Mustafa Kemal Kücük mustafak@uni-bremen.com
Darius Dolha dolha@uni-bremen.de
Vasu Yaduvanshi vasu@uni-bremen.de Übungsgruppe 51

1 Übung 5.1

We are initially going to use the master theorem in order to find a closed form in Θ-notation for the recursion ecuation.
Let $a \geq 1$, $b \geq 1$ constants, $f : \mathbb{N} \to \mathbb{N}$ and $T(n) = \cdot T(\frac{n}{2}) + 3$. We know that $a = 2$, $b = 2$ and $f(n) = 3$ (constant function).
If $f(n) \in \Theta(n^{\log_b a})$ for an $\epsilon > 0$ then:

$$T(n) \in \Theta(n^{\log_b a} \log n)$$

This will be true, but we need to prove by induction. We need to prove that:

$$T(n) \in \Theta(\log n)$$

1.1 First $T(n) \in O(\log n)$

Induction of n. Induction hypotheses:
$T(m) \leq c \log m$ for all $m < n$
Induction step: $T(n) = T(\frac{n}{2}) + 3 \leq \log c(\frac{n}{2}) + 3 = c \log n - c \log 2 + 3 \leq c \log n$. Only when $3 - c \leq 0$. So for $c \geq 3$, the inequality holds. $(\log 2 = 1)$
Base case: $T(2) = T(1) + 3 = 4 \leq c \log 2 = c$. Let $c = 4$. $T(2) \leq 4$

1.2 Second $T(n) \in \Omega(\log n)$

Induction hypotheses: $T(m) > \log m, \forall m < n$. Let $c = 1$
$T(n) = T(\frac{n}{2}) + 3 > \log n - \log 2 + 3 = \log n + 2 > \log n$.
Base case: $T(2) = 4 > \log 2 = 1$

1.3 Conclusion

Since we also know that $\Theta(g) = O(g) \cap \Omega(g)$, we can conclude that:

$$T(n) \in \Theta(\log n)$$

2 Übung 5.2

We are going to use the Bellman-Ford Algorithm to find the shortest path. It can be seen that the graph has a negative cycle($B- > C- > B$). We are going to execute the algorithm anyway. First, we will make $n - 1$ iterations with $n = 4$.
The algorithm revolves around constant relaxation of the edges.

for all $(u, v) \in E$ **do**
  if dist$(u) + c(u, v) <$ dist(v) **then**
    dist$(v) \leftarrow$ dist$(u) + c(u, v)$
  end if
end for

The table represents the node(v) and the shortest distance to that node dist(v), starting from node A.

Iteration 1:

A	B	C	D
0	∞	∞	∞

A	B	C	D
0	$\cancel{2}$ -2	3	1

First $dist(A) = 0$, $dist(B) = dist(C) = dist(D) = \infty$. Then we have $dist(A) = 0$, $dist(B) = 2$ $dist(C) = 3$ $dist(D) = 1$ using the edges $(A, B), (A, C), (A, D)$. The edges $(B, A), (B, D), (C, D), (D, C)$ do not change the result. The edge (D, B) leads us to $dist(B) = -2$

Iteration 2

A	B	C	D
$\cancel{0}$ -1	$\cancel{-2}$ -3	3	$\cancel{1}$ 0

Iteration 3

A	B	C	D
$\cancel{-1}$ -2	$\cancel{-3}$ -4	$\cancel{3}$ 2	$\cancel{0}$ -1

Starting the negative cycle test, we notice that, for example: $dist(B) + c(B, D) < dist(D)$ so there exists a negative cycle in the graph, since the relaxation can continue past 3 iterations

3 Übung 5.3

3.a

The greedy algorithm follows these steps:

1. $i := 1$
2. While $i \leq n$:
   - If $c_{i+1} > r_i + r_{i+1}$, skip week i and complete a complex task in week $i + 1$.
   - Otherwise, complete a routine task in week i.
3. $i := i + 1$ or $i := i + 2$ based on the choice.

Counterexample: Consider the following instance:

$$\begin{aligned} &n = 3, \\ &r_1 = 5,\ r_2 = 1,\ r_3 = 6, \\ &c_1 = 0,\ c_2 = 10,\ c_3 = 15. \end{aligned}$$

Optimal Solution: Complete a routine task in week 1 and week 3, yielding a total of 11.
Greedy Algorithm: Completes a complex task in week 2, yielding 10.
This shows the greedy algorithm does not yield the optimal solution.

3.b

The DP table has dimensions $n \times 3$. The element $M[0, i]$ stores the maximum revenue by week i if a routine task is done in week i. Similarly, $M[1, i]$ stores the revenue if a complex task is done in week i, and $M[2, i]$ represents the revenue if no task is done.
The base cases are: $M[0, 0] = r_1$, $M[1, 0] = 0$, and $M[2, 0] = 0$. The recursive relation is as follows:

$$\begin{aligned} &M[0, i] = r_i + \max\{M[0, i-1], M[1, i-1], M[2, i-1]\}, \\ &M[1, i] = c_i + M[2, i-1], \\ &M[2, i] = \max\{M[0, i-1], M[1, i-1], M[2, i-1]\}. \end{aligned}$$

Each step executes in constant time, except for the loop, which takes $O(n)$. Hence, the overall complexity is $O(n)$.
The algorithm terminates successfully, ensuring that for each week up to n, the DP table stores the optimal revenue for all cases. By iterating through the weeks, the final value in the DP table represents the optimal solution.

Algorithm 1 Compute Maximum Revenue using Dynamic Programming

Create array M of size $n \times 3$
Set $M[0,0] = r_1$
Initialize $M[1,0] = 0$ and $M[2,0] = 0$
for $i \leftarrow 1$ to n **do**
    $M[0,i] = r_i + \max(M[0,i-1], M[1,i-1], M[2,i-1])$
    $M[1,i] = c_i + M[2,i-1]$
    $M[2,i] = \max(M[0,i-1], M[1,i-1], M[2,i-1])$
end for
return $\max(M[0,n], M[1,n])$

3.c

To reconstruct the task plan, trace back from $DP[n]$:

1. If $DP[i] = DP[i-1] + r_i$, complete a routine task in week i.
2. If $DP[i] = DP[i-2] + c_i$, skip week $i-1$ and complete a complex task in week i.
3. Repeat until week 1 is reached.

This yields the optimal task plan for maximum revenue.

4 Übung 5.4

1) Aussage 1 ist FALSCH.
Gegenbeispiel: Betrachte einen Graphen mit 3 Knoten $V = \{a, b, c\}$ und Kanten:

- a-b mit Gewicht 1
- b-c mit Gewicht 1
- a-c mit Gewicht 3

Der MST besteht aus den Kanten a-b und b-c. Für die Knoten a und c ist der kürzeste Weg zwischen ihnen aber a-c (Gewicht 3), nicht der Weg über b (Gewicht 2), der nur MST-Kanten verwendet.

2) Aussage 2 ist FALSCH.
Dies ist eine stärkere Version von Aussage 1 - wenn schon Aussage 1 falsch ist, muss auch diese falsch sein. Das gleiche Gegenbeispiel kann verwendet werden.

3) Aussage 3 ist FALSCH.
Auch hier kann das gleiche Gegenbeispiel verwendet werden. Es gibt keinen Knoten s, von dem aus alle kürzesten Wege zu anderen Knoten nur MST-Kanten verwenden würden.

4) Aussage 4 ist WAHR.
In diesem speziellen Fall, wo alle Kanten Gewicht 1 haben:

- Jeder kürzeste Weg ist einfach der Weg mit den wenigsten Kanten
- In einem MST gibt es zwischen je zwei Knoten genau einen Pfad
- Da alle Kanten Gewicht 1 haben, ist jeder Pfad im MST auch ein kürzester Weg
- Wähle einen beliebigen Knoten s im MST als Wurzel
- Von s zu jedem anderen Knoten t gibt es genau einen Pfad im MST, und dieser ist automatisch ein kürzester Weg in G

Dieser Fall unterscheidet sich fundamental von den vorherigen, weil die einheitlichen Kantengewichte garantieren, dass MST-Pfade auch kürzeste Wege sind.