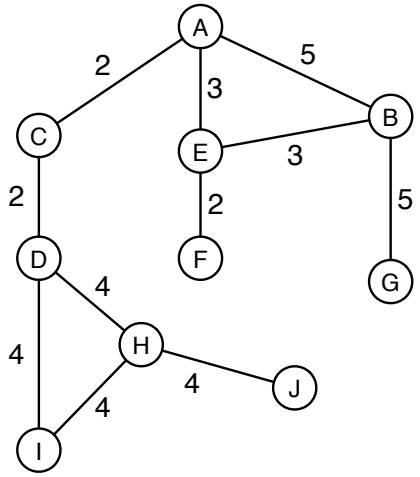
CS255: Artificial Intelligence
Seminar Sheet 1 — Uninformed Search

1. Consider the state space below where nodes represent valid states and edges represent valid transitions from one state to the next. The edges are labelled with their cost of traversal. The start state is labelled A and the goal state is labelled J.



   Using a lowest-cost-first *tree search* (i) show the resulting search tree, (ii) give the state of the queue as the search progresses, (iii) state the sequence in which nodes are selected for expansion and how many nodes are expanded, and (iv) state the route found and its cost.

   When searching the state space, if two nodes are equally desirable then the selection should be done in alphabetic order. When expanding a node n, if n has a child node n' that is also an ancestor of n within the search tree then n' can be omitted from the expansion of n.

2. Using a lowest-cost-first *graph search* in the state space from Question 1, (i) give the state of the frontier at each step of the search, (ii) give the state of the closed set at each step, (iii) list any paths that are pruned, and (iv) state the route found and its cost. You should apply cycle checking and multiple-path pruning.

   When selecting paths from the frontier, if two paths are equally desirable then the selection should be done aphabetically according to the nodes at the end of the paths. If there is a tie-break, then you should select the path that has been in the frontier the longest.

3. In a tree search, what would be the result of using $f(n) = -depth(n)$ for inserting nodes into the queue?