# DYNAMISCHE PROGRAMMIERUNG

## Arora's PTAS für das Euklidsche TSP

# Vorbereitungen

Wir beschränken uns auf $\mathbb{R}^2$

Verschieben: s.d. die minimum Koordinaten (vert. und horiz.) 0 werden

Skalieren: s.d. die maximum Koordinate (vert. oder horiz.) $n^2$ wird

Runden: jede Koordinate wird abgerundet:

$$P = (x, y) \longrightarrow P' = (\lfloor x \rfloor, \lfloor y \rfloor)$$

# Ein Baum wird definiert

- Wir definieren einen Baum $B$ : jeder Knoten hat entweder vier Kinder, oder ist ein Blatt;
- die Wurzel entspricht einem Quadrat $Q_0 \supseteq [0, n^2] \times [0, n^2]$
- ihre vier Kinder sind die vier Teilquadrate der halben Länge,... usw....
- ein Quadrat wird nicht weiter zerlegt falls es 0 oder 1 Eingabepunkt enthält

# Legale Rundreisen

- an den beiden Trennlinien jedes Teilquadrats $Q$ definieren jeweils $m$ *Türen* (Treffpunkte) im gleichen Abstand
- Distanz zweier Türen auf einer Trennlinie der Schicht $i$ ist

$$\approx \frac{n^2}{2^i \cdot m}$$

- eine Rundreise heißt *legal* wenn sie ein Quadrat nur über eine Tür betritt und verlässt
- der Algorithmus sucht eine kürzeste *legale* Rundreise

# Definition: Besuchsmuster

Ein <u>Besuchsmuster</u> $\widetilde{T}$ für ein Teilquadrat $Q$
ist *eine Menge von Tür-Paaren* am Rand des Teilquadrats

$$\widetilde{T} = \{\{T_1, T_1'\}, \{T_2, T_2'\}, \{T_3, T_3'\}, \ldots, \{T_r, T_r'\}\}.$$

(Die Ordnung der Türen und der Paaren zählt nicht.)

Sei $L_Q(\widetilde{T})$ die minimale Länge einer Teilrundreise
mit dem Besuchsmuster $\widetilde{T}$ im Quadrat $Q$.

# Dynamische Programmierung

- für ein Blatt-Quadrat $Q_b$ *ohne* Eingabepunkt ist
  $$L_{Q_b}(\widetilde{T}) = \sum_{i=1}^{k} d(T_i, T_i');$$
- für ein Blatt $Q_b$ mit *einem* Eingabepunkt $P$
  nehmen wir den Umweg $T_i \to P \to T_i'$ mit der kleinsten *zusätzlichen* Länge;
- für ein Vater-Quadrat $Q$
  1. betrachte *alle* Kombinationen von Besuchsmuster der 4 Kind-Quadrate $(\widetilde{T}_1, \widetilde{T}_2, \widetilde{T}_3, \widetilde{T}_4)$
  2. <u>falls konsistent</u> mit $\widetilde{T}$, berechne
     $L_{Q_1}(\widetilde{T}_1) + L_{Q_2}(\widetilde{T}_2) + L_{Q_3}(\widetilde{T}_3) + L_{Q_4}(\widetilde{T}_4)$.
  3. $L_Q(\widetilde{T})$ ist das Minimum dieser Summen über alle passende $(\widetilde{T}_1, \widetilde{T}_2, \widetilde{T}_3, \widetilde{T}_4)$.

# Laufzeit

die Anzahl möglicher Besuchsmuster für ein Quadrat:

$$2^{\mathcal{O}(m)}$$

Laufzeit für ein Quadrat:

da alle Kombinationen $(\widetilde{T}, \widetilde{T}_1, \widetilde{T}_2, \widetilde{T}_3, \widetilde{T}_4)$ evaluiert werden

$$[2^{\mathcal{O}(m)}]^5 = 2^{\mathcal{O}(m)}$$

Der Baum $B$ hat $\mathcal{O}(n \log n)$ Knoten(=Quadrate)

Laufzeit für alle Quadrate (alle Knoten im Baum):

$$\mathcal{O}(n \cdot \log n) \cdot 2^{\mathcal{O}(m)} = \mathcal{O}(n \cdot \log n) \cdot n^{\mathcal{O}(1/\varepsilon)}$$

# Beobachtungen

Beobachtung 1: Es gibt eine kürzeste legale Rundreise die

sich nicht kreuzt, und sich nur in Türen trifft.

Beobachtung 2: Eine kürzeste legale Rundreise besucht jede Tür

höchstens zweimal.

Aus Beobachtungen 1 und 2 folgt:
die Anzahl möglicher Besuchsmuster für ein beliebiges Quadrat:

$$2^{\mathcal{O}(m)} = n^{\mathcal{O}(1/\varepsilon)}.$$

# Wann ist der Algorithmus $(1+\varepsilon)$-approximativ?

Brauchen: $\mathrm{OPT}_{\mathrm{legal}} \leq (1+\varepsilon) \cdot \mathrm{OPT}$

- In einer optimalen Rundreise verschiebe jede Kreuzung mit einer Trennlinie in eine Tür
- die Verlängerung wegen *einer* Kreuzung der Schicht $i$ ist $\leq n^2/(2^i \cdot m) \rightarrow$ kann zu viel werden!
- berechne die erwartete Verlängerung, falls wir das einschließende Quadrat vergrößern, und zufällig positionieren!

$$\mathbb{E}[\mathit{Verlaengerung\ einer\ Kreuzung}] =$$

$$= \sum_{i=0}^{2\log n} \frac{2n^2}{2^i \cdot m} \cdot \frac{2^i}{2n^2} = \sum_{i=0}^{2\log n} \frac{1}{m} = \frac{2\log n}{m} = \frac{2\log n \cdot \varepsilon}{2\log n} = \varepsilon$$

# Approximationsfaktor

Beobachtungen: Eine Rundreise der Länge $\mathrm{OPT}$ kreuzt $\leq \mathrm{OPT}$ horizontale und $\leq \mathrm{OPT}$ vertikale Trennlinien, das ergibt $\leq 2 \cdot \mathrm{OPT}$ Kreuzungen.

– Die *erwartete* Verlängerung wegen 'Legalmachung' in *einer* Kreuzung ist $\leq \varepsilon$

– Die *erwartete* Verlängerung insgesamt ist $\leq \varepsilon \cdot 2 \cdot \mathrm{OPT}$.

Korollar: Mit Wahrscheinlichkeit $\geq 1/2$ ist die Verlängerung $\leq 4\varepsilon$, und die legale Rundreise ist $(1 + 4\varepsilon)$-approximativ.

# Arora's Algorithmus

$n$ Punkte in $\mathbb{R}^2$ sind gegeben

- sei $L > n^2$ Zweierpotenz, und sei $[0, L] \times [0, L]$ das kleinste umschließende Quadrat (nach Skalieren)
- sei $Q_0$ ein Quadrat der Länge $2L$ mit Eckpunkt in $[-L, 0] \times [-L, 0]$ *zufällig* gewählt.
- bestimme den Baum der Teilquadrate von $Q_0$ mit $m$ Türen auf jeder Trennlinie ( $m > (2 \log n)/\varepsilon$ Zweierpotenz)
- bestimme die Längen der Teilrundreisen $L_Q(\widetilde{T})$ bottom-up

$$\mathrm{OPT}_{\mathrm{legal}} = L_{Q_0}(\emptyset)$$

- bestimme die Rundreise $R_{\mathrm{OPT}}^{\mathrm{legal}}$ top-down

Theorem: Arora's Algorithmus findet eine Rundreise mit erwarteter Länge $\leq (1 + 2\varepsilon)\mathrm{OPT}$.

Mit Wahrscheinlichkeit $\geq 1/2$ wird ihre Länge $\leq (1 + 4\varepsilon)\mathrm{OPT}$.

Die Laufzeit kann auf $n \cdot (\log n)^{\mathcal{O}(1/\varepsilon)}$ verbessert werden.

(Derandomisierung: *Alle* Quadrate $Q_0$ mit Eckpunkt in $[-L, 0] \times [-L, 0]$ werden ausprobiert und die kürzeste Rundreise gewählt.)

Siehe auch: Vazirani: Approximation Algorithms S. 84.–88.

# Zusammenfassung: Approximierbarkeit von TSP

- das euklidsche TSP: $(1 + \varepsilon)$-approximierbar (Arora's PTAS)

- das metrische TSP: $\frac{3}{2}$-approximierbar (Christofides)

  keine polynomielle Approximation $< \frac{220}{219}$

- das allgemeine TSP 'gar nicht' approximierbar

Theorem: Es gibt keinen effizienten $f(n)$-approximativen Algorithmus mit $f(n) = \mathcal{O}(2^n)$ für das allgemeine TSP.