1 Allgemeines

Ein **Graph** G wird beschrieben durch $G = (V, E)$.

Ein Graph $K_n = (V, E)$ ist **vollständig** wenn alle möglichen Kanten für die Knoten V in E enthalten sind.

Ein Knoten x hat den **Grad** $n =: deg(x)$ wenn er genau n Nachbarn hat. Ist $deg(x) = 0$ so ist x ein **isolierter Knoten**. Ist $deg(x) = 1$ so ist x ein **Blatt**.

Ein Graph G ist genau dann **zusammenhängend** wenn jeder Knoten direkt oder indirekt mit jedem anderen Verbunden ist.

Ein Graph $C_n = (V, E)$ ist ein **Kreisgraph** wenn alle Knoten den Grad zwei haben und der Graph zusammenhängend ist.

Eine **Clique** in einem Graphen G ist ein Subgraph dieses Graphen der isomorph zu einem Graphen K_n ist (für ein beliebiges n). Das größtmögliche $n =: \omega(G)$ ist die **Cliquenzahl**

Eine **Stabile Menge** in einem Graphen G ist ein Subgraph dieses Graphen der isomorph zu einem Graphen E_n ist (für beliebiges n). Das größtmögliche $n =: \alpha(G)$ ist die **Stabilitätszahl**.

Der Graph $E_n = (V, \emptyset)$ ist der **leere Graph**.

Ein **Matching** ist eine Auswahl an eindeutigen Zuordnungen von Elementen einer Menge.

Eine **Knotenüberdeckung (Vertex Cover)** für einen Graphen $G = (V, E)$ ist eine Menge $V' \subset V$ wobei es für jedes $e \in E$ ein $v \in V'$ gibt für das $v \in e$.

2 Färbung

Chromatische Zahl ist $\chi(G) := min\{k \in \mathbb{N} : \text{G hat k Färbung}\}$.

Für jeden Graphen $G(V, E)$ gilt $\omega(g) \leq \chi(g)$ und $\frac{|V|}{\alpha(G)} \leq \chi(g)$.

Ein Graph G ist **bipartit** wenn $\chi(G) \leq 2$.

Graph ist bipartit $\Leftrightarrow \nexists$ Kreis ungerader Länge in G

2.1 Greedy (Färbung)

Für alle Knoten $v \in V$ aus $G = (V, E)$: nehme ausschließlich bekannte Knotenfärbungszahlen aller verbundenen Knoten in eine Menge N, Knotenfärbung von v ist die kleinste Zahl aus den natürlichen Zahlen ohne N.

3 Planarität

Ein Graph ist genau dann planar, wenn er keine Unterteilung des K_5 oder des $K_{3,3}$ besitzt

Eulersche Polyederformel: (R sei die Anzahl der Regionen inkl. Außenregion)

$$|V| - |E| + |R| = 2$$

Weiterhin: $|E| \leq 3n - 6$

$2|E| \leq 3|R|$

4 Netzwerke

4.1 Minimaler Spannbaum

Kruskal, Gewichte aufsteigend betrachten und nur inkludieren wenn neuer Knoten eingebunden wird oder Partitionen verbunden werden.

4.2 Floyd, Dajkstra, Kruskal

4.2.1 Floyd, Kürzeste Pfade

$d(i, j)$ initialisieren. Für alle $k \in 1$ bis n: Für alle Knotenpaare i, j sei $d(i, j) = min(d(i, j), d(i, k) + d(k, j))$.

4.2.2 Dajkstra, Kürzeste Pfade bei nichtnegativen Kanten

Nehme Knoten mit minimaler Distanz zum Startknoten - hinzufügen. (Nachbarknoten ggf. updaten.)

4.2.3 Kruskal, Minimaler Spannbaum

Aus unbenutzten Kanten die kürzeste wählen, die mit den gewählten keinen Kreis bildet. Wiederholen.

5 Komplexitäten (vereinfacht)

5.1 Abschätzung nach oben

$f \in O(g) \Leftrightarrow \lim_{x\to\infty} \left|\frac{f(x)}{g(x)}\right| < \infty$ $\qquad$ $f(n) = O(g(n)) \Leftrightarrow f(n) \leq c \cdot g(n)$

5.2 Abschätzung

$$f \in \Theta(g) \Leftrightarrow 0 < \lim_{x\to\infty} \left|\frac{f(x)}{g(x)}\right| < \infty$$

$$f(n) = \Theta(g(n)) \Leftrightarrow c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

5.3 Abschätzung nach unten

$f \in \Omega(g) \Leftrightarrow 0 < \lim_{x\to\infty} \left|\frac{f(x)}{g(x)}\right|$ $\qquad$ $f(n) = \Omega(g(n)) \Leftrightarrow f(n) \geq c \cdot g(n)$

6 Matroide

TODO

7 Linear Programming

7.1 LP

7.2 Dual Problem

8 SAT

k-SAT mit $k \geq 3$ sind NP-schwer. $SAT \leq 3 - SAT \leq Clique$

9 Misc

$\sum\limits_{v\in V} deg(v) = 2 \cdot |E|$

Ungewichtetes Scheduling: Tasks mit größter Penalty auf ihre Deadline oder davor setzen, abwärts.