Technische Universität München
Department of Computer Science
Prof. Michael Beetz, Ph.D.
Daniel Nyga, M.Sc.

WS 2011/2012
Worksheet 3
08.11.2011

Exercises for the Lecture Techniques in Artificial Intelligence

Solving Problems by Search

Hint: For these exercises the auxiliaries provided at `http://www.aispace.org/search/index.shtml` *may be employed, which implement a selection of the algorithms under consideration.*

1) *Formulation of search problems*

   Give state space, starting state, successor and predecessor function/operators, goal test and cost function for each of the following search problems:

   (a) A map shall be colored with four colors in a way that two neighboring regions do not have the same color.

   (b) A 30cm tall monkey, that can't jump high is trapped in a room where bananas hang in 90cm height from the ceiling. He wants to get a banana. Apart from the monkey and bananas, the room contains two stackable and movable 30cm high cubic blocks that are strong enough to support the monkey standing on top.

   (c) You have three containers with 12, 8 and 3 liters capacity and a large barrel of water. You can fill water into containers and pour their content either into another container or on the floor. You want to measure exactly one liter

2) Consider the state space defined by a start state 1 and a successor function that assignes the states $2n$ und $2n + 1$ as children of each state n. The goal is to find the path to a given number x.

   (a) Draw the state space for the states 1 to 15.

   (b) Let $x = 11$. List the sequence in which the states are visited in a

      i. Breadth-first search

      ii. Depth-first search

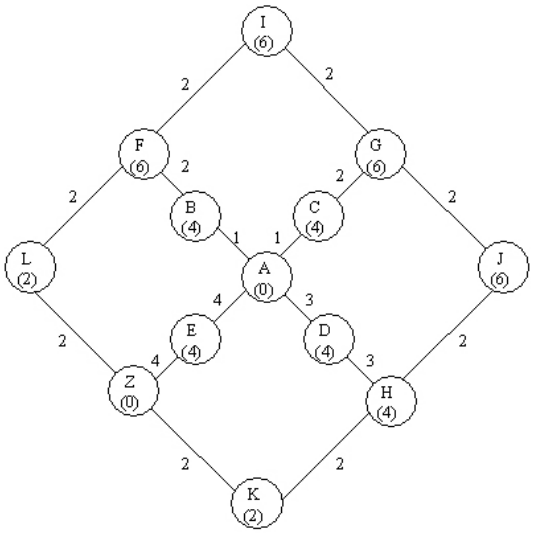
      iii. iterative deepening search

   (c) Would a bi-directional search work aswell for this problem? If so, describe in detail how it works.

   (d) What branching factors are effective for the bi-directional search?
   Can this be exploited to reformulate the problem in a way that leads to a more efficient solution? If so, describe it.

3) *The strategy of graph search*

Consider the adjacent graph for a search problem from A to Z. All edges of the graph are undirected (bi-directional) and labelled with their cost. The labels of the nodes contains a heuristic approximation (in paranthesis) w.r.t. the remaining cost to the goal node Z. Whenever several nodes have identical cost for visiting at a certain point of time, the nodes shall be visited in the lexical order of their labels.



When one searches in such a graph, a search tree is created by assigning adjacent nodes as child nodes of a node (e.g. in a directed graph nodes that can be reached via a directed edge). This means the undirected edges lead to all involved nodes being successors of each other. Therefore, the search tree is of of infinite size although the graph is finite, since nodes could be visited multiple times. For that reason, visiting nodes repeatadly should be avoided. The general search algorithm Graph-Search manages a list of alread visited nodes (*closed*) to avoid visiting the same node multiple times.

```
function GRAPH-SEARCH( problem, fringe) returns a solution, or failure
    closed ← an empty set
    fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
    loop do
        if fringe is empty then return failure
        node ← REMOVE-FRONT(fringe)
        if GOAL-TEST[problem](STATE[node]) then return node
        if STATE[node] is not in closed then
            add STATE[node] to closed
            fringe ← INSERTALL(EXPAND(node, problem), fringe)
    end
```

In this exercise, Graph-Search shall be used. It is to be noted that using Graph-Search instead of Tree-Search can affect the completeness, optimality and the memory consumtion of the algorithms!

(a) Conduct a breadth-first search, depth-first search, greedy best-first search and A*-search on the graph. Which nodes are expanded and which path is returned as the result?

(b) Did the A*-search find the optimal path? Why (not)? How about the other methods?

(c) Do all these search methods find a solution in general, if one exists?