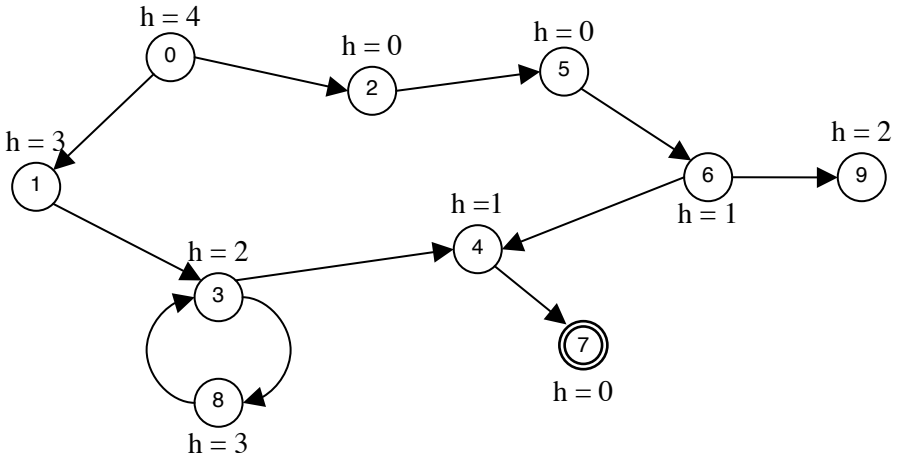
Massachusetts Institute of Technology

16.410-13 Principles of Autonomy and Decision Making

Problem Set #7 Part 2

Problem 3: Search (20 points)

You are trying to find a path from vertex 0 to vertex 7 in the following directed graph using several search algorithms.



Assume the following:

- Edges are length 1.
- h denotes heuristic cost for each vertex.
- Given two or more equally good nodes, explore them in ascending numerical order.
- The search stops as soon as the goal is expanded.

Write the sequence of vertices **expanded** by the specified search methods. A vertex *i* is **expanded** when a search method takes path *N* with head(*N*) = *i* off the queue, and attempts to extend the path *N* to vertex *i*'s children.

Part A. Greedy Search

Show the greedy search expansion sequence up until the first solution path has been found (we have started it for you). Write it as a sequence of pair (<cost>, <vertex>),

where <cost> is the cost the computed when extending a path to <vertex>. Note that no visited list is used.

(4, 0), (0, 2), …

Write the final path found. Is this the optimal solution?

Part B. A Search*

Show the A* expansion sequence up until the first solution path has been found (we have started it for you). Write it as a sequence of pair (<cost>, <vertex>), where <cost> is the cost the computed when extending a path to <vertex>. Please show your work, including the queue and the expansion sequence for each iteration of the search. Write a path in the queue as a pair (<cost>, <path>), where <cost> is the cost estimate computed for a partial path <path>. Assume that an expanded list is used.

Iteration	Q	Expansion Sequence
0	(4, [0])	
1	(1, [2,0]), (4, [1,0])	(4, 0)
2	…	(4, 0), …

Write the final path found. Is this the optimal solution?

You must include in your submission:

A. *The expansion sequence, final path, and the optimality of the final path for Greedy search.*
B. *The expansion sequence and queue for each search iteration, final path, and the optimality of the final path for A* search.*
C. *The amount of time you spent on this problem.*

Problem 4: A* Implementation (40 points)

Implement a Best-first, Greedy, and A* search by implementing the following classes:

A. BestFirstSearch
B. GreedyCostFunction
C. AdmissibleHeuristicCostFunction

Your must use the provided source codes in which some implementations are incomplete. You must not change the specification. Look for "`//*** You fill in here! ***//`" and implement the section as necessary. Read the comments in the source code carefully. The comments will provide additional helpful information.

Part A. BestFirstSearch

Complete the following methods:

```
/**
     * incrementSearch expects that the search already has been initialized
     * @param countdown  the number of queue elements to pop before
     *             halting
     * @return a sequence of actions (a plan) that achieves the goal
     *          or an empty list if one does not exist
     *          or null if time expires.
     **/
    public LinkedList<Action> incrementSearch(int countdown)
```

Part B. GreedyCostFunction

Complete the following method:

```
    /**
     * Returns the cost of a new Node N,
     * The cost is computed as follows:
     * f(N) = h(N)).
     *
     * @param     node        Search node we wish to add
     * @return          Cost with which this node should be prioritized
     */
    public double getCost(Node node) throws NotApplicableException{}
```

Part C. AdmissibleHeuristicCostFunction

Complete the following method:

```
/**
     * Returns the cost of a new path N', which is an extension of a
     * path N to vertex v. The cost is computed as follows:
     * f(N') = g(N') + h(N')
     * @param     node        Search node that we are evaluating
     * @return          Estimated cost of complete path that extends
     *                   <code>node</code> to the goal
     * @throws NotApplicatbleException  if node.State is not of type
     *          <code>RobotState</code>
     */
    public double getCost(Node node) throws NotApplicableException {}
```

You must include in your submission:

A. *[src] Completed implementation of BestFirstSearch.*
B. *[src] Completed implementation of GreedyCostFunction*
C. *[src] Completed implementation of AdmissibleHeuristicCostFunction*
D. *[pdf] The amount of time you spent on this problem.*

Part D. Demonstration

Now you will assess your algorithms by experimenting with BattleCode.

Run your three algorithms with your heuristic function and the map (ps3map.xml) provided. Record:

1. the number of extended paths,
2. the maximum queue size reached during search, and
3. the cost of the path returned.

You must include in your submission:

A. *[pdf] Tabular form of the results.*
B. *[pdf] Discussion of the results.*
C. *[pdf] The amount of time you spent on this problem.*

16.410 / 16.413 Principles of Autonomy and Decision Making

Fall 2010