Tutorium 11

All-Pairs Shortest-Path: Floyd-Warshall-Algorithmus (Tripel)
↳ kürzeste Wege

Input: Digraph $G = (V, E, c)$ mit $|V| = n$ ohne Kreis negativer Länge, Kostenmatrix $C(G)$, Vorgängermatrix $P(G)$

begin
  for $j = 1$ **to** n
    for $i = 1$ **to** n
      for $k = 1$ **to** n
        if $c_{ij} + c_{jk} < c_{ik}$
          $c_{ik} := c_{ij} + c_{jk}$, $p_{ik} := p_{jk}$
        end
      end
    end
  end
end

Output: $C(G)$ enthält $D(G)$ und $P(G)$ enthält $R(G)$

Kruskal-Algorithmus
↳ minimaler spannender Baum

Input: Bewerteter zusammenhängender ungerichteter Graph $G = [V, E, c]$ mit $|V| = n$ und sortierter Kantenmenge $e_1, e_2, \dots, e_m$ mit $c(e_1) \le c(e_2) \le \dots \le c(e_m)$

begin
  $E' := \emptyset$, $i := 1$
  while $|E'| < n - 1$
    if $T = [V, E' \cup \{e_i\}]$ enthält keinen Kreis
      $E' := E' \cup \{e_i\}$
    end
    $i := i + 1$
  end
end

Output: $T = [V, E']$ ist minimaler spannender Baum von G

Bestimmung minimaler 1-Bäume-Algorithmus

Input: Bewerteter zusammenhängender ungerichteter Graph $G = [V, E, c]$, Knoten $a \in V$ mit $\deg(a) \ge 2$ und $G' = [V', E']$ mit $V' := V \setminus \{a\}$, $E' := E \setminus \{e \in E \mid a \in e\}$ zusammenhängend

begin
  Bestimme einen minimalen spannenden Baum $T = [V', \tilde{E}]$ von G'
  Bestimme $e_1, e_2 \in E$, so dass gilt

$$c(e_1) := \min_{\{e \in E \mid a \in e\}} \{c(e)\} \text{ und } c(e_2) := \min_{\{e \in E \mid a \in e\} \setminus \{e_1\}} \{c(e)\}$$

  $\tilde{E} := \tilde{E} \cup \{e_1, e_2\}$
end

Output: $T = [V, \tilde{E}]$ ist minimaler 1-Baum von G mit Knoten a

Fleury's Algorithmus

↳ Eulerscher Kreis

Input: Eulerscher Graph $G = [V, E]$, Startknoten $a \in V$

Brücke = Kante, die einen Graph zusammenhält
ohne Brücke => 2 Teilgraphen

begin
- $C_E := \emptyset, v := a$
- **while** $E \neq \emptyset$
  - Bestimme $e = [v, i] \in E$ mit $i \in NB(v)$; dabei sind zuerst Kanten zu wählen, die keine Brücke sind
  - $v := i$, $C_E := C_E \cup \{e\}$, $E := E \setminus \{e\}$
- **end**

end

Output: Eulerscher Kreis C_E

Chinese Postman Problem-Algorithmus

Input: Bewerteter zusammenhängender ungerichteter Graph $G = [V, E, c]$

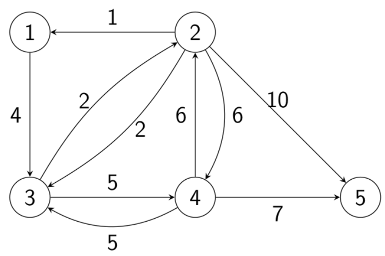
begin
- $V_0 := \{i \in V \mid \deg(i) \text{ ungerade}\}$ $(\Rightarrow |V_0| \text{ gerade})$
- **forall** $v, w \in V_0$
  - Bestimme einen kürzesten Weg von v nach w in G
- **end**
- Bestimme einen vollständigen Graphen K mit Knotenmenge V_0 und einer Kantenbewertung, die der Länge des kürzesten Weges zwischen den Knoten entspricht
- Bestimme ein perfektes Matching M in K mit minimaler Summe der Kantenbewertungen
- **forall** $e \in M$
  - Verdopple in G die Kanten, die zum kürzesten Weg P gehören, der der Kante e zugeordnet ist
- **end**
- Bestimme einen Eulerschen Kreis C_E im erweiterten Graphen von G

end

Output: Optimale Chinese Postman Tour C_E

Aufgabe 1

Gegeben sei der folgende bewertete, gerichtete Graph:



Input: Digraph $G = (V, E, c)$ mit $|V| = n$ ohne Kreis negativer Länge, Kostenmatrix $C(G)$, Vorgängermatrix $P(G)$

begin
  for $j = 1$ **to** n
    for $i = 1$ **to** n
      for $k = 1$ **to** n
        if $c_{ij} + c_{jk} < c_{ik}$
          $c_{ik} := c_{ij} + c_{jk}$, $p_{ik} := p_{jk}$
        end
      end
    end
  end
end

Output: $C(G)$ enthält $D(G)$ und $P(G)$ enthält $R(G)$

Setzen Sie den Tripel-Algorithmus (Algorithmus von Floyd-Warshall) ein, um kürzeste Wege für alle Knotenpaare zu bestimmen.

A1) Initialisierung

$$C(G) = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ \infty & 2 & 0 & 5 & \infty \\ \infty & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \qquad R(G) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 0 & 3 & 3 & 3 & 0 \\ 0 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

1. Iteration für $j=1$:

- keine Veränderung, da über Knoten 1 nur ein Weg von Knoten 2 nach Knoten 3 mit Länge $5 > c_{23} = 2$ existiert.

$$C(G) = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ \infty & 2 & 0 & 5 & \infty \\ \infty & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \qquad R(G) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 0 & 3 & 3 & 3 & 0 \\ 0 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

2. Iteration für $j=2$

$$C(G) = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \qquad R(G) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 3 & 3 & 2 \\ 2 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

3. Iteration für $j=3$

$$C(G) = \begin{bmatrix} 0 & 6 & 4 & 9 & 16 \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \qquad R(G) = \begin{bmatrix} 1 & 3 & 1 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 3 & 3 & 2 \\ 2 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

4. und 5. Iteration für $j=4$ bzw. $j=5$ keine Veränderung

$\Rightarrow$ kürzeste Wege lassen sich aus $R(G)$ bestimmen.
Beispielsweise ergibt sich aufgrund von $r_{15}=2$, $r_{12}=3$ und $r_{13}=1$ der Weg $(1,3,2,5)$ als kürzeste Weg von 1 nach 5.

Verständliches Vorgehen Floyd-Warshall:

$$D_1 = C(G) = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ \infty & 2 & 0 & 5 & \infty \\ \infty & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

Betrachte Spalte und Zeile i zusammen. Schauen, ob es bessere Wege über andere Kanten gibt.

$\Rightarrow$ Für $i/j=1$ keine Verbesserung

$$D_2 = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ \infty & 2 & 0 & 5 & \infty \\ \infty & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \Rightarrow D_3 = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} 0 & \infty & 4 & \infty & \infty \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \Rightarrow D_4 = \begin{bmatrix} 0 & 6 & 4 & 9 & 16 \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

$$D_4 = \begin{bmatrix} 0 & 6 & 4 & 9 & 16 \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

keine Verbesserung $\Rightarrow D_5 = D_4$

$$D_5 = \begin{bmatrix} 0 & 6 & 4 & 9 & 16 \\ 1 & 0 & 2 & 6 & 10 \\ 3 & 2 & 0 & 5 & 12 \\ 7 & 6 & 5 & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

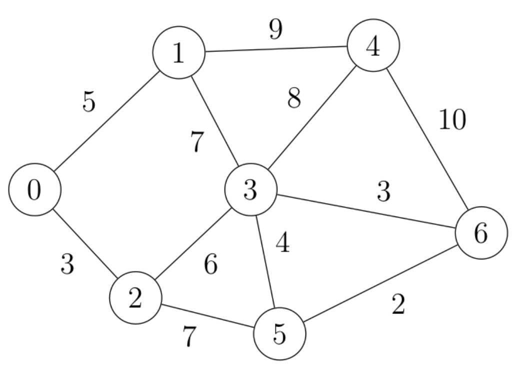
keine Verbesserung $\Rightarrow$ fertig

Aufgabe 2

In einem ländlichen Raum sind die meisten Überlandleitungen für die Stromversorgung schon seit den 1950er Jahren in Betrieb. Da die Leitungsmasten anfangen morsch zu werden, soll das Leitungsnetz in den nächsten Jahren neu gebaut werden. Im konkreten Fall geht es darum, sechs Ortschaften mit dem regionalen Umspannwerk neu zu verbinden. Die bestehenden Leitungen müssen hierbei komplett ersetzt werden.

In einem ersten Schritt hat der Energieversorger bereits ermittelt, zwischen welchen Orten, einschließlich dem Umspannwerk, Überlandleitungen in Frage kommen und wie hoch die Kosten für die Errichtung wären.

Die Lage der sechs Ortschaften und des Umspannwerks (Knoten 0), sowie die möglichen Überlandverbindungen samt Baukosten in Millionen € sind im folgenden Graphen angegeben:



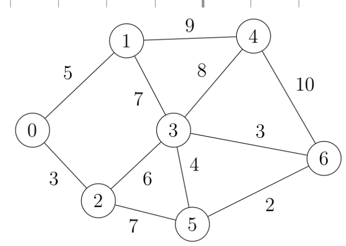
a) Bestimmen Sie, welche der Überlandleitungen errichtet werden sollen, so dass alle Orte mit dem Umspannwerk verbunden sind und die Gesamtkosten dafür minimal sind. Wie hoch sind die Kosten?

A2a) Kruskal-Algorithmus:

Input: Bewerteter zusammenhängender ungerichteter Graph $G = [V, E, c]$ mit $|V| = n$ und sortierter Kantenmenge $e_1, e_2, \dots, e_m$ mit $c(e_1) \le c(e_2) \le \dots \le c(e_m)$

begin
  $E' := \emptyset,\ i := 1$
  while $|E'| < n - 1$
    if $T = [V, E' \cup \{e_i\}]$ enthält keinen Kreis
      $E' := E' \cup \{e_i\}$
    end
    $i := i + 1$
  end
end

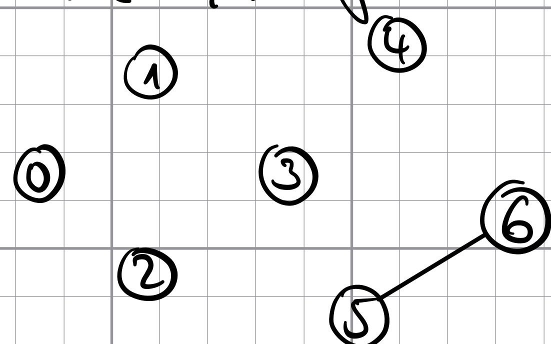
Output: $T = [V, E']$ ist minimaler spannender Baum von G



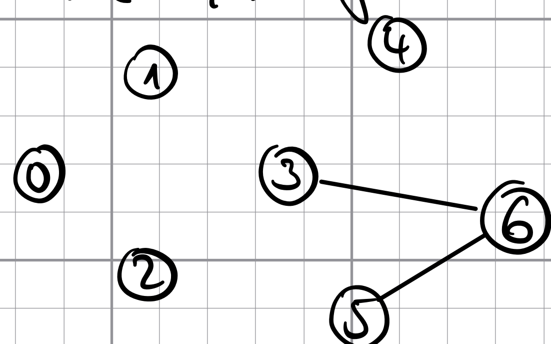
Sortierung der Kanten nach Gewichten:

(5,6); (3,6); (0,2); (3,5); (0,1); (2,3); (1,3); (2,5); (3,4); (1,4); (4,6)

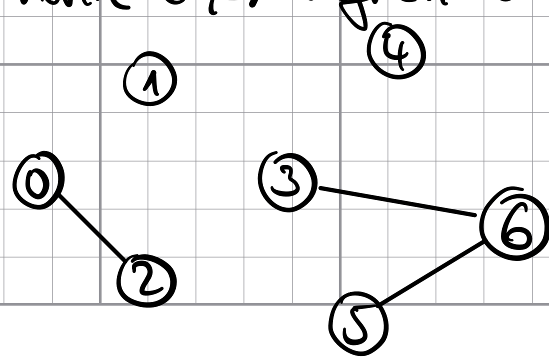
Iteration 1: Kante (5,6) aufnehmen



Iteration 2: Kante (3,6) aufnehmen

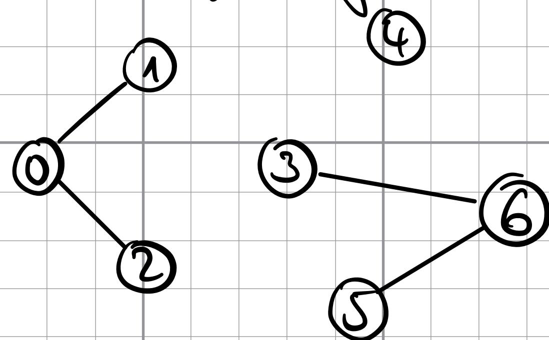


Iteration 3: Kante (0,2) aufnehmen

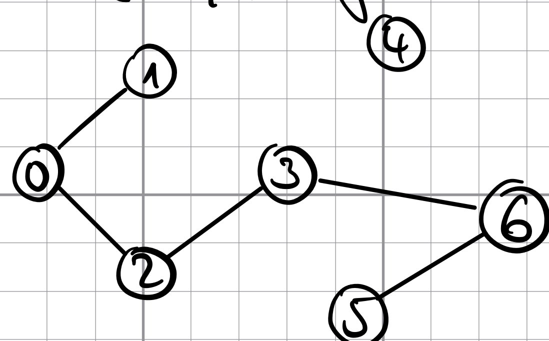


Iteration 4: Kante (3,5) nicht aufnehmen, sonst Kreis

Iteration 5: Kante (0,1) aufnehmen



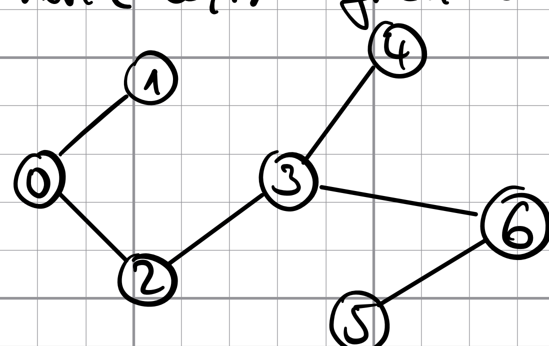
Iteration 6: Kante (2,3) aufnehmen



Iteration 7: Kante (1,3) nicht aufnehmen, sonst Kreis

Iteration 8: Kante (2,5) nicht aufnehmen, sonst Kreis

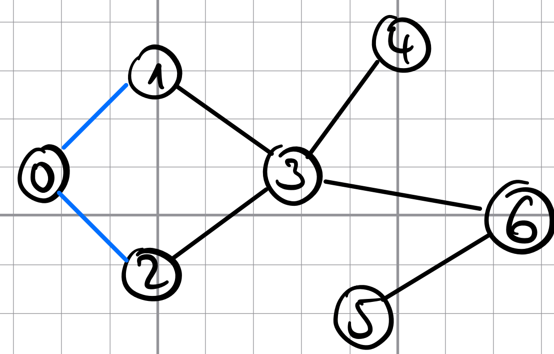
Iteration 9: Kante (3,4) aufnehmen



=> alle Knoten verbunden → Stopp! ▷ Kosten 27 M€

b) Das Umspannwerk in Knoten 0 soll nun in einem Kreis enthalten sein. Wie müssen die Überlandleitungen nun errichtet werden und wie hoch sind die Kosten?

Bestimme zunächst MST ohne Knoten 0

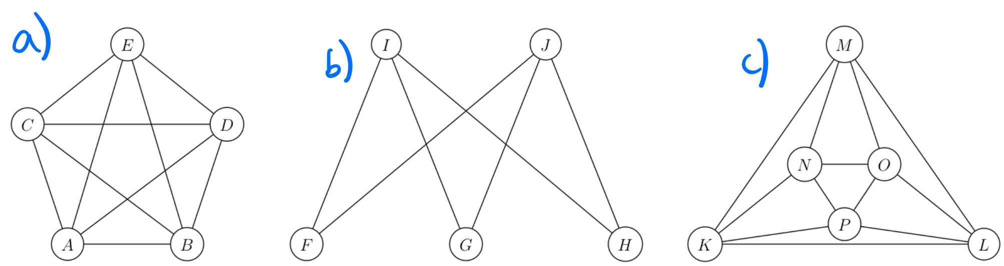


Erzeuge nun den 1-Baum zu Knoten 0 durch Hinzufügen der beiden billigsten zu Knoten 0 inzidenten Kanten

▷ Kosten: 34 M €

Aufgabe 3

Welche der folgenden Graphen sind Eulersche Graphen, welche nicht? Bestimmen Sie für die Eulerschen Graphen einen Eulerschen Kreis mit Hilfe von Fleury's Algorithmus.



A3)

Input: Eulerscher Graph $G = [V, E]$, Startknoten $a \in V$

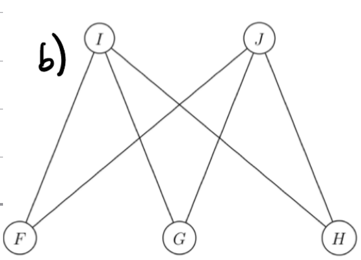
begin
  $C_E := \emptyset, v := a$
  while $E \neq \emptyset$
    Bestimme $e = [v, i] \in E$ mit $i \in NB(v)$; dabei sind zuerst Kanten zu wählen, die keine Brücke sind
    $v := i$, $C_E := C_E \cup \{e\}$, $E := E \setminus \{e\}$
  end
end

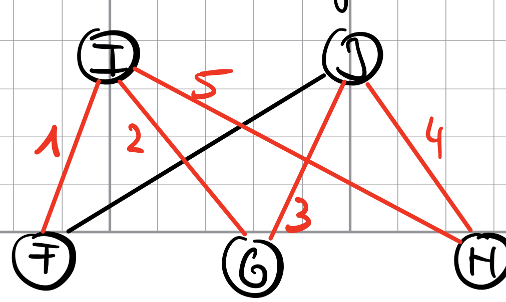
Output: Eulerscher Kreis C_E

Euler-Graph:
alle Knoten geraden Knotengrad

a) Euler Kreis: A-B-D-E-C-A-D-C-B-E-A

b) nicht Eulersch, da Knoten mit ungeradem Knotengrad



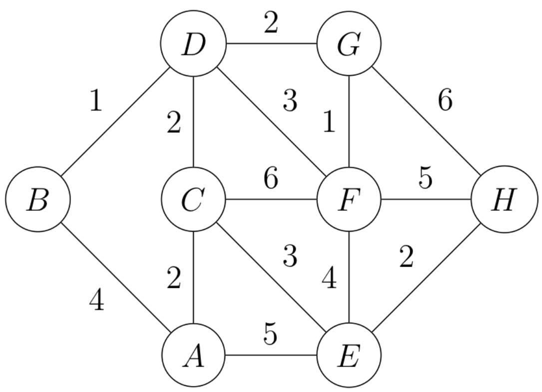


-> gefärbte Kante

c) Euler-Kreis: K-L-M-K-N-M-O-N-P-O-L-P-K

Aufgabe 4

Herr Müller ist Angestellter einer kleinen Stadt und im Winter für die Räumung der Straßen zuständig. Er hat allerdings nur ein Räumfahrzeug zur Verfügung und muss deshalb die Arbeit alleine durchführen. Zu Beginn des Winters überlegt er sich nun, wie er die Straßen der Stadt am besten abfahren kann, so dass jede Straße mindestens einmal geräumt wird, er am Ende der Tour wieder in die Fahrzeughalle zurückkommt und die insgesamt zurückgelegte Strecke minimiert wird. Der folgende Graph gibt das Straßennetz der Stadt an. Dabei entspricht die Kantenbewertung den Entfernungen zwischen den einzelnen Kreuzungspunkten und der Knoten A stellt den Standort der Fahrzeughalle dar.



a) Bestimmen Sie eine optimale Chinese Postman Tour in diesem Graphen, die in Knoten A beginnt und endet. Verwenden Sie dazu das in der Vorlesung angegebene Verfahren und geben Sie die Tour als Folge von Knoten an.

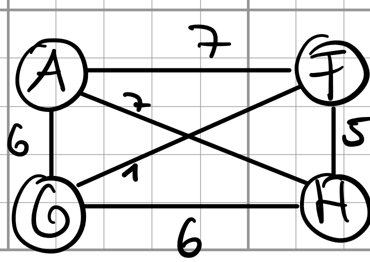
Input: Bewerteter zusammenhängender ungerichteter Graph $G = [V, E, c]$

begin
$V_0 := \{i \in V \mid \deg(i) \text{ ungerade}\}$ $(\Rightarrow |V_0| \text{ gerade})$
forall $v, w \in V_0$
Bestimme einen kürzesten Weg von v nach w in G
end
Bestimme einen vollständigen Graphen K mit Knotenmenge V_0 und einer Kantenbewertung, die der Länge des kürzesten Weges zwischen den Knoten entspricht
Bestimme ein perfektes Matching M in K mit minimaler Summe der Kantenbewertungen
forall $e \in M$
Verdopple in G die Kanten, die zum kürzesten Weg P gehören, der der Kante e zugeordnet ist
end
Bestimme einen Eulerschen Kreis C_E im erweiterten Graphen von G
end

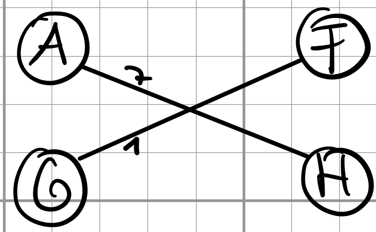
Output: Optimale Chinese Postman Tour C_E

A4) a) Knoten mit ungeradem Knotengrad: $V_0 = \{A, F, G, H\}$

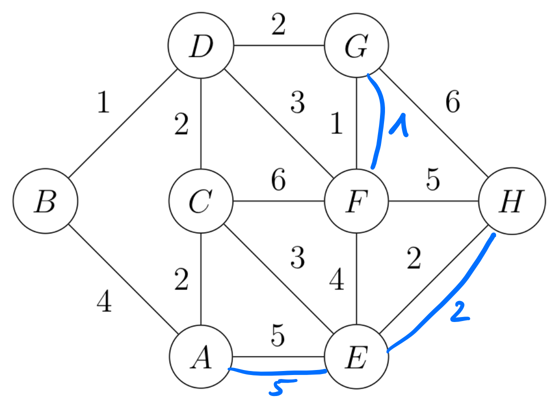
Kürzeste Wege in V_0 \ Vollständiger Graph der Knoten V_0



Kostenminimales perfektes Matching



Erweiterter Graph:



Eulerscher Kreis:

A-B-D-C-A-E-C-F-D-G-F-H-G-F-E-H-E-A

b) Wie lang ist somit die Strecke, die Herr Müller insgesamt zurücklegen muss?

b) Länge der Tour: 54

c) Welche Strecken bzw. Straßen muss Herr Müller dabei zweimal fahren?

c) Doppelte Straßen: A-E, E-H, F-G

nächstes Tutorium:

- Hamilton Graph
- Knotennetzpläne
- Breitensuche

Online-Test

- Mehrkriterielle lineare Optimierung (Pareto-Optimalität)
- Totale Unimodularität
- Transportprobleme