Albert-Ludwigs-Universität Freiburg
Institut für Informatik
P.D. Dr. Alexander Souza
Thomas Janson

WS 2011/12
Sheet 07

Algorithm Theory

Exercise 1 (Dijkstra's algorithm) [Points: 2]

Extend Dijkstra's algorithm to create the shortest-path tree.
Please add the missing instructions to the algorithm presented in the lecture:

Algorithm 3 Dijkstras Algorithm(Graph G, Node s): ?

```

DIST[$s$] $\leftarrow$ 0;

for all $v \in V \setminus \{s\}$ do

    DIST[$v$] $\leftarrow \infty$;

    Insert $(U, \infty, v)$

end for

while $\neg$Empty $(U)$ do

    $(d, u) \leftarrow$ DeleteMin $(U)$;

    for all $e = (u, v) \in E$ do

        if DIST[$v$] $>$ DIST[$u$] + c $(u, v)$ then

            DIST[$v$] $\leftarrow$ DIST[$u$] + c $(u, v)$;

            DecreaseKey $(U, v, \text{DIST}[v])$

        end if

    end for

end while

```

Exercise 2 (Dijkstra's algorithm) [Points: 4]

Dijkstra's algorithm works correctly with non-negative edge weights. Give an instance of a network with arbitrary weights for which Dijkstra's algorithm does not work correctly.

Exercise 3 (Shortest paths) [Points: 5]

Topological sorting is used for finding the shortest path in acyclic networks (Directed Acyclic Graph, DAG for short). The sources in the network are all nodes without incoming edges. Consider the following algorithm which was explained at an example in the lecture (slide 16):

Algorithm 4 Topological Sorting(DAG $G = (V, E)$): Array *num*

```
1: $current \leftarrow 0$
2: $S \leftarrow \{v \in V : v$ is source in $(V, E)\}$;
3: while $S \neq \emptyset$ do
4:     $u \in S$, $S \leftarrow S \setminus \{u\}$;
5:     $current \leftarrow current + 1$;
6:     $num[u] \leftarrow current$;
7:     for all $(u, v) \in E$ do
8:         $E \leftarrow E \setminus \{(u, v)\}$;
9:         if $v$ is source in $(V, E)$ then
10:            $S \leftarrow S \cup \{v\}$;
11:        end if
12:    end for
13: end while
14: return $num$;
```

Give an implementation of Topological Sorting that yields running time $\mathcal{O}(n + m)$ where $n = |V|$ and $m = |E|$.

Hint: Especially consider the implementation details of how to efficiently determine if a node is a source (Algorithm 6 Line 2 and Line 9).

Exercise 4 (Minimum spanning trees) [Points: 4]

Construct a graph where the minimum spanning tree is not equal to the shortest path tree of a single source.

Exercise 5 (Bin packing) [Points: 5]

Show how the running time of First Fit can be improved from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \cdot \log n)$.
(Compare lecture slide 18)

The following two exercises are bonus exercises for additional 10 points.

Exercise 6 (Bin packing) [Points: 4*]

Suppose that in an instance $B_1, \ldots, B_n$ of bin packing we have $B_i > 1/3$ for each $i = 1, \ldots, n$. Show how to solve this problem optimally in $\mathcal{O}(n \log n)$ time.

Exercise 7 (Bin packing) [Points: 6*]

We call an algorithm for bin packing *monotonic* if the number of bins it uses for packing a subsequence of the items is at most the number of bins it uses for packing all n items. Show that Next Fit is monotonic, but First Fit is not.