

Algorithmische Graphentheorie

6. Übungsblatt

Jasper Gude

Pia Röttgers

7. Juni 2026

/ 20

Aufgabe 1 – Wurzelspannbäume und ungerichtete Bäume

Sei $G = (V, E)$ ein gerichteter Graph und s ein ausgezeichnete Knoten. Zeigen Sie:

Der gerichtete Graph G ist ein s -Wurzelspannbaum \iff Der G zugrunde liegende ungerichtete Graph G' ist ein Baum, für jedes $v \in V \setminus \{s\} : \text{indeg}(v) = 1$ und $\text{indeg}(s) = 0$.

\implies Wenn der Graph G ein s -Wurzelspannbaum ist, dann erfüllt er die Eigenschaften eines s -Wurzelbaums, also G azyklisch, $v \in V \setminus \{s\} : \text{indeg}(v) = 1$ und $\text{indeg}(s) = 0$. Der zugrunde liegende ungerichtete Graph G' muss also auch azyklisch, also ein Baum, sein.

\impliedby Fall 1: G' ist kein Baum.

Dann existiert in G entweder ein gerichteter Kreis, was die erste s -Wurzelbaum-Eigenschaft verletzt, oder ein Knoten u hat $\text{indeg}(u) > 1$.

Fall 2: Es existiert ein Knoten $v \in V \setminus \{s\}$ mit $\text{indeg}(v) > 1$.

Das verletzt die zweite s -Wurzelbaum-Eigenschaft. Also kann G kein s -Wurzelspannbaum sein.

Fall 3: Der Knoten s hat $\text{indeg}(s) > 0$.

Das verletzt die dritte s -Wurzelbaum-Eigenschaft. Also kann G kein s -Wurzelspannbaum sein.

/ 4

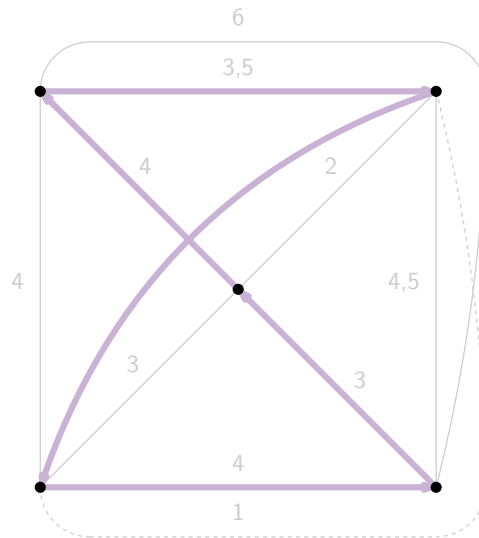


Abbildung 1: Eine kleinere TSP-Tour, die von CRISTOFIDES nicht berechnet werden kann.

Aufgabe 2 – Christofides' Algorithmus

- a) Siehe Abbildung 2. / 4
- b) Siehe Abbildung 1. Diese Tour kann nicht von CHRISTOFIDES berechnet werden, denn egal mit welcher Kante in welche Richtung gestartet wird, nimmt CRISTOFIDES immer eine Kante in die Tour, die nicht in Abbildung 1 ist. / 1

Aufgabe 3 – Matchings in allgemeinen Graphen

a)

`MATCHINGS($G = (V, E)$)`

`$M = \emptyset$`

`visited = Array von False der Größe $|V|$ // Markiert gematchte Knoten`

for e in E do

if $\neg \text{visited}[u] \wedge \neg \text{visited}[v]$ then

`$M = M \cup \{e\}$`

`visited[u] = True`

`visited[v] = True`

return M

Angenommen M wäre erweiterbar (d.h. es gibt eine Kante $\{u, v\}$ mit $u, v \notin V(M)$). Dann wurden sowohl u als auch v während des Algorithmus beim Durchlaufen der Kante $\{u, v\}$ als frei angesehen und wäre somit der Menge M hinzugefügt worden. Das ist ein Widerspruch.

Laufzeit:

Initialisierung des Arrays: $\mathcal{O}(V)$

Schleife: Jede Kante wird genau einmal betrachtet ($\mathcal{O}(V)$) und die Überprüfung und Markierung passieren in $\mathcal{O}(1)$. Also insgesamt $\mathcal{O}(E)$

Das ergibt eine Gesamtlaufzeit von $\mathcal{O}(V + E)$

/ 3

b)

/ 2

c) Sei M das vom Algorithmus berechnete maximale Matching und M^* das optimale maximale Matching.

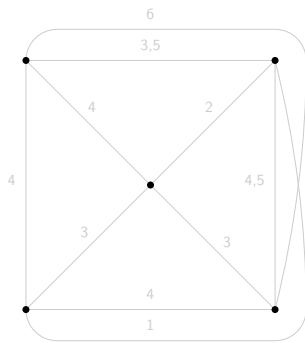
Für jede Kante $e^* = \{u, v\} \in M^*$ gilt: mindestens einer der beiden Knoten u oder v muss über eine Kante aus M abgedeckt werden, sonst wäre der M nicht nicht-erweiterbar.

Eine Kante aus M hat genau zwei Endknoten und kann daher höchstens zwei verschiedene Kanten aus M^* „blockieren“. Da alle Kanten in M^* disjunkt sind kann man daraus folgern:

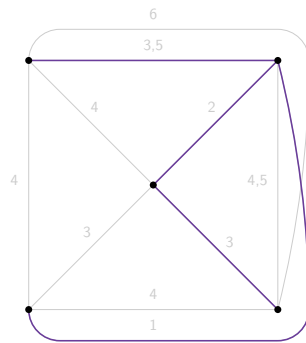
$$|M^*| \leq 2 \cdot |M| \implies |M| \geq \frac{1}{2} |M^*|$$

Somit ist der Algorithmus aus der Teilaufgabe a) eine $1/2$ Approximation für ein optimales Matching.

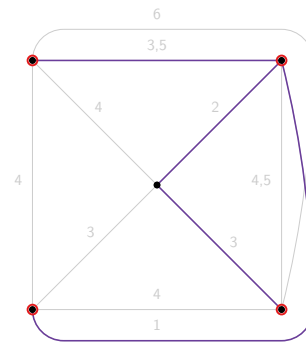
/ 3



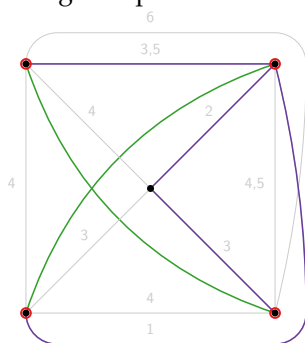
(a) Der gewichtete, vollständige Graph G .



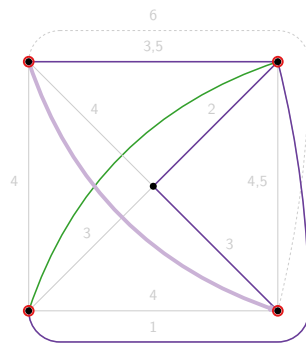
(b) Minimalen Spannbaum finden.



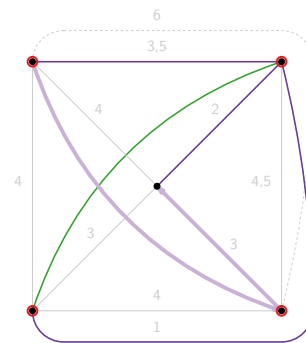
(c) Knoten U mit ungeraden Graden finden.



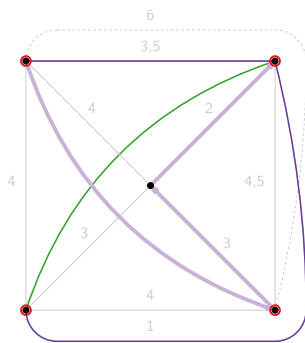
(d) Minimales, perfektes Matching auf $G[U]$ finden.



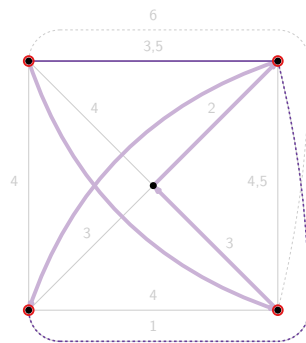
(e) Eulertour konstruieren.



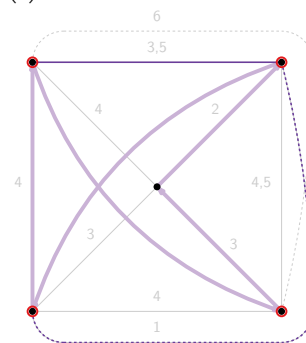
(f) Eulertour konstruieren.



(g) Eulertour konstruieren.



(h) Eulertour konstruieren.



(i) Schon besuchte Knoten überspringen. TSP vollständig.

Abbildung 2: Cristofides' Algorithmus

d) Zielfunktion:

$$\arg \min \sum_{e \in E} x_e \geq 1$$

Entscheidungsvariablen: für jede Kante $e \in E$:

$$x_e \in \{0, 1\} \quad \forall e \in E$$

Die Variable nimmt den Wert 1 an, wenn die Kante im Matching M ist und 0 falls sie das nicht ist.

Nebenbedingungen:

1. Matching: Jeder Knoten darf von maximal einer Matching-Kante berührt werden

$$\forall v \in V: \sum_{e \in \delta(v)} x_e \leq 1$$

$\delta(v)$ ist die Menge aller Kanten welche an v anliegen

2. Nicht-Erweiterbarkeit: Für jede Kante $\{u, v\}$ muss die Summe der Matching-Kanten an u und v mindestens 1 sein

$$\forall \{u, v\} \in E: \sum_{e \in \delta(u)} x_e + \sum_{e' \in \delta(v)} x'_{e'} \geq 1$$

$\delta(v)$ ist die Menge aller Kanten welche an v anliegen. Äquivalent für $\delta(u)$.

/ 3

e)

/ 2
