Algorithmen & Datenstrukturen

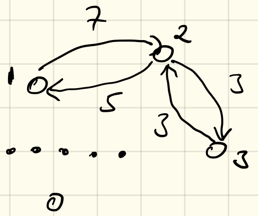
Herbst 2018

Vorlesung 10

Travelling Salesman (Rundreiseproblem)

n Städte $1 \dots n$ vollständiger Graph

Distanzen d_{ij}, $1 \le i,j \le n$ (muss nicht symmetrisch sein, d.h. $d_{ij} \neq d_{ji}$ ist möglich)



etc. . . . . . .

gesucht: Rundreise die jede Stadt genau einmal besucht und minimale Länge hat

trivialer Algorithmus: schaue alle Permutationen der n Städte an

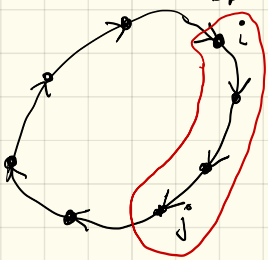
Laufzeit: $O(n \cdot n!)$

Länge ausrechnen

alle Permutationen

optimale Rundreise

Induktion: (Idee)



Reisestück: $\underbrace{i \to \dots \to j}_{S \subseteq \{1 \dots n\}}$

ist kürzest unter allen von i nach j und alle $s \in S$ besuchen

"Optimalitätsprinzip"

DP Versuch:

min über alle $s \in S \setminus \{j\}$

$$D(S, i, j) = \min_{\substack{s \in S \\ s \neq j}} \left(D(S \setminus \{j\}, i, s) + d_{sj} \right)$$

kürzester Weg $i \to j$ mit Knoten in S

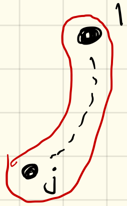
kürzester Weg $i \to s$ mit Knoten in $S \setminus \{j\}$

Distanz $s \to j$

Induktion über $|S|$.

o. B. d. A.: wähle $i = 1$ als Anfangsstadt:

$$D(S, j) = \min_{\substack{s \in S \\ s \neq j}} \left(D(S \setminus \{j\}, s) + d_{sj} \right)$$



also immer $1 \in S$

Algorithmus:

$D(\{1\}, 1) = 0$ // Initialisiere ($|S| = 1$, $1 \in S$)
for $z = 2..n$ do // Induktion über $|S| = z$
  for each $S \subseteq \{1..n\}$ mit $1 \in S$ und $|S| = z$
    $D(S, 1) = \infty$
    for each $j \in S$, $j \neq 1$ do
      $D(S, j) = \min_{\substack{s \in S \\ s \neq j}} \left(D(S \setminus \{j\}, s) + d_{sj} \right)$

return $\min_j \left(D(\{1..n\}, j) + d_{j1} \right)$

Analyse:

äussere 2 for loops: # Teilmengen S mit $1 \in S$: 2^{n-1}

innere for loop: $\leq n$ Iterationen

je Iteration minimum: $O(n)$

Laufzeit: $O(n^2 2^n)$