Dr. Ulf-Peter Schroeder

Paderborn, 17. Mai 2017
Abgabe: Keine!

Übungen zur Vorlesung
Methoden des Algorithmenentwurfs
SS 2017
Blatt 4

Aufgabe 9:
Let $G = (V, E)$ be an undirected graph with n nodes. Recall that a subset of the nodes is called an *independent set* if no two of them are joined by an edge. Finding large independent sets is difficult in general; but here we'll see that it can be done efficiently if the graph is "simple" enough.
Call a graph $G = (V, E)$ a *path* if its nodes can be written as $v_1, v_2, \ldots, v_n$, with an edge between v_i and v_j if and only if $|i - j| = 1$. With each node v_i we associate a positive integer weight w_i. The goal is to solve the following problem: Find an independent set in a path G whose total weight is *as large as possible*.

a) Give an example to show that the following algorithm *does not* always find an independent set of maximum total weight.

Algorithm 1 The "heaviest-hitter" greedy algorithm

```
S ← ∅
while some node remains in G do
    pick a node v of maximum weight
    S ← S ∪ {v}
    delete v and its neighbors from G
return S
```

---

b) Give an algorithm based on *dynamic programming* that takes an n-node path G with weights and returns an independent set of maximum total weight. The running time should be $O(n)$.

Aufgabe 10:
Suppose you are running a lightweight consulting business – just you, two associates, and some rented equipment. Your clients are distributed between the East Coast and the West Coast, and this leads to the following question. Each month, you can either run your business from an office in New York (NY) or from an office in San Francisco (SF). In month i, you will incur an *operating cost* of N_i if you run the business out of NY; you will incur an operating cost of S_i if you run the business out of SF. (It depends on the distribution of client demands for that month.) However, if you run the business out of one city in month i, and then out of the other city in month $i + 1$, then you incur a fixed *moving cost* of M to switch base offices.
Given a sequence of n months, a *plan* is a sequence of n locations – each one equal to either NY or SF – such that the i^{th} location indicates the city in which you will be based in the i^{th}

month. The *cost* of a plan is the sum of the operating costs for each of the n months, plus a moving cost of M for each time you switch cities. The plan can begin in either city.
Given a value for the moving cost M, and sequences of operating costs $N_1, \ldots, N_n$ and $S_1, \ldots, S_n$, find a plan of minimum cost via a *Dynamic Programming* approach..

Aufgabe 11:
Take a look at *Aufgabe 4* on *Blatt 2*. Show how to find the optimal pair of days i and j in time $O(n)$ via *Dynamic Programming*.