Propädeutikum Diskrete Mathematik

TSP Approximationen

Prof. Dr. R. Hemmecke

B.Sc. W.F. Riedl   Dipl-Math. M. Silbernagl

Technische Universität München

WS 2013/14

Polynomielle und exponentielle Laufzeit

	Problemgröße n					
	10		20		50	
n	0.00001	s	0.00002	s	0.00005	s
n^2	0.0001	s	0.0004	s	0.0025	s
n^5	0.1	s	3.2	s	5.2	min
2^n	0.001	s	1.0	s	$35,7$	Jahre
5^n	0.059	s	58	min	$2 \cdot 10^{11}$	Jahre

Polynomielle und exponentielle Laufzeit

	Problemgröße n		
	10	20	50
n	0.00001 s	0.00002 s	0.00005 s
n^2	0.0001 s	0.0004 s	0.0025 s
n^5	0.1 s	3.2 s	5.2 min
2^n	0.001 s	1.0 s	35, 7 Jahre
5^n	0.059 s	58 min	$2 \cdot 10^{11}$ Jahre

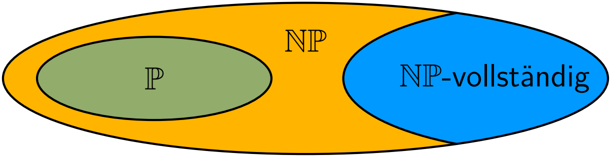
$f(n)$	cn	cn^k	2^n
$f(n+1)$	$c(n+1) = cn + c$	$c(n+1)^k = cn^k + \mathcal{O}\left(n^{k-1}\right)$	$2^{n+1} = 2 \cdot 2^n$
$f(2n)$	$2 \cdot cn$	$c(2n)^k = 2^k cn^k$	$2^{2n} = (2^n)^2$

$\mathbb{P}$ und $\mathbb{NP}$

Min. Spannbaum $<$ Kürzeste Wege $\ll$ Rundreise/ TSP $<$ Färbung

$\mathbb{P}$: in polynomieller Zeit lösbar

$\mathbb{NP}$-vollständig: nur exp. Algorithmen bekannt



$\mathbb{P} = \mathbb{NP}$?

Traveling Salesman Problem (TSP)

TSP

Sei $G = (V, E, l)$ ein gewichteter Graph mit $l : E \to \mathbb{R}_{\geq 0}$. Ein Hamiltonkreis in G heißt TSP-Tour. Eine optimale TSP-Tour in G ist eine TSP-Tour mit kürzestmöglicher Länge bzgl. l.

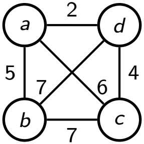
Traveling Salesman Problem (TSP)

TSP

Sei $G = (V, E, l)$ ein gewichteter Graph mit $l : E \to \mathbb{R}_{\geq 0}$. Ein Hamiltonkreis in G heißt TSP-Tour. Eine optimale TSP-Tour in G ist eine TSP-Tour mit kürzestmöglicher Länge bzgl. l.
Genügt die Gewichtsfunktion l der Dreiecksungleichung, d.h.

$$l(\{x, y\}) + l(\{y, z\}) \geq l(\{x, z\}) \qquad \forall x, y, z \in V,$$

so spricht man vom metrischen TSP.



Metrischer Graph

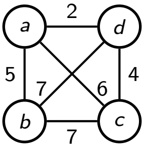
Traveling Salesman Problem (TSP)

TSP

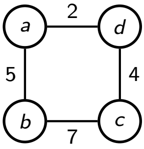
Sei $G = (V, E, l)$ ein gewichteter Graph mit $l : E \to \mathbb{R}_{\geq 0}$. Ein Hamiltonkreis in G heißt TSP-Tour. Eine optimale TSP-Tour in G ist eine TSP-Tour mit kürzestmöglicher Länge bzgl. l.
Genügt die Gewichtsfunktion l der Dreiecksungleichung, d.h.

$$l(\{x, y\}) + l(\{y, z\}) \geq l(\{x, z\}) \qquad \forall\, x, y, z \in V,$$

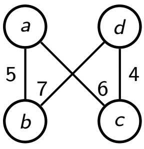
so spricht man vom metrischen TSP.



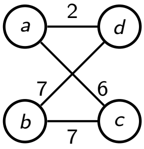
Metrischer Graph



Länge 18, optimal



Länge 22



Länge 22

Nächster-Nachbar-Heuristik (NN)

Idee: Gehe zum nächstgelegenen unbesuchten Nachbarn.

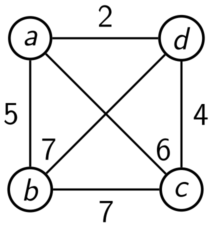
Input : Vollständiger gew. Graph $G = (V, E, l)$ mit $l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;
$W \leftarrow \{v_1\}$;
while $W \neq V$ **do**
  $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l\left(\left\{v_{|W|}, y\right\}\right)$;
  $W \leftarrow W \cup \left\{v_{|W|+1}\right\}$;
end

Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

```
Wähle beliebigen Startknoten v_1 ∈ V;
W ← {v_1};
while W ≠ V do
    v_{|W|+1} ← Knoten y ∈ V \ W mit
    minimalem l({v_{|W|}, y});
    W ← W ∪ {v_{|W|+1}};
end
```



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;

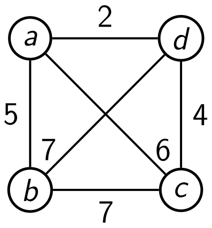
$W \leftarrow \{v_1\}$;

while $W \neq V$ **do**

  $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;

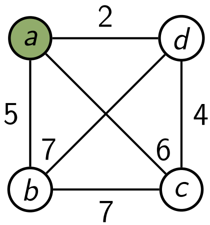
  $W \leftarrow W \cup \{v_{|W|+1}\}$;

end



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

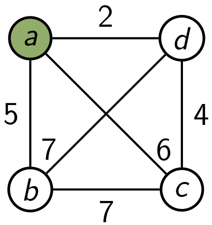
```
Wähle beliebigen Startknoten v_1 ∈ V;
W ← {v_1};
while W ≠ V do
    v_{|W|+1} ← Knoten y ∈ V \ W mit
    minimalem l({v_{|W|}, y});
    W ← W ∪ {v_{|W|+1}};
end
```



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

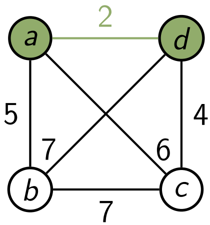
Wähle beliebigen Startknoten $v_1 \in V$;
$W \leftarrow \{v_1\}$;
while $W \neq V$ **do**
    $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;
    $W \leftarrow W \cup \{v_{|W|+1}\}$;
end



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;
$W \leftarrow \{v_1\}$;
while $W \neq V$ **do**
    $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;
    $W \leftarrow W \cup \{v_{|W|+1}\}$;
end



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;

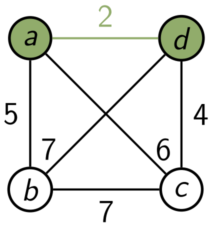
$W \leftarrow \{v_1\}$;

while $W \neq V$ **do**

  $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;

  $W \leftarrow W \cup \{v_{|W|+1}\}$;

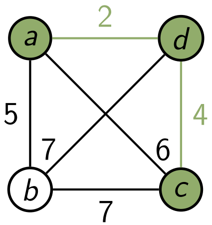
end



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;
$W \leftarrow \{v_1\}$;
while $W \neq V$ **do**
    $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;
    $W \leftarrow W \cup \{v_{|W|+1}\}$;
end



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \dots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;

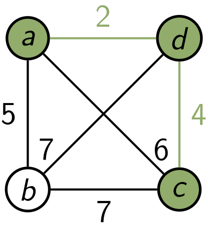
$W \leftarrow \{v_1\}$;

while $W \neq V$ **do**

  $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;

  $W \leftarrow W \cup \{v_{|W|+1}\}$;

end



Input : Vollst. gew. Graph $G = (V, E, l)$ mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;

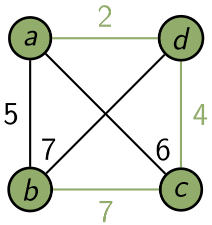
$W \leftarrow \{v_1\}$;

while $W \neq V$ **do**

  $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;

  $W \leftarrow W \cup \{v_{|W|+1}\}$;

end



Input : Vollst. gew. Graph $G = (V, E, l)$
mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Wähle beliebigen Startknoten $v_1 \in V$;

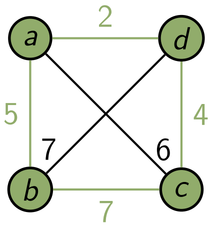
$W \leftarrow \{v_1\}$;

while $W \neq V$ **do**

  $v_{|W|+1} \leftarrow$ Knoten $y \in V \setminus W$ mit minimalem $l(\{v_{|W|}, y\})$;

  $W \leftarrow W \cup \{v_{|W|+1}\}$;

end



Heuristik mit minimalen Spannbäumen (MST)

Input : Vollständiger gew. Graph $G = (V, E, l)$ mit $l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Alg. von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \mathrm{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \text{Pop}(S)$;

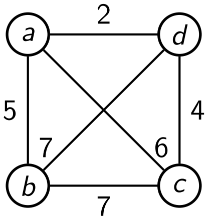
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

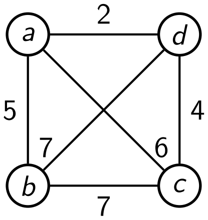
end



Input : Vollst. gew. Graph $G = (V, E)$ mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \dots, v_n)$

```
Berechne minimalen Spannbaum
T = (V, E') in G (Algorithmus von Prim);
Finde geschl. Kantenzug (x_1, x_2, ..., x_l), der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel S = (x_1, x_2, ..., x_l);
W ← ∅;
while W ≠ V do
    y ← Pop(S);
    if y ∉ W then
        W ← W ∪ {y};
        v_|W| ← y;
    end
end
```



Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \dots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \dots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \dots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \mathrm{Pop}(S)$;

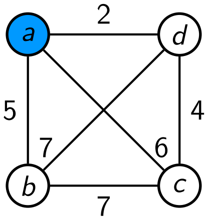
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

end



Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \mathrm{Pop}(S)$;

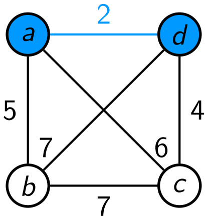
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

end



Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \mathrm{Pop}(S)$;

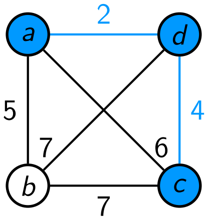
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

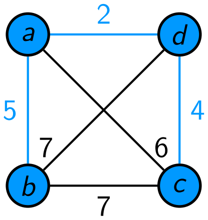
  end

end



Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \mathrm{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end



Input : Vollst. gew. Graph $G = (V, E)$ mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

    $y \leftarrow \mathrm{Pop}(S)$;

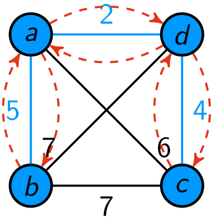
    if $y \notin W$ **then**

        $W \leftarrow W \cup \{y\}$;

        $v_{|W|} \leftarrow y$;

    end

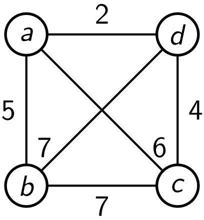
end



$E' = \{\{a, b\}, \{a, d\}, \{c, d\}\}$

$S = (b, a, d, c, d, a, b)$

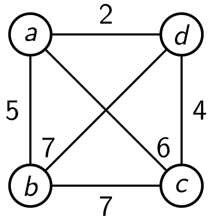
Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \dots, v_n)$
Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \dots, x_l)$, der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel $S = (x_1, x_2, \dots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \text{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end



$S = (b, a, d, c, d, a, b)$

$y =$

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$
Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \text{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end



$S = (a, d, c, d, a, b)$
$y = b$

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \dots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \dots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \dots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \text{Pop}(S)$;

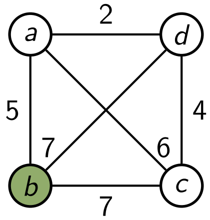
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

end

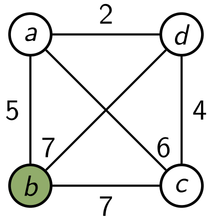


$S = (a, d, c, d, a, b)$

$y = b$

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \mathrm{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end



$S = (d, c, d, a, b)$
$y = a$

Input : Vollst. gew. Graph $G = (V, E)$ mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \text{Pop}(S)$;

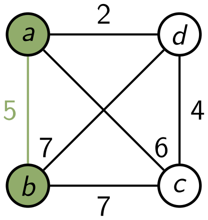
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

end



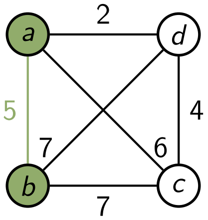
$S = (d, c, d, a, b)$

$y = a$

Input : Vollst. gew. Graph $G = (V, E)$ mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

```
Berechne minimalen Spannbaum T = (V, E') in G (Algorithmus von Prim);
Finde geschl. Kantenzug (x_1, x_2, ..., x_l), der jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel S = (x_1, x_2, ..., x_l);
W ← ∅;
while W ≠ V do
    y ← Pop(S);
    if y ∉ W then
        W ← W ∪ {y};
        v_{|W|} ← y;
    end
end
```

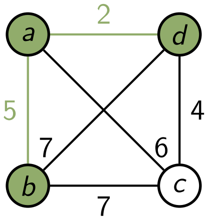


$S = (c, d, a, b)$

$y = d$

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \text{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end



$S = (c, d, a, b)$
$y = d$

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \dots, v_n)$

Berechne minimalen Spannbaum
$T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \dots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \dots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \mathrm{Pop}(S)$;

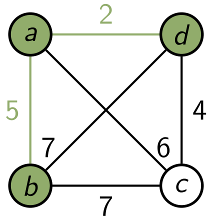
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

end



$S = (d, a, b)$

$y = c$

Input : Vollst. gew. Graph $G = (V, E)$ mit $l : E \to \mathbb{R}_{\geq 0}$

Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);

Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;

$W \leftarrow \emptyset$;

while $W \neq V$ **do**

  $y \leftarrow \text{Pop}(S)$;

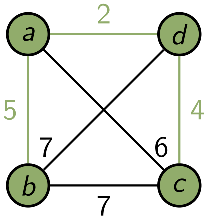
  if $y \notin W$ **then**

    $W \leftarrow W \cup \{y\}$;

    $v_{|W|} \leftarrow y$;

  end

end

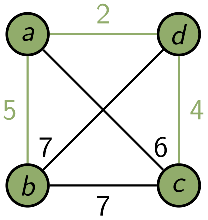


$S = (d, a, b)$

$y = c$

Input : Vollst. gew. Graph $G = (V, E)$ mit
$l : E \to \mathbb{R}_{\geq 0}$
Output : TSP Tour $(v_1, v_2, \ldots, v_n)$

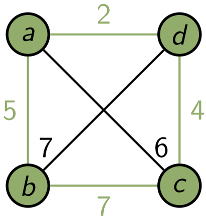
Berechne minimalen Spannbaum $T = (V, E')$ in G (Algorithmus von Prim);
Finde geschl. Kantenzug $(x_1, x_2, \ldots, x_l)$, der jede Kante in E' genau zwei Mal verwendet und speichere als Stapel $S = (x_1, x_2, \ldots, x_l)$;
$W \leftarrow \emptyset$;
while $W \neq V$ **do**
    $y \leftarrow \text{Pop}(S)$;
    if $y \notin W$ **then**
        $W \leftarrow W \cup \{y\}$;
        $v_{|W|} \leftarrow y$;
    end
end



$S = (d, a, b)$

$y = c$

```
Input : Vollst. gew. Graph G = (V, E) mit
        l : E → ℝ≥0
Output : TSP Tour (v_1, v_2, ..., v_n)
Berechne minimalen Spannbaum
T = (V, E') in G (Algorithmus von Prim);
Finde geschl. Kantenzug (x_1, x_2, ..., x_l), der
jede Kante in E' genau zwei Mal verwendet
und speichere als Stapel S = (x_1, x_2, ..., x_l);
W ← ∅;
while W ≠ V do
    y ← Pop(S);
    if y ∉ W then
        W ← W ∪ {y};
        v_|W| ← y;
    end
end
```



TSP-Tour mit Länge 18

Wie gut sind diese Heuristiken?

- $H_{\mathrm{NN}}(G)$: von NN-Heuristik gefundene TSP-Tour in G
- $H_{\mathrm{MST}}(G)$: von MST-Heuristik gefundene TSP-Tour in G
- $H_{\mathrm{opt}}(G)$: optimale TSP-Tour in G

Wie gut sind diese Heuristiken?

- $H_{\mathrm{NN}}(G)$: von NN-Heuristik gefundene TSP-Tour in G
- $H_{\mathrm{MST}}(G)$: von MST-Heuristik gefundene TSP-Tour in G
- $H_{\mathrm{opt}}(G)$: optimale TSP-Tour in G

Aufgabe 6.4c)

Es gibt kein $r \in \mathbb{R}_{\geq 0}$, so dass

$$l(H_{\mathrm{NN}}(G)) \leq r \cdot l(H_{\mathrm{opt}}(G))$$

für alle (metrischen) Graphen $G = (V, \binom{V}{2})$ und $l : \binom{V}{2} \to \mathbb{R}_{\geq 0}$ gilt.

Wie gut sind diese Heuristiken?

- $H_{NN}(G)$: von NN-Heuristik gefundene TSP-Tour in G
- $H_{MST}(G)$: von MST-Heuristik gefundene TSP-Tour in G
- $H_{opt}(G)$: optimale TSP-Tour in G

Aufgabe 6.4c)

Es gibt kein $r \in \mathbb{R}_{\geq 0}$, so dass

$$l(H_{NN}(G)) \leq r \cdot l(H_{opt}(G))$$

für alle (metrischen) Graphen $G = (V, \binom{V}{2})$ und $l : \binom{V}{2} \to \mathbb{R}_{\geq 0}$ gilt.

Aufgabe 4.3b)

Für alle metrischen Graphen $G = (V, \binom{V}{2})$ und $l : \binom{V}{2} \to \mathbb{R}_{\geq 0}$ gilt

$$l(H_{MST}(G)) \leq 2 \cdot l(H_{opt}(G)).$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf?

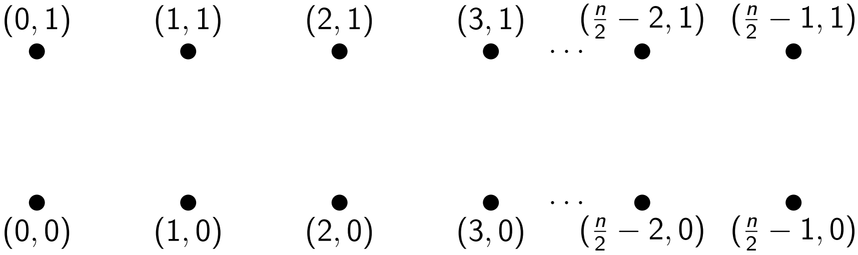
Wir zeigen: für jedes $\varepsilon > 0$ existiert ein metrischer Graph G, so dass

$$l(H_{\mathrm{MST}}(G)) \geq (2 - \varepsilon) \cdot l(H_{\mathrm{opt}}(G)).$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf?

Wir zeigen: für jedes $\varepsilon > 0$ existiert ein metrischer Graph G, so dass

$$l(H_{\mathrm{MST}}(G)) \geq (2 - \varepsilon) \cdot l(H_{\mathrm{opt}}(G)).$$

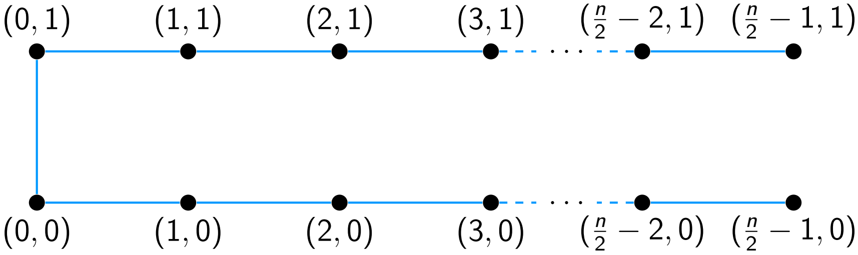


$$l(\{(x_1,y_1),(x_2,y_2)\}) = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf?

Wir zeigen: für jedes $\varepsilon > 0$ existiert ein metrischer Graph G, so dass

$$l(H_{\mathrm{MST}}(G)) \geq (2-\varepsilon) \cdot l(H_{\mathrm{opt}}(G)).$$



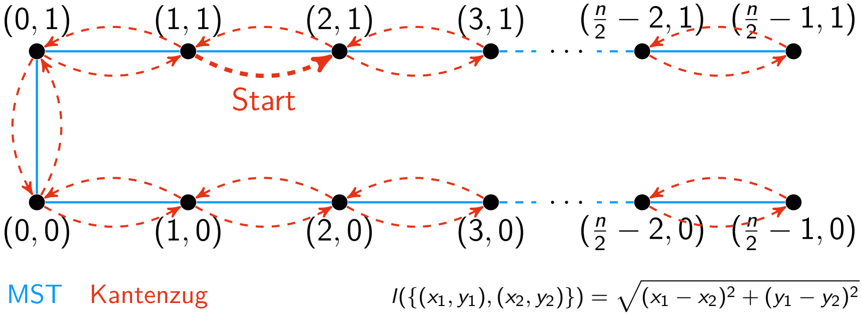
MST

$$l(\{(x_1,y_1),(x_2,y_2)\}) = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf?

Wir zeigen: für jedes $\varepsilon > 0$ existiert ein metrischer Graph G, so dass

$$l(H_{\mathrm{MST}}(G)) \geq (2-\varepsilon) \cdot l(H_{\mathrm{opt}}(G)).$$

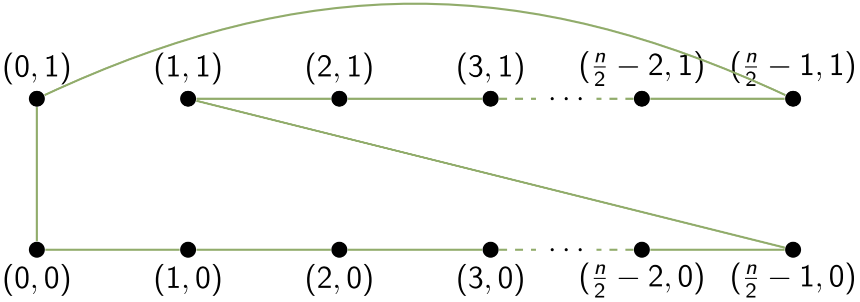


$$l(\{(x_1,y_1),(x_2,y_2)\}) = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf?

Wir zeigen: für jedes $\varepsilon > 0$ existiert ein metrischer Graph G, so dass

$$l(H_{\mathrm{MST}}(G)) \geq (2 - \varepsilon) \cdot l(H_{\mathrm{opt}}(G)).$$



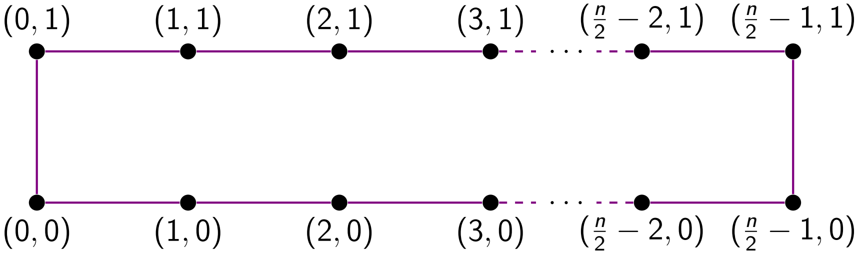
$H_{\mathrm{MST}}(G)$

$$l(\{(x_1,y_1),(x_2,y_2)\}) = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf?

Wir zeigen: für jedes $\varepsilon > 0$ existiert ein metrischer Graph G, so dass

$$l(H_{\mathrm{MST}}(G)) \geq (2-\varepsilon) \cdot l(H_{\mathrm{opt}}(G)).$$



$H_{\mathrm{opt}}(G)$

$$l(\{(x_1, y_1), (x_2, y_2)\}) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Ist $l(H_{\mathrm{MST}}(G)) \leq 2 \cdot l(H_{\mathrm{opt}}(G))$ scharf? (ctd.)

- Sei $\varepsilon > 0$. Wähle $n \geq \frac{5}{\varepsilon}$.
- Für den Graphen G ist $l(H_{\mathrm{opt}}(G)) = n$.
- Weiter ist

$$\begin{aligned} l(H_{\mathrm{MST}}(G)) &= \left(\frac{n}{2} - 2\right) + \left(\frac{n}{2} - 1\right) + \frac{n}{2} + \sqrt{\left(\frac{n}{2} - 2\right)^2 + 1} \\ &\geq \frac{3}{2}n - 3 + \left(\frac{n}{2} - 2\right) \\ &= 2n - 5 \\ &= (2 - \frac{5}{n}) \cdot n \\ &\geq (2 - \varepsilon) \cdot l(H_{\mathrm{opt}}(G)). \end{aligned}$$