Homework Assignment due 09-06-09

1) Bellman's algorithm

Consider the following version of Bellman's algorithm:

Algorithm 3: [Bellman, 1957] Iterative Suche nach dem optimalen Suchbaum T.

1: **for** $i = 0, \ldots, n$ **do**
2: $\quad w_{i+1,i} = q_i$
3: $\quad c_{i+1,i} = 0$
4: **end for**
5: **for** $k = 0, \ldots, n-1$ **do**
6: $\quad$ **for** $i = 1, \ldots, n-k$ **do**
7: $\quad\quad j = i + k$
8: $\quad\quad$ Bestimme m mit $i \le m \le j$, so dass $c_{i,m-1} + c_{m+1,j}$ minimal ist.
9: $\quad\quad r_{i,j} = m$
10: $\quad\quad w_{i,j} = w_{i,m-1} + w_{m+1,j} + p_m$
11: $\quad\quad c_{i,j} = c_{i,m-1} + c_{m+1,j} + w_{i,j}$
12: $\quad$ **end for**
13: **end for**

This does not consider the new value for $w_{i,j}$ which also depends on m

Check if this version is correct by solving one of the following alternatives:

i) Find arguments why the search for the optimal m need not consider $w_{i,j}$

ii) Give a counterexample in which this algorithm makes the wrong decision in line 8.

Homework Assignment due 09-06-09

2) Algorithm of Dijkstra

Prove the following time complexities for a graph with n nodes and m edges:

General Time complexity: $O((m+n)\log n)$

for arbitrary graphs: $O(n^2 \log n)$
for graphs with a constant number of neighbors per node: $O(n \log n)$

Give arguments by specifying how to solve the details in the following outline:

Algorithm of Dijkstra for SSSP: (for graphs G with nonnegative edge costs only)

- Initialize the node set **Done** by s;
  Initialize the node set **Undone** by all other nodes of graph G;
  For all nodes v of the graph G:
  Let label (v) := length of edge from s to v (∞ if no edge is existing, 0 if v = s);
- While **Undone** is not empty:
  Search and delete the node v from **Undone** with minimal label;
  Insert v into **Done**;
  Update all neighbors n of v that are in **Undone**:
  If label (n) > label (v) + length of edge between v and n:
  Replace label (n) by that number;
  Let v be the predecessor of n.

Homework Assignment due 09-06-09

3) Algorithm of Floyd-Warshall

Prove the correctness of the algorithm by complete induction over k:

Let V = {1,...n}.
$d_{ij}^{(k)}$ is the length of the shortest path between i and j using in between at most nodes from {1,...k}.

for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, n$ **do**
        $d_{ij}^{(0)} = \begin{cases} c(i,j): & \text{falls } (i,j) \in E \\ \infty: & \text{sonst} \end{cases}$
    end for
end for
for $k = 1, \ldots, n$ **do**
    for $i = 1, \ldots, n$ **do**
        for $j = 1, \ldots, n$ **do**
            $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$
        end for
    end for
end for