhile
    Return Path_stack
EndFunc
```

4. a) Given the directed graph in **Figure 2** below representing a state space (where the nodes are the states and the numerical values associated with the edges are the step costs), show the search tree that will be generated by *uniform-cost search*. Assume that *S* is the initial state and *G* is the goal state. Additionally, list the sequence of nodes expanded and their associated path cost *g(n)* in parenthesis.

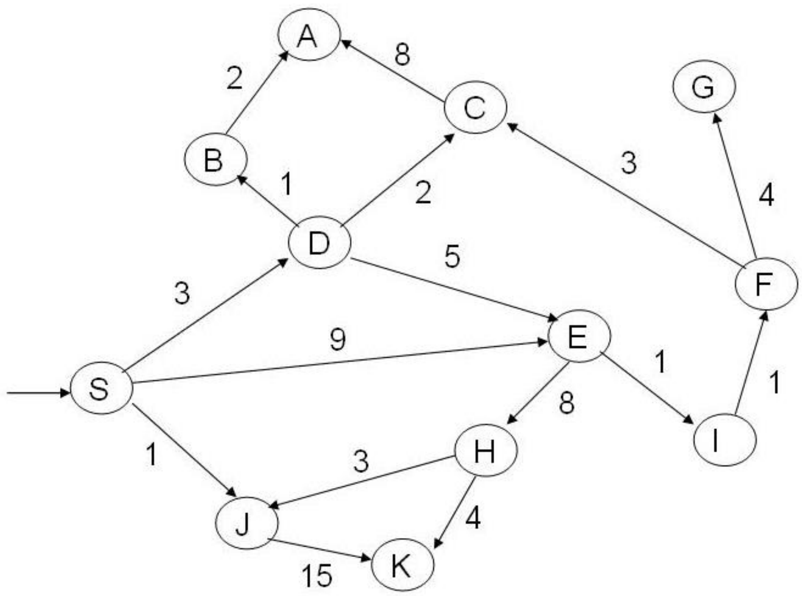
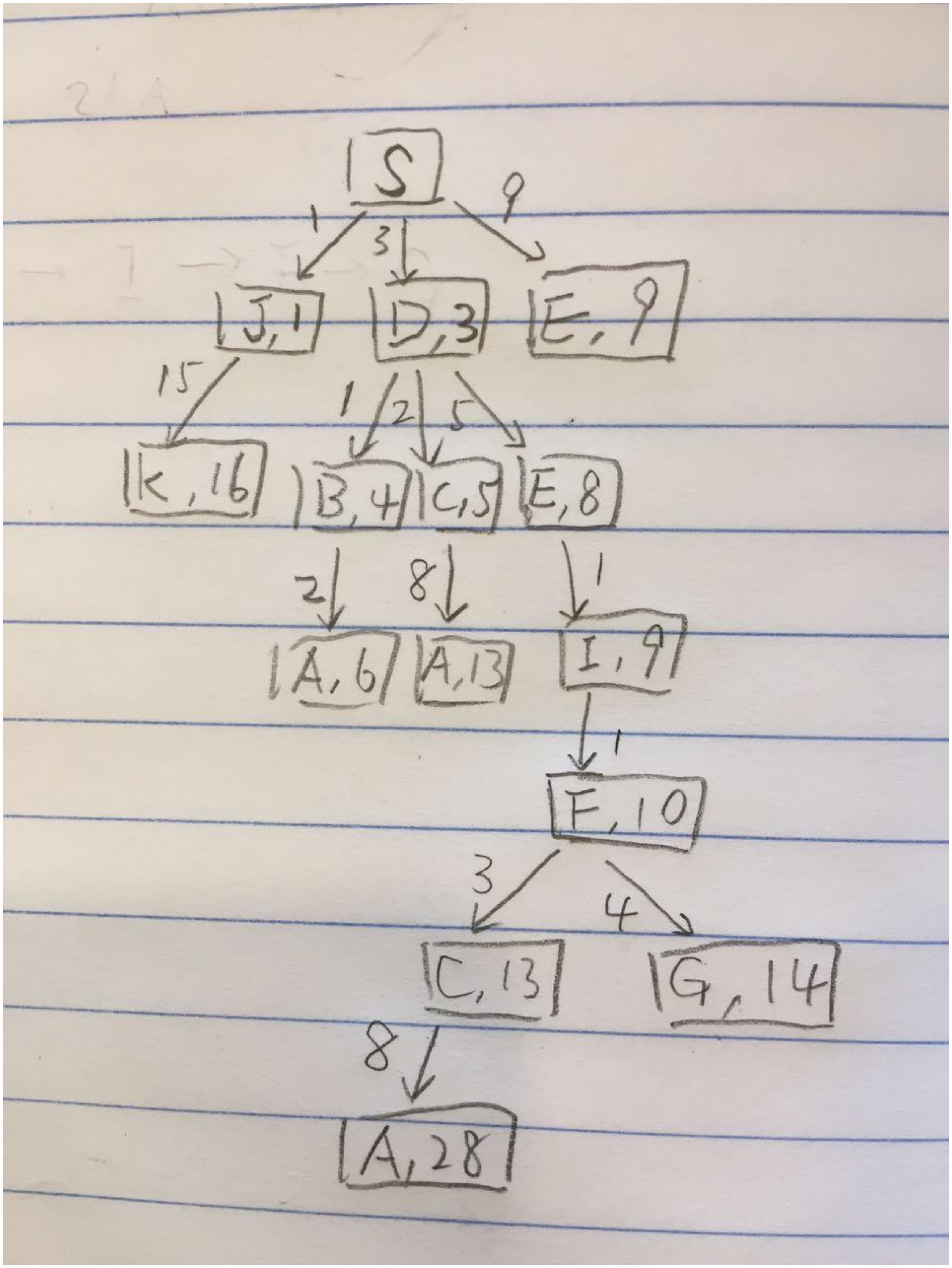


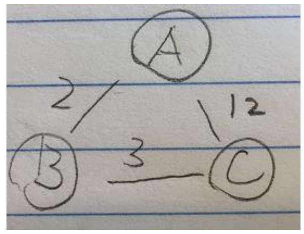
Figure 2 (for Problem 4a)

b) In uniform-cost search, why is it important to apply the goal test to a node when it is selected for expansion rather than when it is originally generated? Please explain briefly (in 1-2) sentences and give an example to illustrate your answer.

(a)


S
1
3
9
J,1
D,3
E,9
15
1
2
5
K,16
B,4
C,5
E,8
2
8
1
A,6
A,13
I,9
1
F,10
3
4
C,13
G,14
8
A,28

(b) That is because test the expanding node will grantee the current path has the min f score, otherwise, it may choose the larger cost path. For example, a simple graph



If test goal state when it is originally generated, we get the answer A->C with total cost 12, which is not good as A->B->C with total cost 5.

5. Explain in 1-2 concise, to-the-point sentences each of the following statements:
(Part (a) is worth 2 points; all other parts are worth 3 points each)

a. Breadth-first search is a special case of uniform-cost search.
BFS is special case of UCS that the cost of every edge is identical.

b. Breadth-first search is a special case of best-first search.
BFS is a special case of best-first search when f score is identical for every node

c. Depth-first search is a special case of best-first search.
DFS is a special case of best-first search when the 1^{st} child always has the best heuristic value.

d. Uniform-cost search is a special case of A* search.
UFC is a special case of A* search when f(n) = g(n)

e. Greedy best-first search is a special case of A* search.
Greedy best-first search is a special case of A* search when f(n) = h(n)

f. Hill climbing is a special case of beam search.
Hill climbing is a special case of beam search when beam width =1

g. Depth-first search is a special case of depth-limited search.
Depth-first search is a special case of depth-limited search when the goal state is within the depth limit.