**CSE112 Artificial Intelligence**，**Week 4 2019**

**Exercises and Tutorial Questions**

# Review questions about intelligent agent:

1. In terms of an agent, what does it mean to say that its environment is accessible? inaccessible?

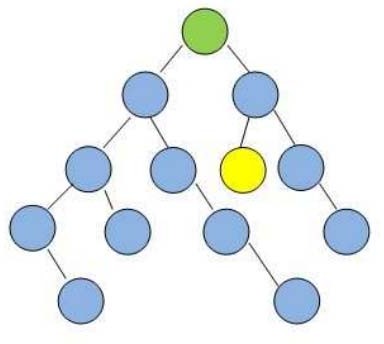
If there is enough information to search.

1. What is a successor function?

Primitive recursive function

# Review questions about BFS and DFS:

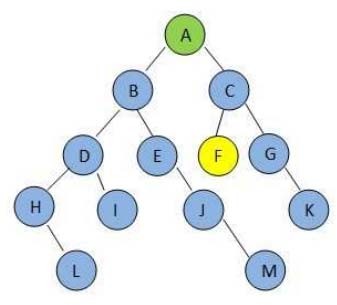
1. In the following graphs, assume that if there is ever a choice amongst multiple nodes, both the BFS and DFS algorithms will choose the left-most node first



Q1. Starting from the green node at the top, which algorithm will visit the least number of nodes before visiting the yellow goal node? A

A: BFS B: DFS

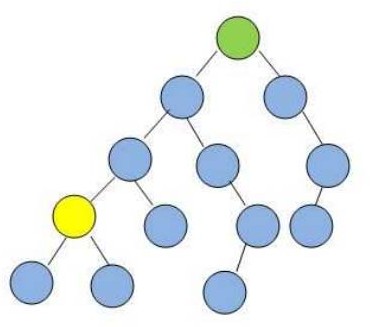
C: Neither BFS nor DFS will ever encounter the goal node in this graph.

D: BFS and DFS encounter same number of nodes before encounter the goal node Q2. How can we get ?

BFS: ABCEDF

DFS: ABDHLIEJMCF

1. In the following graphs, assume that if there is ever a choice amongst multiple nodes, both the BFS and DFS algorithms will choose the left-most node first.



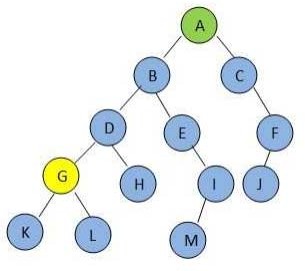
Q1. Starting from the green node at the top, which algorithm will visit the least number of nodes before visiting the yellow goal node? B

A: BFS B: DFS

C: Neither BFS nor DFS will ever encounter the goal node in this graph.

D: BFS and DFS encounter same number of nodes before encounter the goal node

Q2. How can we get ?



DFS: ABDG

BFS: ABCDEFG

**Question 1.** State whether each of the following statements is true or false.

1. Depth-first search with iterative deepening uses less storage than breadth-first search, but runs asymptotically as fast. F
2. Uniform-cost search is a special case of Breadth- first search. F
3. A\* is a special case of uniform-cost search. F
4. Breadth-First search is a complete algorithm even if all operators do not have the same cost. T
5. Breadth-first search can be thought of as a special case of A\*. Briefly explain your answer. T

Always minimize the overall cost

1. Breadth-first search can be thought of as a special case of uniform cost search. Briefly explain your answer. T

When the cost is equal, they are the same.

**Question 2.** Answer briefly the following questions. Please use your own words, and cite appropriately.

1. What is Iterative Deepening Search(IDS) and how does it works?

Formed from DFS, increase the depth limit per round.

1. When we say a particular search strategy is complete, what do we mean?

Find all nodes to the end.

1. Under what conditions is breadth-first search complete? Optimal?

When the depth is finite, it is complete, not optimal in general.

1. What is the time and space complexity of iterative-deepening search?

When the number of nodes is infinite, it is complete and optimal.

1. What is the difference between greedy and A\* search?

A\* is past cost + remaining cost, greedy is just remaining cost.

1. Under what conditions does A\* search produce the optimal solution?

When the node has finite number of children.

1. What does admissible heuristic mean for A\* search?

Utilizing the information to do search in order to find the solution quickly.

**Question 3.** When comparing tree-search algorithms, we measure the number of nodes expanded. How many nodes are expanded (in the worst case) by each of the following search techniques when searching a tree with branching factor *b* to find a goal at a depth of *d*? You can uses ellipses in your answer to indicate a sequence. Do not use big Oh notation.

1. Breadth-first search:

1+b+b2 +b3 +b4 +bd +b(bd-1) O(bd+1)

1. Depth-first search:

O(bm)

1. Depth-limited search (limit = *d*):
2. Iterative deepening depth-first search:

(d+1)b0+db1+(d-1)b2+…+bd O(bd)

**Question 4.** In general, which is the preferred search method when (a) there is a large search space, (b) the depth of the solution is unknown, (c) an optimal solution is desired, and (d) a consistent admissible heuristic is available? (Mark one blank with “X”)

x Depth-first search

b Breadth-first search

x Uniform-cost search

x deep-limited search

c A\* search

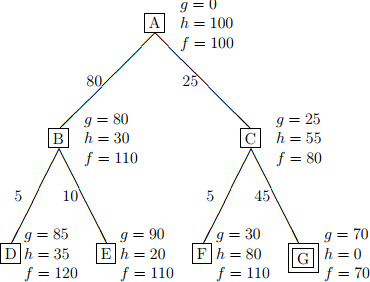
d greedy best-first search

a iterative-deepening search

**Question 5.** Assume that node cost is a function only of node depth, and that the cost function never decreases.

1. Is depth-first search optimal? N
2. Is breadth-first search optimal? N
3. Is uniform-cost search optimal? Y
4. Is depth-limited search optimal? N
5. Is iterative-deepening search optimal? Y

**Question 6.** Use the following tree to indicate the order that nodes are expanded, for different types of search. Assume that G (double box) is the only goal node. Here, path costs are shown to the right of each path, g = cost of path so far, h = estimate of remaining cost to goal, f = estimate of total path cost.



For each search strategy, write down the order in which nodes are expanded. Stop at G.

a. DEPTH-FIRST SEARCH:

ABDECFG

b.BREADTH-FIRST SEARCH:

ABCDEFG

c. ITERATIVE DEEPENING

ABDECFG

d. UNIFORM-COST SEARCH:

ABCFG

1. GREEDY BEST-FIRST SEARCH:

ABEDCG

1. A\* SEARCH:

ACG

**Question 7.** For each of the following search strategies, list the order in which nodes are visited in the state graph below with node a as the start node, and show the frontier nodes at any stage. Where there is a choice of which node to expand, choose the nodes in alphabetical order (e.g, the leftmost branch). Assume that, at any stage of search, nodes that have already been explored (visited) are not explored again.

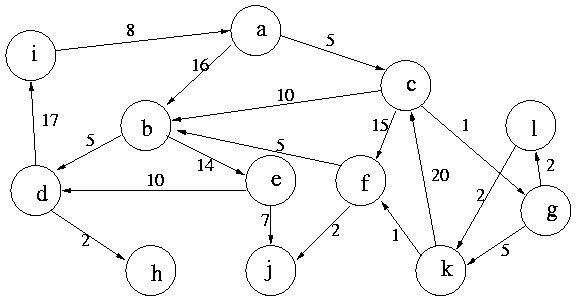
1. Breadth-first search

a bc debfg ihdjdebjlk……

1. Depth-first search

abdiabdia……

1. Depth-first iterative deepening (start with a depth of 1 and increase the depth by 1 on each iteration 同DFS



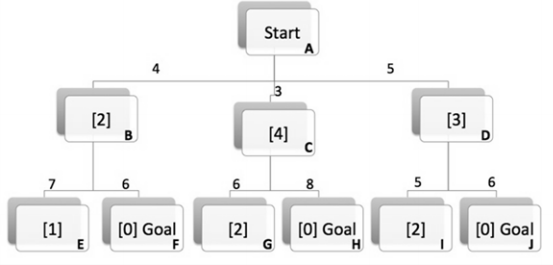
**Question 8.** For the Uniform-Cost-First (Lowest-Cost-First) search strategy, list the order in which nodes are visited in the state graph above with node **a** as the start node and node **j** as the goal node. Assume that, for any frontier node, the information about the lowest-cost path from start node to that node is also kept.

acglkfj

**Question 9.** Recall that for the A\* algorithm, the evaluation function, f, for a node n is such that f(n) = g(n) + h(n) where h(n) is a non-overestimating estimate of the cost to reach a goal from n. What interesting things can you say about the behavior of the algorithm when

(1). h is a perfect estimate (2). h(n) is zero for every n

**Question 10.** Following is an A\* search graph. The weight of each arch (e.g., 5) and the heuristic value at each node is written in brackets (e.g., [3]). The name of the node is a letter in the node's lower right-hand corner. What is the order in which nodes will be "removed" from the fringe for expansion? b



1. A,C,D,J
2. A,C,B,D,F
3. A,C,B,F
4. A,B,C,D,F
5. A,C,D,B,F

**Question 11.** Consider the following search problem:

A

F

B

H

G

C

J

I

D

G

K

E

L

G

1. The above search space has the start node A, and the goal states are labeled G. For each kind of search given below, give the order in which the nodes are searched, and the first solution path found.
   * Depth first search

ABEKG

* + Breadth first search

ABCDEFG

* + Depth first search with iterative deepening

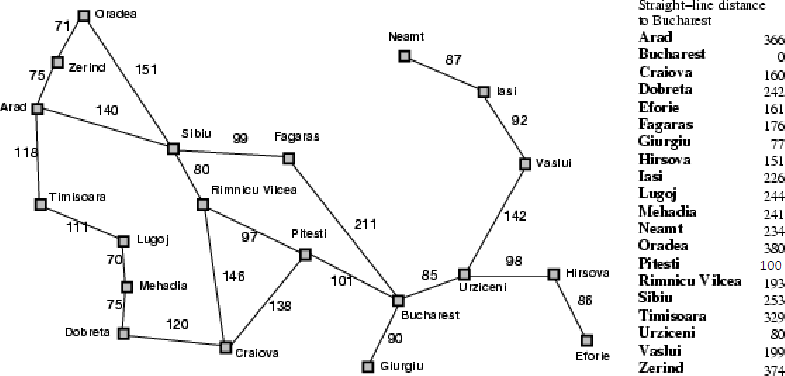
ABEFCG

1. In the following search tree, the value at each node is the result of a heuristic evaluation of that node. **Low is good**. In what order would best-first-search search this

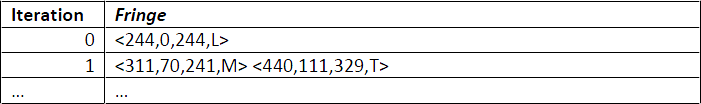
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | start | | | |  | | | | | | | | | | |
|  | | | | | | | | | | | | |  | | | | | | | | | | | | |
|  | | | |  | | | | | | | | |  | | | | | | | | |  | | | |
|  | | A:4 | | | |  | | | | | B:3 | | | |  | | | | | C:5 | | | |  | |
|  | | | |  | | | | | | | | |  | | | | | | | | |  | | | |
|  |  | | | | | |  | | |  | | | | | |  | | |  | | | | | |  |
| D:1 | | |  | | E:7 | | |  | F:2 | | |  | | G:9 | | |  | H:6 | | |  | | I:8 | | |

tree (do not stop at G)? Hint: Use f(n) = g(n) + h(n). Show your work. Note that g(n) is the distance of n from the start node, so for A, B and C g(n) = 1, and for the rest of the nodes, g(n) = 2.

**Question 12.** Consider the graph-search version of A\* using the straight-line distance heuristic hSLD(n) to find a shortest path from Lugoj(L) to Bucharest(B).

a) Finish the table below showing a possible content of the fringe queue in each iteration ( in iteration 0, no nodes have been expanded; in iteration 1, one node has been expanded, etc.). Represent a fringe node by a triple <f,g,h,s>,

where the state s is given by the first letter of the city it represents



|  |  |
| --- | --- |
| 2 | <387,145,242,D> <595,229,366,A> |

b).What solution is returned by A\* after finishing the last iteration in your table?