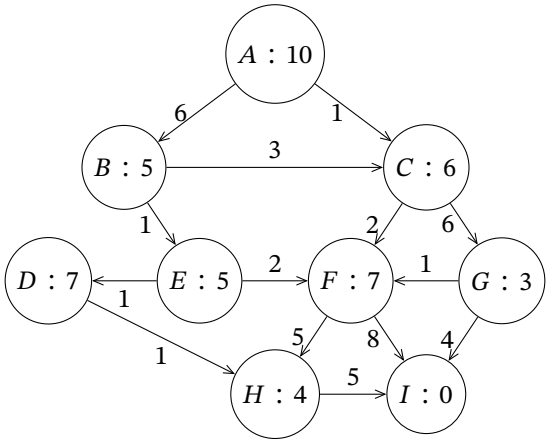
Assignment3 – Search

Problem 3.1 (Search Algorithms)

Consider the following *directed graph*:



Every *node* is labeled with $n : h(n)$ where n is the identifier of the node and $h(n)$ is the *heuristic* for estimating the cost from n to a goal node. Every edge is labeled with its actual cost.

1. Assume that I is the goal node. Argue whether or not the *heuristic* is *admissible*.

Now assume you have already expanded the node A. List the **next** 4 **nodes (i.e., excluding** A**)** that will be expanded using the respective algorithm. If there is a tie, break it using alphabetical order.

2. depth-first search
3. breadth-first search
4. uniform-cost search
5. greedy-search
6. A^*-search

Problem 3.2 (Formally Modeling a Search Problem)

Consider the Towers of Hanoi for 7 disks initially stacked on peg A.

Is this problem deterministic? Is it fully observable?

Formally model it as a *search problem* in the sense of the mathematical definition from the slides. Explain how your *mathematical* definition models the problem.

Note that the formal model only defines the problem — we are not looking for solutions here.

Note that modeling the problem corresponds to defining it in a programming language, except that we use *mathematics* instead of a programming language. Then explaining the model corresponds to documenting your *implementation*.

Problem 3.3 (Heuristic Searches)

Consider the graph of Romanian cities with edges labeled with costs $c(m, n)$ of going from m to n. $c(m, n)$ is always bigger than the straight-line distance from m to n. $c(m, n)$ is infinite if there is no edge.

Our search algorithm keeps:

- a list E of expanded nodes n together with the cost $g(n)$ of the cheapest path to n found so far,
- a fringe F containing the unexpanded neighbors of expanded nodes.

We want to find a cheap path from Lugoj to Bucharest. Initially, E is empty, and F contains only Lugoj. We terminate if E contains Bucharest.

Expansion of a node n in F moves it from F to E and adds to F every neighbor of n that is not already in E or F. We obtain $g(n)$ by minimizing $g(e) + c(e, n)$ over expanded nodes e.

As a *heuristic* $h(n)$, we use the straight-line distance from n to Bucharest as given by the table in the *lectures*.

Explain how the following algorithms choose which node to expand next:

1. greedy search with *heuristic* h
2. A* search with path cost g and *heuristic* h
3. Explain what h^* is here and why h is *admissible*.
4. For each search, give the order in which nodes are expanded.
   (You only have to give the nodes to get the full score. But to get partial credit in case you're wrong, you may want to include for each step all nodes in the fringe and their cost.)

Problem 3.4 (Heuristics)

Consider *heuristic search* with *heuristic* h.

1. Briefly explain what is the same and what is different between A^* search and greedy search regarding the decision which node to expand next.
2. Is the constant function $h(n) = 0$ an *admissible heuristic* for A^* search?