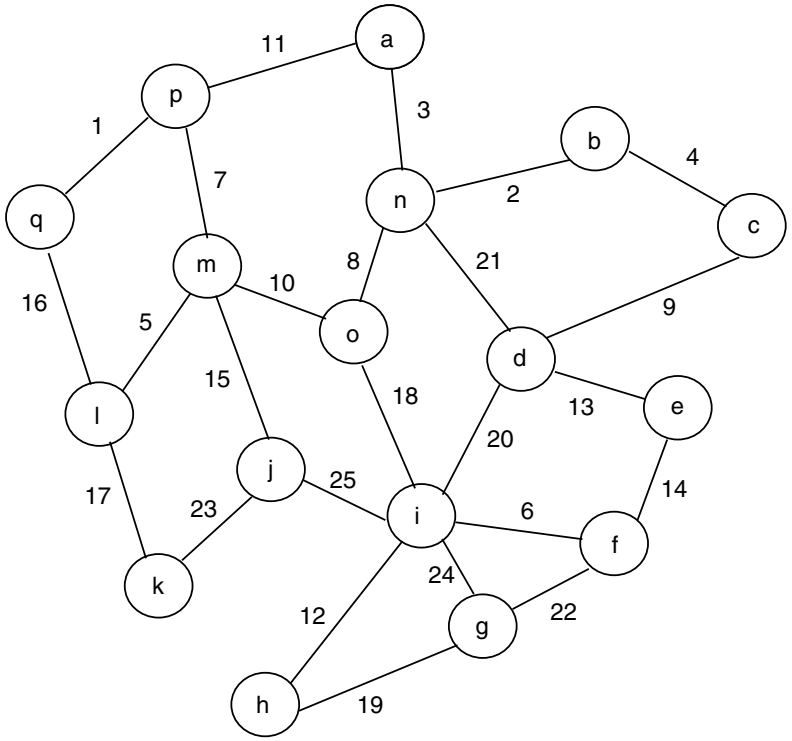
University of Paderborn
Planning and Heuristic Search
WS 2021/22
Dr. Th. Lettmann

November 11, 2021

Class Exercise 3

Exercise 1 : Search Graph for Best-First Search

Let be given the following search graph with edges representing operators that have some cost value assigned. Path costs are defined as sum of edge cost values. The evaluation function $f(n)$ returns cost of the solution base given by n (sum of edge cost values along the path).



Start node is node a; there are no goal nodes. Tie-Breaking: If there are multiple nodes with minimal path cost values, select nodes that were generated by earliest node expansions and among these select in alphabetical order.

(a) Describe the first six node expansions by algorithm Basic_BF. What is the resulting traversal tree defined by the nodes in OPEN and CLOSED and their backpointers.

(b) Describe the first six node expansions by algorithm BF. What is the resulting traversal tree defined by the nodes in OPEN and CLOSED and their backpointers.

(c) What differences between Basic_BF and BF can be observed with respect to the resulting traversal trees?

(d) What differences can be observed with respect to the resulting traversal trees if we replace the algorithms by Basic_BF* and BF* ?

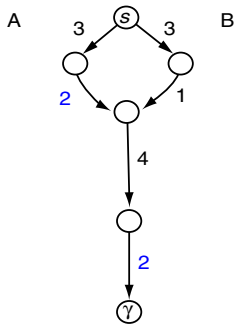
Exercise 2 : Optimum Cost Solution Paths

Let f be an optimistic evaluation function for an OR-graph search problem.

How would you have to change BF resp. BF^* to get ALL optimum cost solution paths?

Exercise 3 : Best-First Search

The figure below shows an example search space graph; the numbers denote edge weights. Best-First search algorithms BF and BF* store at most one solution base for each node n. If the goal is to find the shortest path from s to t, this is not an issue, since edge A can be discarded. However, if the goal is to find a solution path with two edges of the same weight, there is a problem.



(a) Describe the problem.

(b) What requirement must be satisfied so that BF can be applied to a search problem?

Exercise 4

What is the worst case runtime (number of node expansions) and memory consumption (number of nodes in OPEN and CLOSED) for DFS, BFS, Basic_BF*, BF*, and A* applied to a uniform directed binary tree of depth k ?