



Datenstrukturen und Algorithmen SS07

Datum: 27.6.2007

Michael Belfrage

mbe@student.ethz.ch

belfrage.net/eth

Programm von Heute

- *Online Algorithmen*
 - *Update von Listen*
 - *Move to Front (MTF)*
 - *Transpose*
- *Approximationen für das Traveling Salesman Problem (TSP)*
 - *MST-Heuristik*
 - *Christofides-Heuristik*

Update von Listen

- *Gegeben:*
 - *Liste von Elementen*
 - *Sequenz von Anfragen*
- *Gesucht*
 - *Algorithmus welcher die Anfragekosten minimiert. (Dabei darf die Liste nach einer Anfrage umgeordnet werden)*
- *Kosten:*
 - *i -tes Element zugreifen kostet i*
 - *Umordnen ist gratis (in diesem Modell)*

Update von Listen

Beispiel

Liste von Elementen:



Sequenz von Anfragen:

T, E, S, T, T, E, S, T

Update von Listen

Beispiel

Liste von Elementen:



Sequenz von Anfragen:

T, E, S, T, T, E, S, T

Anfragekosten (ohne Umordnung):

$$4 + 5 + 3 + 4 + 4 + 5 + 3 + 4 = 32$$

Update von Listen

Umordnungsstrategien

– Move To Front (MTF):

- Setze das zuletzt zugegriffene Element an den Anfang der Liste*

– Transpose:

- Vertausche das zugegriffene Element mit seinem Vorgänger (falls vorhanden)*

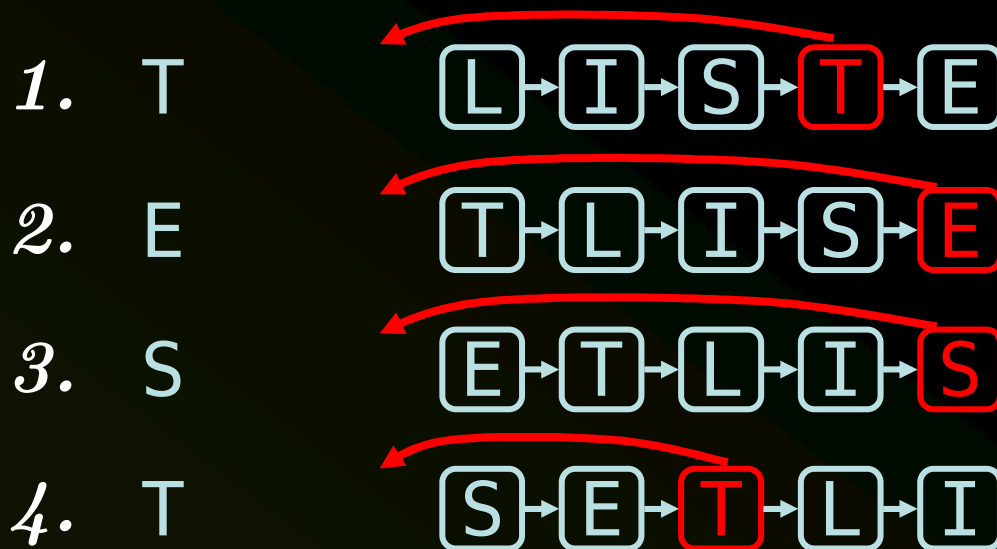
Update von Listen

Beispiel: *Move to Front*

Liste von Elementen:



Sequenz von Anfragen:



Update von Listen

*[Fortsetzung] Beispiel: **Move to Front***

Liste von Elementen:



Sequenz von Anfragen:

T, E, S, T, T, E, S, T

Anfragekosten:

$$4 + 5 + 5 + 3 + 1 + 3 + 3 + 3 = 27$$

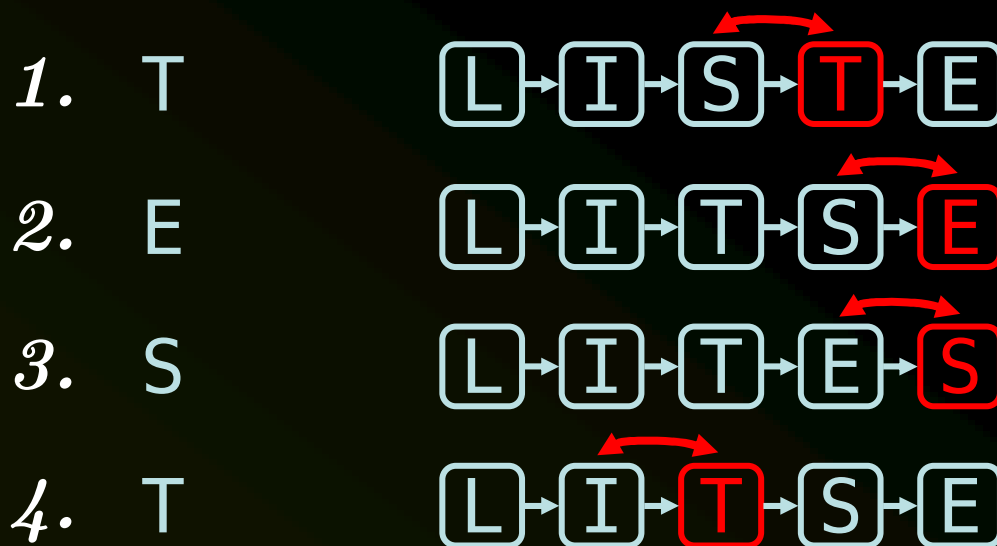
Update von Listen

Beispiel: *Transpose*

Liste von Elementen:



Sequenz von Anfragen:



Update von Listen

*[Fortsetzung] Beispiel: **Transpose***

Liste von Elementen:



Sequenz von Anfragen:

T, E, S, T, T, E, S, T

Anfragekosten:

$$4 + 5 + 5 + 3 + 2 + 5 + 5 + 1 = 30$$

Update von Listen

Anfragekosten im worst-case mittels kompetitiver Analyse:

– Move To Front

- das doppelte des optimalen Algorithmuses*

– Transpose

- beliebig höher als der optimale Algorithmus
:(*

Für Beweise siehe:

- D. D. Sleator, and R. E. Tarjan (1985), Amortized efficiency of list update and paging rules. *Communications of the ACM* 28, 202–208.*
- www.cs.cmu.edu/~sleator/papers/amortized-efficiency.pdf*

TSP-Approximationen

- *Traveling Salesman Problem (TSP)*
 - *Problembeschreibung:*
 - *Siehe dazu den ausgezeichneten Artikel:*
http://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden
 - *Approximationen unter der Annahme, dass Dreiecksungleichung gilt im Graph:*
 - *MST-Heuristik*
 - *Maximal Faktor 2 Abweichung vom Optimum*
 - *Christofides-Heuristik*
 - *Maximal Faktor 1.5 Abweichung vom Optimum*

MST-Heuristik

Input:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
(z.B. mittels Kruskal)
2. Verdoppele alle Kanten vom MST
3. Traversiere MST mit Abkürzung

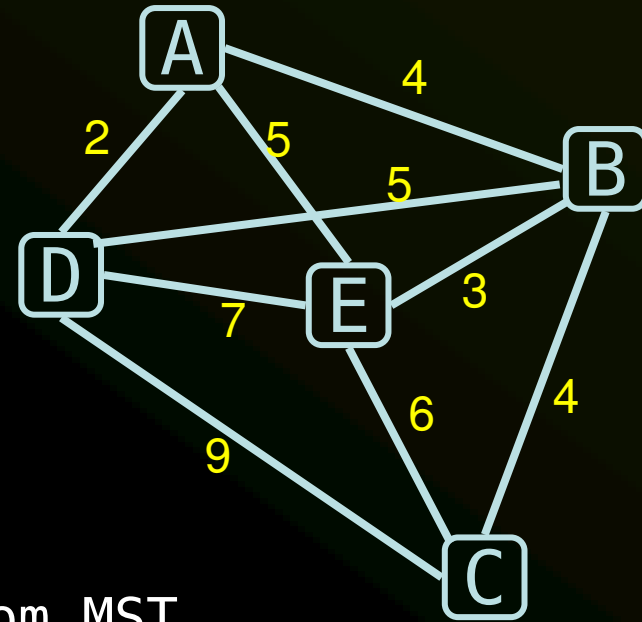
MST-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Verdoppele alle Kanten vom MST
3. Traversiere MST mit Abkürzung



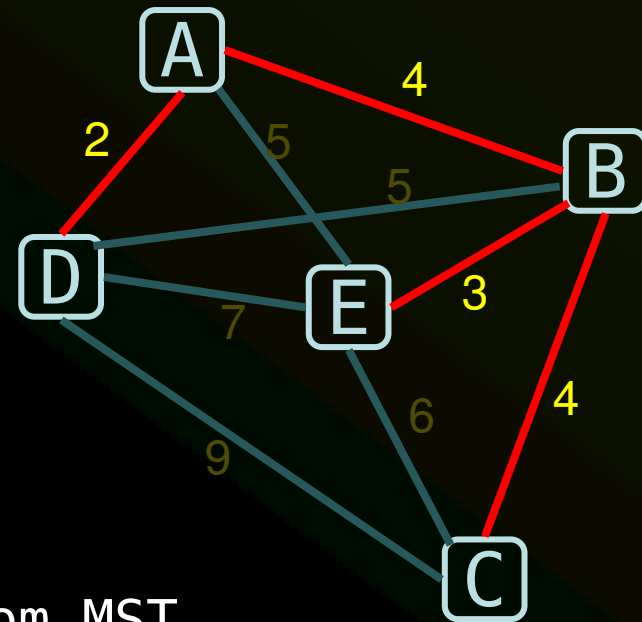
MST-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Verdoppele alle Kanten vom MST
3. Traversiere MST mit Abkürzung



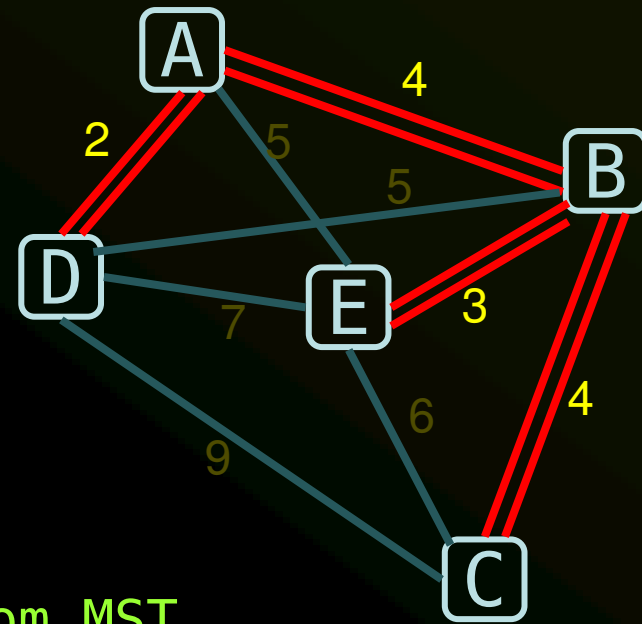
MST-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Verdoppele alle Kanten vom MST
3. Traversiere MST mit Abkürzung



MST-Heuristik

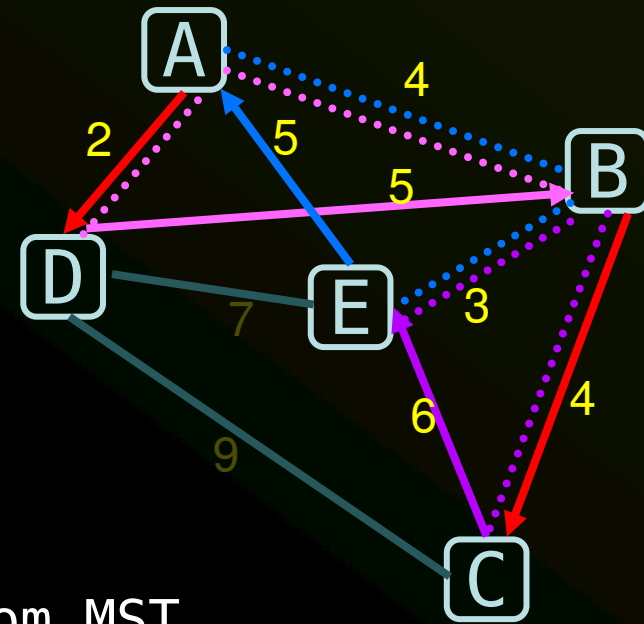
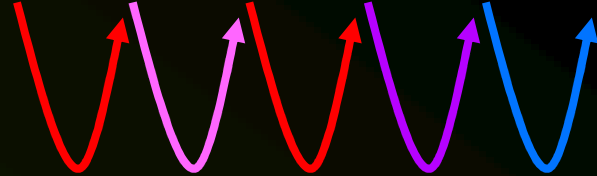
Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Verdoppele alle Kanten vom MST
3. Traversiere MST mit Abkürzung

(A , D , B , C , E , A)



Christofides-Heuristik

Input:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
(z.B. mittels Kruskal)
2. Bestimme **MCM** auf ungeraden Knoten
3. Füge Kanten vom MCM hinzu
4. Traversiere mit Abkürzungen

*Definition: Ein **Matching** M auf einem Graphen $G := (V, E)$ ist eine Teilmenge von E , sodass jeder Knoten von G in genau einer Kante aus M enthalten ist*

Minimal Cost Matching

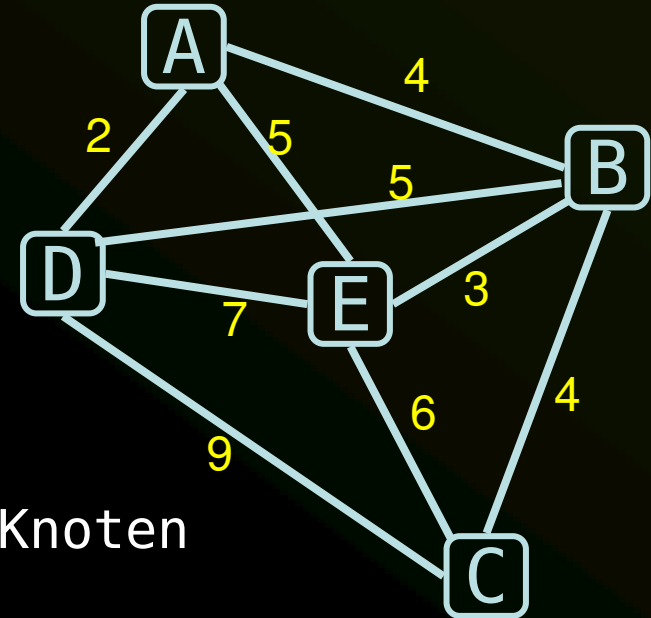
Christofides-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Bestimme MCM auf ungeraden Knoten
3. Füge Kanten vom MCM hinzu
4. Traversiere mit Abkürzungen



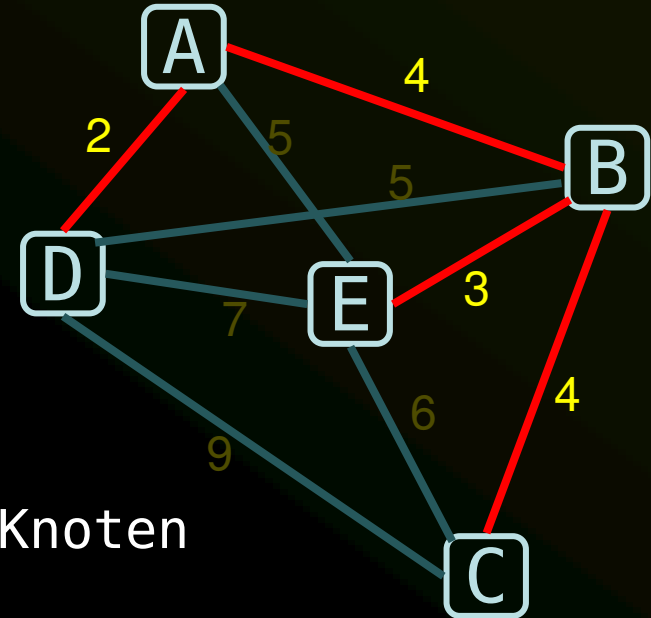
Christofides-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Bestimme MCM auf ungeraden Knoten
3. Füge Kanten vom MCM hinzu
4. Traversiere mit Abkürzungen



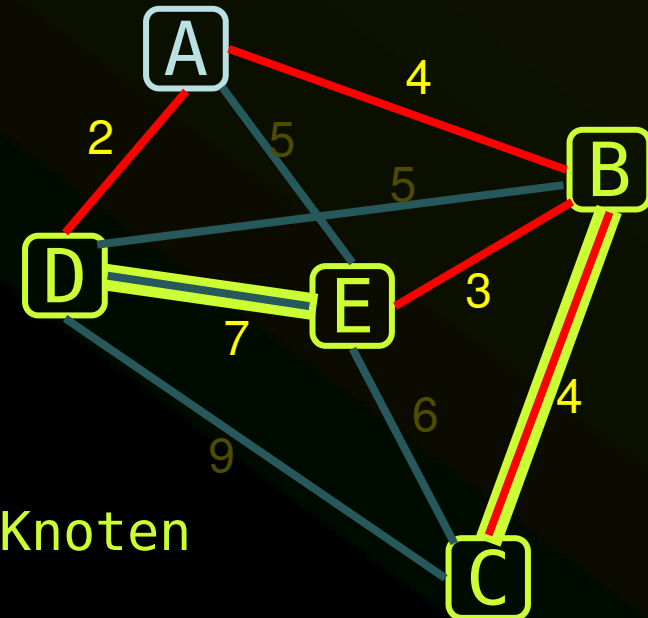
Christofides-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Bestimme MCM auf ungeraden Knoten
3. Füge Kanten vom MCM hinzu
4. Traversiere mit Abkürzungen



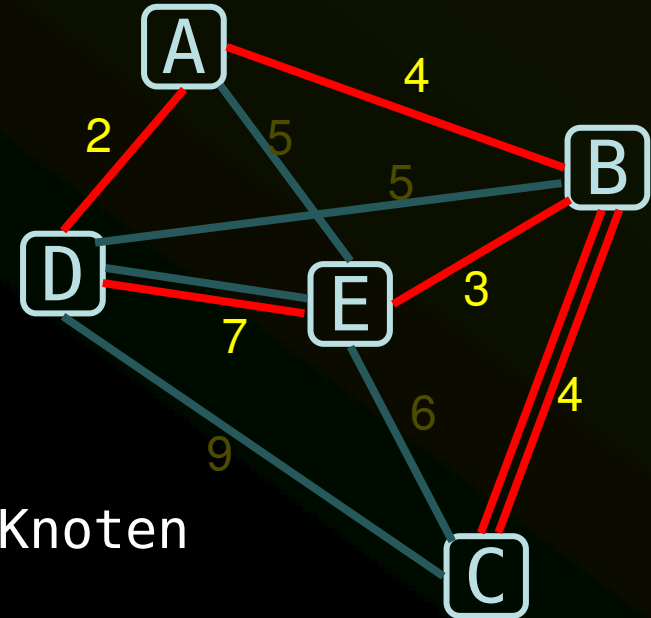
Christofides-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Bestimme MCM auf ungeraden Knoten
3. Füge Kanten vom MCM hinzu
4. Traversiere mit Abkürzungen



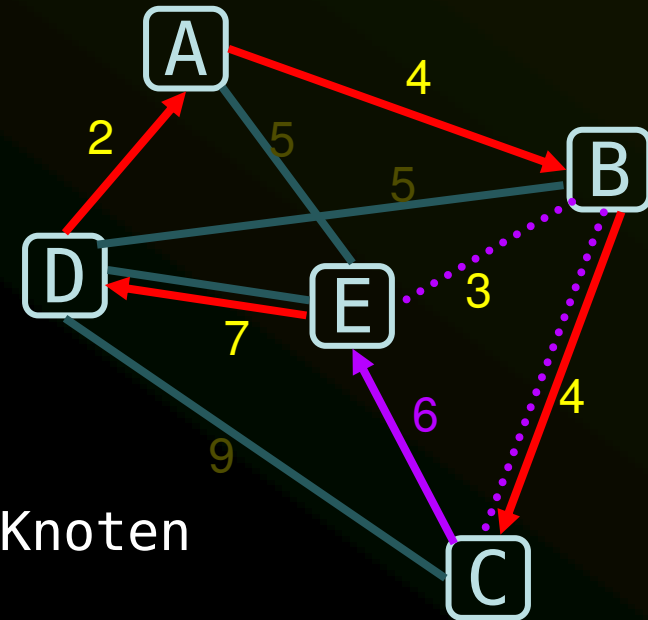
Christofides-Heuristik

Beispiel:

Graph $G := (V, E)$

Algorithmus:

1. Konstruiere MST von G
2. Bestimme MCM auf ungeraden Knoten
3. Füge Kanten vom MCM hinzu
4. Traversiere mit Abkürzungen
(A, B, C, E, D, A)



Ende der Stunde.

Questions?

if yes, send an e-mail.