Quiz Submissions - ECE457 A Midterm

Zahin Mohammad (username: z5mohamm)

Attempt 2

Written: Jun 25, 2020 7:03 PM - Jun 26, 2020 10:42 AM

Submission View

Released: Jul 10, 2020 4:49 PM

Question 1 **3 / 3 points**

Iterative deepening search is preferred over breadth-first search for its space complexity advantage

- ✓ (●) True
- ( ) False

Question 2 **3 / 3 points**

No cost and no heuristic information are given. You need to find the shallowest solution (path from initial state to goal state), you want to optimize time, and you have plenty of memory/space to use. Select the correct statements about selecting search algorithms to be used.

- ✗ [ ] BFS requires less time than Iterative Deepening.
- ✗ [✓] Depth-Limited if we know the depth.
- ✓ [✓] BFS, when there no limits for space usage.
- ✓ [ ] DFS is the best choice to be used to meet these requirements.

Question 3 **3 / 3 points**

An adaptive genetic algorithm that uses the Rechenberg 1/5 success rule has determined that it had produced 100 mutations, 23 of which can be considered improving (produced better individuals), thus it (the algorithm) will

- ✓ (●) increase the mutation size
- ( ) keep the mutation size unchanged
- ( ) reduce the mutation size

Question 4 **1.5 / 3 points**

A state graph with no link costs or heuristic values. The algorithm should find paths to a goal with the least number of states and the space requirements should depend on the depth of the first goal found. What is the best search algorithm to be used?

- ✗ [✓] BFS.
- ✓ [ ] DFS.
- ➡ ✓ [✓] Iterative deepening.
- ✗ [✓] Uniform Cost Search.

Question 5 **3 / 3 points**

Consider the traveling salesman problem. If the initial solution is [1,3,2,5,4,1], which of the followings can not be a neighboring solution?

- ( ) [1,3,5,2,4,1]
- ( ) [1,5,2,3,4,1]
- ✓ (●) [1,3,2,5,1,4]
- ( ) [3,1,2,5,4,3]

Question 6 **3 / 3 points**

A Self-adaptive GA achieves mutation adaptation by

- ( ) taking feedback from the current state of the search and heuristically controls the mutation parameter values
- ✓ (●) incorporating the mutation parameter values into the chromosomes
- ( ) adapting the mutation parameter as a function of time

Question 7 **1 / 3 points**

A tree search space with a fixed depth and all the goals at the leaves of the tree. What are the correct statements below to find any goal as quickly as possible and heuristic is provided.

- ➡ ✗ [ ] Uniform cost search will find the optimal solution.
- ✓ [ ] We must use a strategy to avoid repeated states to reduce the required states.
- ➡ ✓ [✓] Greedy best first search is the best choice as a search algorithm.
- ✓ [ ] Iterative deepening is the best choice as a search algorithm.

Question 8 **3 / 3 points**

Which one of the following situations represents an example of a dynamic environment?

- ( ) Paying chess
- ( ) Playing tic-tac-toe (X-O)
- ✓ (•) An autonomous car on the road
- ( ) A robot finding a stationary goal in a maze

Question 9 **3 / 3 points**

Which one of the followings is an example of a well-structured problem?

- ( ) Identifying objects in a picture
- ( ) Clustering (categorizing) a set of pictures
- ✓ (•) Classifying pictures to colored and grey scaled
- ( ) designing a picture classifier

Question 10 **3 / 3 points**

BFS is preferable over DFS, when:

- ✓ [✓] A shallow solution (path from initial state to goal state) is preferred.
- ✗ [ ] The search tree contains large or possibly infinite branches.
- ✓ [✓] Abundant memory space to store the search tree (or the queue) is available.

Question 11 **4 / 4 points**

When do we use an approximate algorithm rather than an exact algorithm?

Answer: Approximate algorithm's are used when near-optimal solutions can be accepted and can be found quickly. ✗ **(When finding the optimal solution has high computational cost)**

Question 12 **4 / 4 points**

Why is the Beam Search not complete and not optimal?

Answer: Beam search is not complete therefore it cannot be optimal. It is not complete because it only expands beta first promising children, whereas the goal node may lay in further then the first beta promising nodes of a level. ✗ **(Consider the case that all the paths that Beam Search follows do not contain a solution.)**

Question 13 **4 / 4 points**

Why do we use gray coding in genetic algorithms?

Answer: Gray coding makes it easier to bit-flip the phenotype of an individual by simply increasing the Gray code number. This due to the fact that the hamming distance between Gray codes is 1. ✗ **(To get proportional genotype-phenotype mapping)**

Question 14 **4 / 4 points**

What element makes a search algorithm "informed"?

Answer: Domain specific knowledge encoded into a heuristic function used in the search algorithm makes an algorithm informed. ✗ **(Heuristic function)**

Question 15 **4 / 4 points**

Why should we care about space complexity of the problems?

Answer: We should care about space complexity because the algorithms are implemented on real computers that have finite memory. Many algorithms have large search spaces therefore it may be impossible to store the entire search space in memory. ✗ **(Because of the memory constraints in our computational machines)**

Question 16 **2 / 2 points**

SA is hard to implement due to a large number of required iterations.

- ( ) True
- ✓ (•) False

Question 17 **2 / 2 points**

Meta-heuristic algorithms can only be applied to discrete problems

- ( ) True
- ✓ (•) False

Question 18 **2 / 2 points**

The mutation operator used in GA is used to enable exploration of the search space

- ( ) True
- ✓ (•) False

Question 19 **2 / 2 points**

beam search is complete if Beta is finite

- ( ) True
- ✓ (•) False

Question 20 **2 / 2 points**

A state with heuristic function value equal to zero, means that the optimal path must pass this state.

- ○ True
- ✓ ◉ False

Question 21 **2 / 2 points**

Iterative-deepening search always finds the optimal path to the goal.

- ○ True
- ✓ ◉ False

Question 22 **2 / 2 points**

If all moves are unit cost, DFS always finds the optimal path to the goal.

- ○ True
- ✓ ◉ False

Question 23 **2 / 2 points**

Opposite to SA, TS can be applied to continuous optimization problems.

- ○ True
- ✓ ◉ False

Question 24 **2 / 2 points**

If we have two admissible heuristics, A* will search states in different orders for these two heuristics.

- ✓ ◉ True
- ○ False

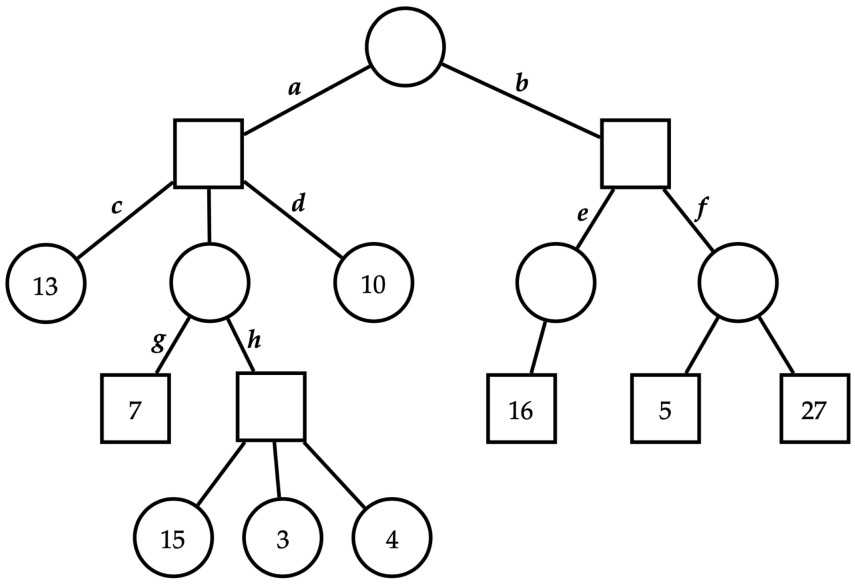
Question 25 **0 / 2 points**

SA allows only a certain number of hill climbing.

- ✗ ◉ True
- ➡ ○ False

Question 26 **6 / 6 points**

Assume the following zero-sum game tree,
where ◯ are the minimising nodes, and □ are the maximising nodes:



Suppose you had used alpha-beta pruning, which branches would have been cut off from the game tree?

- ○ e,f
- ○ h,g
- ○ d,h
- ✓ ◉ h,f

Question 27 **6 / 6 points**

Consider the two parents:

Parent A 4 8 7 | 3 6 5 | 1 10 9 2

Parent B 3 1 4 | 2 7 9 | 10 8 6 5

If the PMX crossover is initiated first at the three middle genes of parent A (i.e., 3,6,5). The crossover will produce the following offsprings

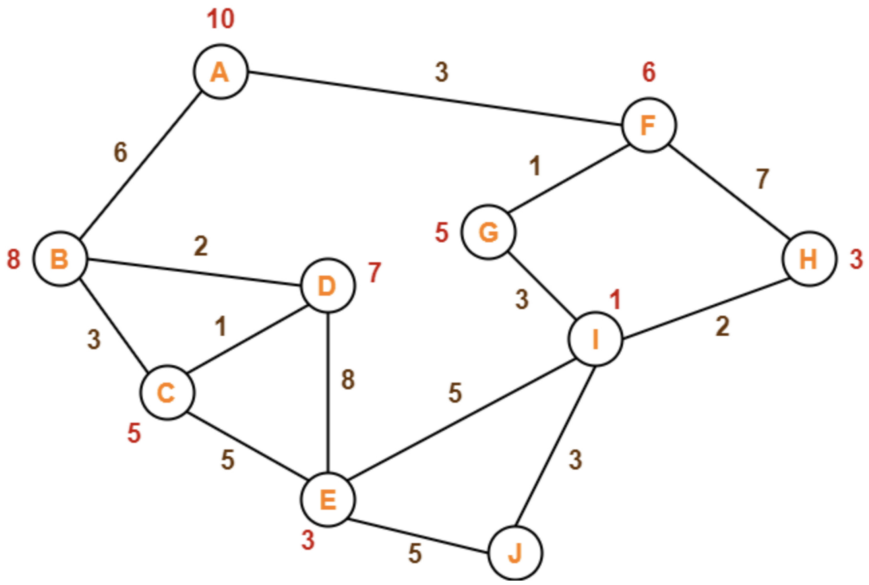
- ○ Child A 3 8 6 2 7 9 1 10 5 4

  Child B 9 1 4 3 6 5 10 8 7 2

✓ Child A 4 8 6 2 7 9 1 10 5 3
Child B 2 1 4 3 6 5 10 8 7 9

Child A 4 6 8 2 7 9 1 10 5 3
Child B 2 4 1 3 6 5 10 8 7 9

Question 28 **6 / 6 points**



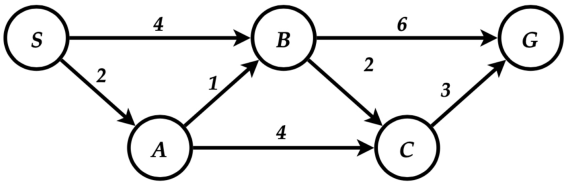
For the above search tree, the A* algorithm will determine the following order of nodes as the optimal path to the state J:

✓ A-F-G-I-J

A-F-B-G-I-D--J

A-F-B-D-E-J

Question 29 **0 / 6 points**



	h_1	h_2	h_3
S	8	7	3
A	3	4	0
B	7	5	4
C	2	2	0
G	0	0	0

For the above search problem, S is the start node and G is the goal node. Three different heuristic functions are defined, h1, h2 and h3 are considered.

What solutions will be returned by breadth-first tree search, and A* tree search using the h3 heuristics?

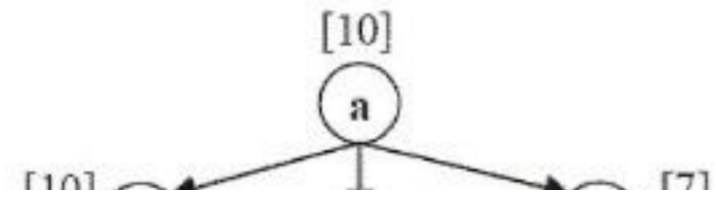
✗ BFS:S A C G with cost 9, A*:S A C G with cost 9

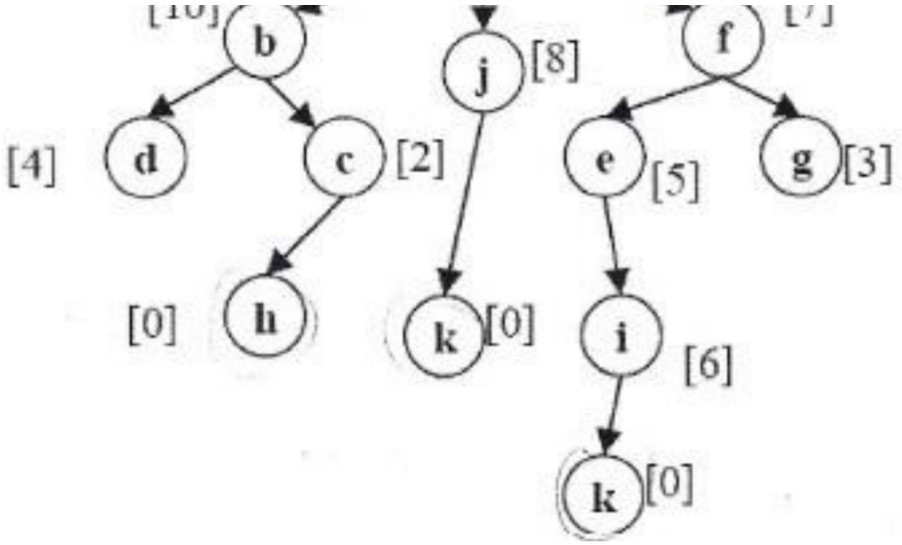
➡ BFS: S B G with cost 10, A*:S A B C G with cost 8

BFS: S B G with cost 10, A*:S A C G with cost 9

Question 30 **6 / 6 points**

if we use 2-Beam Search to find the goal state k, then the search will progress in the following order





- a-f-j-g-e-i-k
- a-f-c-g-e-j-k
- ✓ a-f-j-g-e-k

Attempt Score:88.5 / 100 - 88.5 %
Overall Grade (highest attempt)**:**88.5 / 100 - 88.5 %

Done