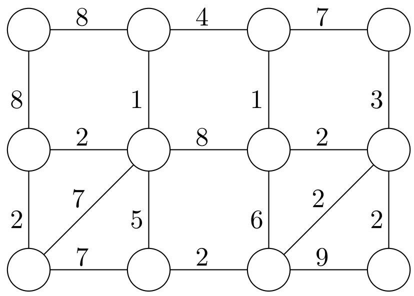
Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

Datum: 27.06.2019

Übungsblatt 11

Aufgabe T37

Finden Sie jeweils einen minimalen Spannbaum mithilfe der Algorithmen von Kruskal und Prim für folgenden Graphen:



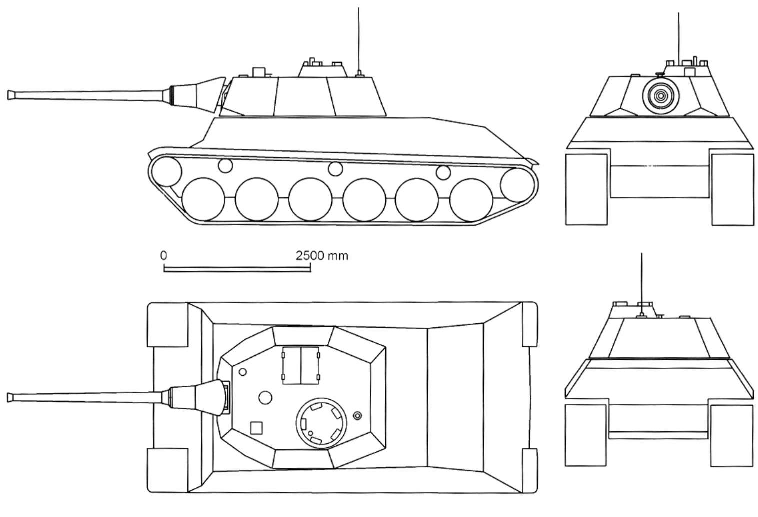
Aufgabe T38

Gegeben sind n sportliche Aktivitäten mit je einem zugeordneten Zeitintervall. Die $i$te sportliche Aktivität dauert dabei von Zeitpunkt s_i bis f_i mit $s_i < f_i$.
Entwerfen Sie einen effizienten Algorithmus, der eine möglichst große Menge an sportlichen Aktivitäten auswählt, sodass sich keine ausgewählten Aktivitäten überlappen. Was ist die asymptotische Laufzeit ihres Algorithmus? Hinweis: Verwenden Sie einen Greedy-Algorithmus.

Aufgabe T39

Gegeben sei eine $\mathbf{Z}^{n\times 4}$-Matrix. Der *Wert* einer Zeile dieser Matrix sei die Summe der Einträge dieser Zeile. Wir wollen eine linear unabhängige Menge an Zeilen auswählen, sodass die Summe ihrer Werte maximal ist.
Geben Sie einen Algorithmus an, welcher dieses Problem optimal löst. Beweisen Sie, dass ihr Algorithmus optimal ist.

Aufgabe T40

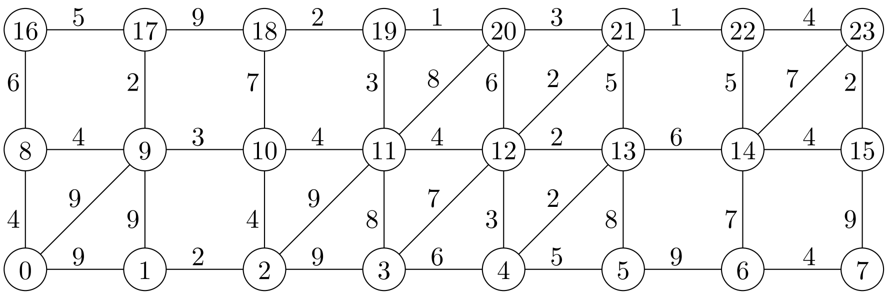


Aufgabe T41

Es sei $G = (V, E)$ ein Graph. Wir definieren $\mathcal{P} \subseteq 2^E$ folgendermaßen: $P \in \mathcal{P}$ genau dann, wenn die Zusammenhangskomponenten des Graphen (V, P) isolierte Knoten oder Pfade sind. Ist $(E, \mathcal{P})$ ein Matroid?

Aufgabe H31 (10 Punkte)

Finden Sie einen minimalen Spannbaum mithilfe des Algorithmus von Kruskal, Prim oder Borůvka[1] für folgenden Graphen. Es reicht aus, wenn Sie den resultierenden, minimalen Spannbaum angeben.



Aufgabe H32 (10 Punkte)

Gegeben sei ein Graph $G = (V, E)$. Wir definieren eine Menge $\mathcal{W} \subseteq 2^E$ mit $W \in \mathcal{W}$ genau dann, wenn eine Kante $e \in W$ exisiert, sodass $(V, W \setminus \{e\})$ ein Wald ist. Zeigen oder widerlegen Sie, dass $(E, \mathcal{W} \cup \{\emptyset\})$ ein Matroid ist.

Aufgabe H33 (10 Punkte)

Das klassische *Rucksackproblem* ist wie folgt definiert. Gegeben seien n Gegenstände welche jeweils ein Gewicht $w_i > 0$ und einen Wert $v_i > 0$ besitzen. Es gilt nun einen Rucksack, welcher ein Gesamtgewicht von maximal $b > 0$ Gewichtseinheiten tragen kann, so zu packen, dass der Wert der enthaltenen Gegenstände maximiert wird. Dabei muss für jeden Gegenstand entschieden werden, ob dieser eingepackt wird oder nicht. Man sucht also eine Menge $I \subseteq \{1, \ldots, n\}$ sodass $\sum_{i \in I} w_i \leq b$, welche $\sum_{i \in I} v_i$ maximiert. Dieses Problem ist im allgemeinen schwierig zu lösen.

Wir betrachten nun eine Abwandlung dieses Problems, das *fraktionale Rucksackproblem.* Wir dürfen nun zusätzlich Gegenstände beliebig zerteilen. Für ein beliebiges $0 \leq \lambda_i \leq 1$ dürfen wir einen Teil des Gegenstands i mit Gewicht $\lambda_i w_i$ und Wert $\lambda_i v_i$ einpacken. Man sucht also eine Sequenz $\lambda_1, \ldots, \lambda_n$ mit $0 \leq \lambda_i \leq 1$ und $\sum_{i=1}^{n} \lambda_i w_i \leq b$, welche $\sum_{i=1}^{n} \lambda_i v_i$ maximiert.

Geben Sie einen effizienten Algorithmus an, welcher das fraktionale Rucksackproblem optimal löst. Argumentieren Sie, warum ihr Algorithmus optimal ist.

[1] `https://dml.cz/bitstream/handle/10338.dmlcz/500114/Boruvka_01-0000-6_1.pdf`