Algorithms & Datastructures
Laboratory Exercise Sheet 10

Wolfgang Pausch <wolfgang.pausch@uibk.ac.at>
Heiko Studt <heiko.studt@uibk.ac.at>
René Thiemann <rene.thiemann@uibk.ac.at>
Tomas Vitvar <tomas.vitvar@uibk.ac.at>

June 1st, to be discussed on June 8th

Exercise 1) Dynamic Programming You have to cut a cable of length n into pieces, where each piece has a length $\ell \in \mathbb{N}$ with $1 \leq \ell \leq n$. You can sell each piece of length $1 \leq \ell \leq n$ for a price p_ℓ. You have to determine the maximum revenue for a cable of length n for a given table of prices. For example, consider a cable of length 4 with prices $p_1 = 1$, $p_2 = 5$, $p_3 = 6$, $p_4 = 9$. Then one can cut the cable into pieces of length 1 and 3 (resulting in 7) or in two pieces of length 2 (resulting in 10), or in one piece of length 4 (resulting in 9), or ....

1. Implement a recursion algorithm that computes the maximum revenue for a cable of length n. Use the following code skeleton.

```
public class CutCable {

    public static int cutCable(byte[] p, int n) {
        // implement me!
    }

    public static void main(String[] args) {
        byte[] p = {1,5,6,9};
        System.out.println(cutCable(p, p.length));
    }

}
```

2. Draw the recursion tree for the cable of length 4.
3. Determine the running time of the algorithm. Why is the algorithm inefficient?
4. Use dynamic programming for optimal cable cutting. Implement the improved algorithm.

Exercise 2) Greedy Algorithm You have 8 cities connected with roads as shown in Figure 1 where each road has its length. In winter you want to maintain only a set of roads such that still all cities are connected and that the length of the opened roads is minimal.

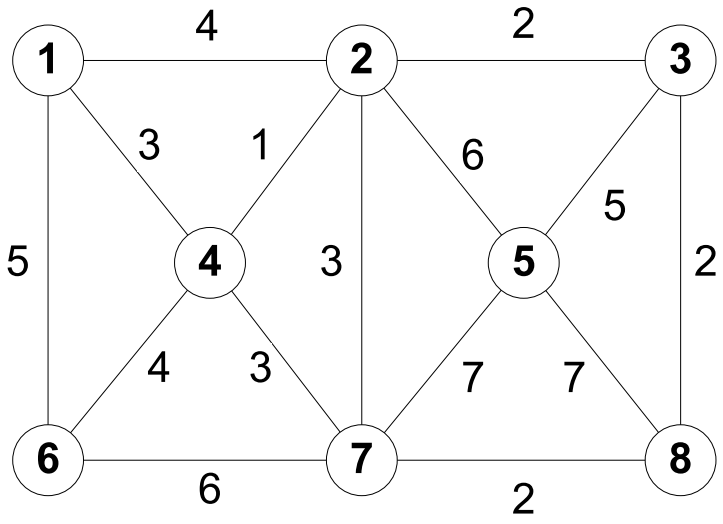

Figure 1: Cities and their connections with lengths

Design a greedy algorithm to find the minimal connections among all cities and show each iteration step (hint: the result is a tree, i.e. there must not exist a circle). Give an intuitive idea why your algorithm returns an optimal solution.