Karlsruher Institut für Technologie

Institut für Theoretische Informatik
T.T.-Prof. Dr. Thomas Bläsius
https://scale.iti.kit.edu

# Zusatzaufgaben 05

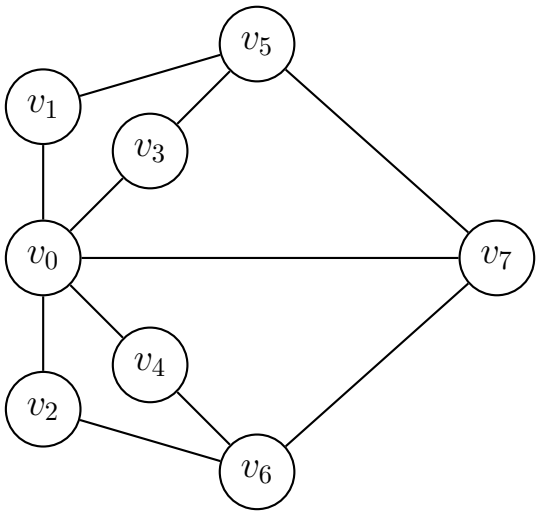
## Algorithmen I – Sommersemester 2023

**Gesamtpunkte**: 36

### Aufgabe 1 - Eulerkreise (7 Punkte)

Gegeben sei ein zusammenhängender Graph $G = (V, E)$. Unter einem Eulerkreis verstehen wir einen Kreis in $G$, der alle Kanten aus $E$ enthält. Wir wollen uns nun einen Algorithmus überlegen, der in einem gegebenen Graphen einen Eulerkreis bestimmt, sofern er denn existiert.

1. Besitzt der folgende Graph einen Eulerkreis? Falls dem so ist, dann gib die Kantenreihenfolge in diesem Kreis an. Ansonsten begründe kurz, warum dem nicht so ist. (1 Punkt)




2. Zeige: $G$ enthält genau dann einen Eulerkreis, wenn die Kantenmenge $E$ die Vereinigung disjunkter Kreise in $G$ ist. (3 Punkte)
*Hinweis:* Disjunkte Pfade sind solche, die sich keine Kanten teilen.

3. Beschreibe einen Algorithmus, der in $\Theta(n+m)$ bestimmt, ob ein gegebener Graph einen Eulerkreis enthält. Begründe kurz Laufzeit und Korrektheit deines Algorithmus. (3 Punkte)

## Aufgabe 2 - Und täglich grüßt das Murmeltier (25 Punkte)

Die folgenden Probleme lassen sich jeweils mit einem dynamischen Programm lösen. Deine Aufgabe ist es, dir zu überlegen, wie diese Programme aussehen sollten. Beantworte dazu für jedes Problem die folgenden Fragen:

i) Wie lautet die Lösung der Beispielinstanz, die dir gegeben ist?

ii) Wie kann man eine gegebene Probleminstanz in Teilprobleme zerlegen? Wie sieht die Lösung eines Teilproblems aus? Wann kann ein Teilproblem nicht mehr weiter zerlegt werden?

iii) Wie lautet die Rekurrenz, anhand derer die Lösung einer Probleminstanz berechnet werden kann? Benenne dabei explizit die Parameter, die in deiner Rekurrenz vorkommen.

iv) Wie erhalten wir eine tatsächliche Lösung für eine gegebene Probleminstanz? Wie nutzen wir dabei die Rekurrenz?

1. MaxSumIncreasingSubarray
   Gegeben sei ein Array $A\colon [\mathbb{Z}; n]$. Wir suchen ein zusammenhängendes Teilarray $A[i \dots j]$ so, dass $A[k] \leq A[k+1]$ für alle $i \leq k < j$ und $\sum_{k=i}^{j} A[k]$ maximal ist.

   Beispielinstanz: $A = \langle -1, 3, 5, -2, 7, -2, 1, 8 \rangle$

2. MinSumSubarray
   Gegeben sei ein Array $A\colon [\mathbb{Z}; n]$. Wir suchen ein zusammenhängendes Teilarray $A[i \dots j]$ so, dass $\sum_{k=i}^{j} A[k]$ minimal wird. Zum Beispiel wäre für $A = \langle 1, 3, 5, -2, 4, -1, -2 \rangle$ das Teilarray $A[5 \dots, 6] = \langle -1, -2 \rangle$ die eindeutige Lösung.

   Beispielinstanz: $A = \langle 1, 8, 4, -1, 3, -2, 1, -2 \rangle$

3. minCostWalk
   Gegeben sei ein Raster mit $n \times m$ Zellen. Jede Zelle $(i, j)$ enthält einen

Wert $\texttt{val}(i, j) \in \mathbb{N}$. Wir wollen uns schrittweise durch dieses Raster bewegen, wobei wir uns in einem Schritt immer nur entlang einer Koordinate und nur in positiver Richtung bewegen dürfen. Stehen wir z.B. an Zelle $(3, 4)$, können wir uns nur zu den Zellen $(4, 4)$ oder $(3, 5)$ bewegen. Gesucht ist nun eine Folge von Zellen, wobei aufeinander folgende Zellen jeweils genau einen gültigen Schritt entfernt sind, von der Zelle $(0, 0)$ zur Zelle $(n - 1, m - 1)$ so, dass die Summe der `val`-Werte der durchlaufenen Zellen minimal ist.

Beispielinstanz: Das folgende Raster mit $3 \times 4$ Zellen (die Zelle (0, 0) steht in der linken oberen Ecke):

| | | |
|---|---|---|
| 2 | 4 | 7 |
| 2 | 5 | 1 |
| 9 | 1 | 1 |
| 4 | 2 | 3 |

4. LongestCommonSubsequence
   Gegeben seien zwei Arrays $A$: [char; $n$], $B$: [char; $m$]. Wir suchen die längste Zeichenfolge, die sowohl in $A$ als auch in $B$ enthalten ist.
   Beachte: diese Zeichenfolge muss nicht in einem zusammenhängenden Teilarray von $A$ oder $B$ stehen. Die längste gemeinsame Sequenz von $\langle$`s, c, h, a, f, f, n, e, r`$\rangle$ und $\langle$`k, a, r, t, o, f, f, e, l`$\rangle$ etwa ist $\langle$`a, f, f, e`$\rangle$.

   Beispielinstanz:

$$A = \langle \texttt{a, l, p, e, n, g, l, o, c, k, e} \rangle$$

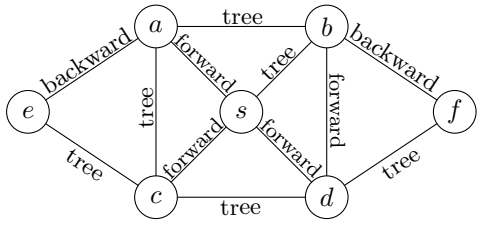
$$B = \langle \texttt{f, r, a, k, t, a, l, g, e, o, m, e, t, r, i, s, c, h} \rangle$$

5. LongestPalindromicSubstring
   Gegeben sei ein Array $A$: [char; $n$]. Wir suchen das längste zusammenhängende Teilarray von $A$, welches ein Palindrom bildet.

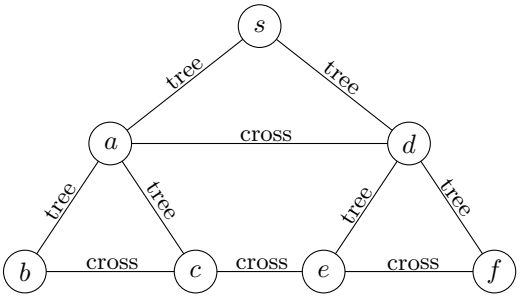
   Beispielinstanz: $A = \langle$`b, o, s, s, l, e, v, e, l`$\rangle$

## Aufgabe 3 - Suchen suchen (4 Punkte)

Gegeben seien die beiden Graphen, $G_1$ und $G_2$, deren Kanten bereits nach einer Suche markiert worden sind.




$G_1$




$G_2$

1. Entscheide, ob die Labels der Graphen $G_1$ das Ergebnis einer Breitensuche ist. Wenn ja, gib eine mögliche Knotenreihenfolge der Abarbeitung an.

2. Entscheide, ob die Labels der Graphen $G_1$ das Ergebnis einer Tiefensuche ist. Wenn ja, gib eine mögliche Knotenreihenfolge der Abarbeitung an.

3. Entscheide, ob die Labels der Graphen $G_2$ das Ergebnis einer Breitensuche ist. Wenn ja, gib eine mögliche Knotenreihenfolge der Abarbeitung an.

4. Entscheide, ob die Labels der Graphen $G_2$ das Ergebnis einer Tiefensuche ist. Wenn ja, gib eine mögliche Knotenreihenfolge der Abarbeitung an.