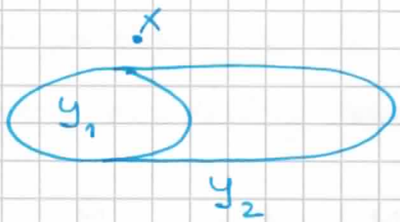
**Theorem:** Ein Durchschnitt von $k$ Matroiden ist $k$-austauschbar und deshalb ein $k$-Matroid.

**Beweis:** $(\mathcal{M}_1, X)\ (\mathcal{M}_2, X) \ldots \ldots (\mathcal{M}_k, X)$

seien 1-Matroide; sie sind also 1-austauschbar

Sei $\mathcal{M} = \mathcal{M}_1 \cap \mathcal{M}_2 \cap \mathcal{M}_3 \cap \ldots \cap \mathcal{M}_k$

d. h. $Y$ ist unabhängig falls es in allen $\mathcal{M}_i$ unabhängig ist




Seien

$Y_1 \subset Y_2 \quad Y_1, Y_2 \in \mathcal{M}$

und $(Y_1 \cup \{x\}) \in \mathcal{M}$

Können wir jetzt höchstens $k$ Elemente $y_1\, y_2 \ldots y_k$ aus $Y_2 \setminus Y_1$ entfernen so dass dann $(Y_2 \setminus \{y_1 \ldots y_k\}) \cup \{x\} \in \mathcal{M}$

$\exists\, y_1 \quad (Y_2 \setminus \{y_1\}) \cup \{x\} \in \mathcal{M}_1$

$\exists\, y_2 \quad (Y_2 \setminus \{y_2\}) \cup \{x\} \in \mathcal{M}_2$

$\vdots$

$\exists\, y_k \quad (Y_2 \setminus \{y_k\}) \cup \{x\} \in \mathcal{M}_k$

$$\Longrightarrow (Y_2 \setminus \{y_1\, y_2 \ldots y_k\}) \cup \{x\} \in \mathcal{M}_1 \cap \mathcal{M}_2 \cap \ldots \cap \mathcal{M}_k = \mathcal{M}$$

$\square$

A1.

# Arora's PTAS für das euklidische Travelling Salesman Problem

(Skizze) (Siehe auch bei Vazirani S. 84.)

Wir beschränken uns auf die Ebene:
Seien $n$ Punkte (Orte) in der Ebene $\mathbb{R}^2$ gegeben.

## a.) Vorbereitungen

Wir möchten, dass die Koordinaten der Punkte ganzzahlig sind, aber so dass das Raster aller dieser ganzen Zahlen nicht zu grob (→ schlechte Approximation) und nicht zu fein (→ nicht-polynomielle Laufzeit) gesetzt wird.

Zu diesem Zweck wenden wir die Koordinaten der Eingabepunkte (verschieben,) skalieren, und runden.

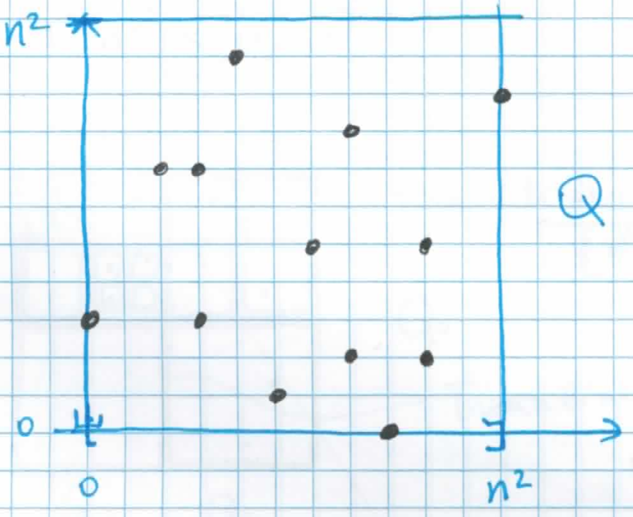
Verschieben: damit die kleinsten Koordinaten horizontal und vertikal 0 sind

Skalieren: so dass die größte Koordinate entweder horizontal oder vertikal gleich $n^2$ ist.

Runden: jede Koordinate wird abgerundet

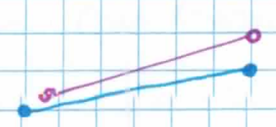
$$P(x,y) \longrightarrow P'(\lfloor x \rfloor, \lfloor y \rfloor)$$

Jetzt sind die Positionen ganzzahlig und alle im Quadrat $\{0, n^2] \times [0, n^2] = Q$

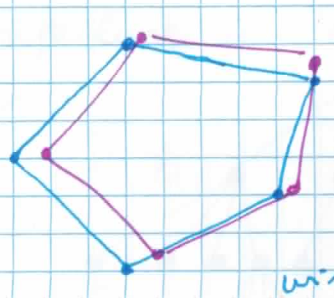



Damit haben wir auch eine untere Schranke für die Länge einer optimalen Rundreise: wir müssen vom linkesten ~~xxxxx~~ Punkt zum rechtesten und zurück: $OPT \geq 2n^2$ (oder vom untersten zum obersten)

Wie ändert die Rundung die Länge einer optimalen Rundreise? Beachte, dass $d(P, P') < \sqrt{2}$ gilt für die Distanz einer Eingabepunkt P vom abgerundeten Punkt P'




die Länge einer "Kante" in der Rundreise ändert sich höchstens um einen additiven $\pm 2\sqrt{2}$ (warum?)




die Länge einer optimalen Rundreise über die gerundeten Punkte ändert sich um $\pm n \cdot 2 \cdot \sqrt{2} \leq O\left(\frac{1}{n}\right) OPT$

$= \frac{\sqrt{2}}{n} \cdot 2n^2$

↓

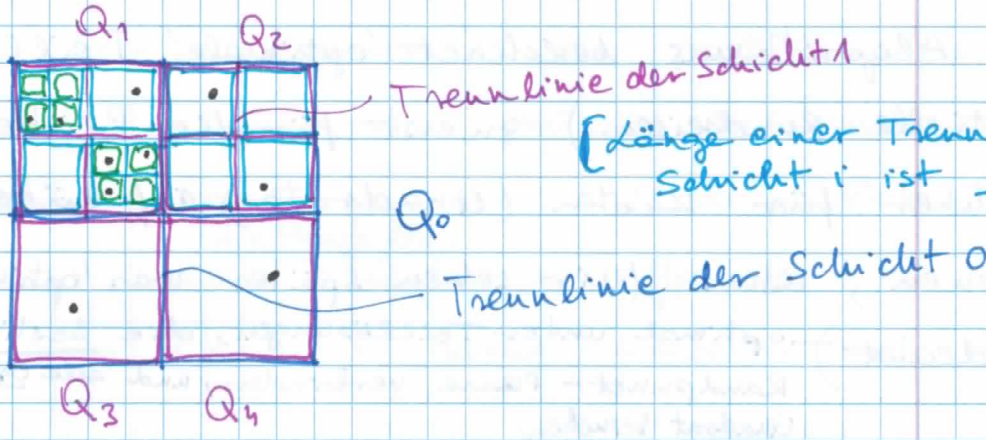
wir berechnen eigentlich eine

$\left(1 + \varepsilon + O\left(\frac{1}{n}\right)\right)$ – approximative Rundreise

## b) eine fast-optimale Rundreise mit dynamischer Programmierung

(die Idee)

$Q_0 = [0, n^2) \times [0, n^2]$



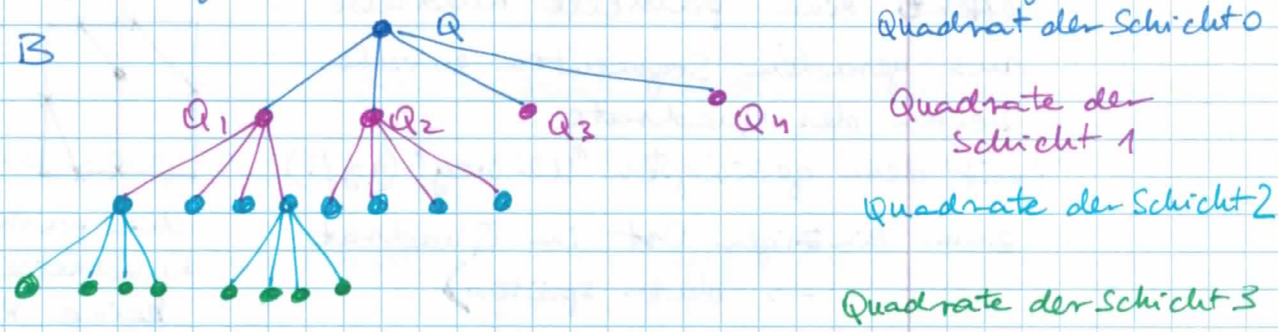

[Länge einer Trennlinie der Schicht $i$ ist $\frac{n^2}{2^i}$]

- wir Teilen das Quadrat mit zwei Trennlinien in vier gleichgroße Teilquadrate; diese teilen wir weiter in vier gleichgroße Teilquadrate, usw.

- ein Quadrat wird <u>nicht</u> weiter aufgeteilt falls es nur noch 0 oder 1 Punkt von der (abgerundeten) Eingabe enthält.
  (dies erfolgt spätestens bei Quadrate der Länge 1, weil die Orte gerundet sind)
  $\Longrightarrow \leq \log n^2 = 2\log n$ Schichte

- wir werden optimale <u>partielle</u> Rundreisen in den Teilquadraten bottom-up in gute partielle Rundreisen in den größeren Quadraten zusammenbauen. (Achtung: diese sind keine echten Rundreisen, nur Teillösungen die aus geraden Segmenten bestehen!) Was bedeutet hier "bottom-up"?

- Die Aufteilung in Teilquadrate entspricht dem folgenden Baum



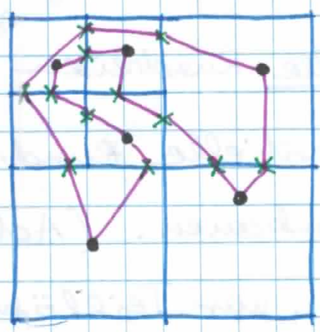
A4.

- jeder Knoten entspricht einem Quadrat; jeder Knoten im Baum B ist entweder ein Blatt oder hat vier Kinder (4-ärer Baum)

- der Algorithmus berechnet 'optimale' Teillösungen (partielle Rundreisen) zuerst für die Blätter, nachher für Knoten (Quadrate) auf höheren Ebenen, usw. (wir sehen später was 'optimal' bedeutet) – optimal unter Teillösungen, die bestimmte Randpunkt-Paare verbinden, und alle Orte im Quadrat besuchen

[ die Tiefe des Baums B ist höchstens $\log n^2 = O(\log n)$ weil Quadrate der Länge 1 enthalten 0 oder 1 Punkt. Die Anzahl seiner Knoten (der Quadrate) ist $O(n \cdot \log n)$ siehe später ]

- Was ist eine partielle Rundreise, und wie werden diese zusammengebaut?

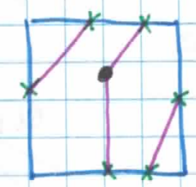


× Tür

• Punkt (Ort)

zusammenpassende partielle Rundreisen

→ wenn ein Quadrat im B einem Blatt entspricht, besteht eine partielle Rundreise aus geraden Segmenten zwischen Seiten des Quadrats (mit dem geringsten "Umweg" (ggf.) zum einzigen Ort im Quadrat → dazu später)



die partielle Rundreise hat keine Richtung(en)

→ wir möchten, dass wir solche partielle Rundreisen für die Teilquadrate aneinander kleben können, so dass sie Teillösungen (partielle Rundreisen) für ihr Vater-Quadrat bestimmen.

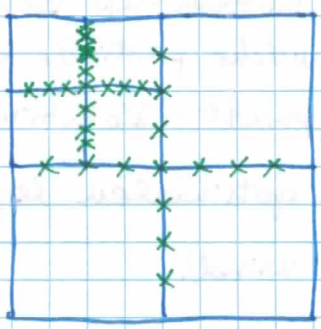
Aber: die Treffpunkte × der partiellen Rundreisen in den Teilquadraten müssen zusammenpassen, sonst bilden sie keine partielle Rundreise im Vater-Quadrat

⇒ wir müssen $\underbrace{\text{möglichen}}_{}$ (die Menge der) möglichen Treffpunkte × am Anfang festlegen (damit es nicht unendlich viele Möglichkeiten gibt); diese festgelegten Treffpunkte nennen wir <u>Türen</u>.

Die Anzahl der Türen muss sorgfältig gewählt werden:

(zu wenig Türen: zu ~~viele~~ lange Umwege in der gefundene Rundreise (zu "eckig"), und deshalb schlechte Approximation

zu viele Türen: zu viele mögliche Partielle Rundreisen in der Tabelle der dynamischen Programmierung verursacht nicht-polynomielle Laufzeit)



— an beiden Trennlinien eines Teilquadrats legen wir $m$ Türen im gleichen Abstand fest. $\left(m \approx \frac{\log_2 n}{\varepsilon}\right)$

die Anzahl $m$ der Türen ist gleich für die langen wie für die kürzeren Trennlinien.

[die Distanz benachbarten Türen auf einer Trennlinie der Schicht $i$ ist $\sim \frac{n^2}{2^i \cdot m}$ ]

A6.

- eine (komplette) Rundreise heißt __legal__ wenn sie jede Trennlinie nur in Türen kreuzt.
Unser Algorithmus berechnet mit dynamischer Programmierung eine minimale __legale__ Rundreise

(für den Approximationsfaktor muss man später zeigen, dass für eine optimale Rundreise eine legale Rundreise existiert die höchstens um Faktor $1 + O(\varepsilon)$ länger ist)

- die minimale legale Rundreise wird bottom-up berechnet, angefangen mit den kleinsten Quadraten mit maximal einem Ort.

Für jedes Quadrat wird __für alle möglichen__
festgelegte Randpunkt-Paare (Tür-Paare)
__Treffpunkt-Kombinationen am Rand des Quadrats__
die minimale Länge einer legalen Rundreise berechnet und gespeichert.

Beachte dass bis zum Ende der Bottom-up Berechnung
(bzw. welche partielle Rundreise)
nicht bekannt ist, welche Treffpunkt-Kombination am Rand des Quadrats in der optimalen legalen Rundreise letztendlich vorkommen wird!

Was bedeutet eine Treffpunkt-Kombination am Rand eines Quadrats?? (bzw. Tür-Kombination)

# Treffpunkt-Kombination = Besuchsmuster

Ein Besuchsmuster $\tilde{T}$ für ein Teilquadrat (für einen Knoten von B) ist eine ~~Vektor~~ Menge von Paaren von Türen am Rand dieses Quadrats:

$$\tilde{T} = \{\{T_1 T_1'\}, \{T_2 T_2'\}, \{T_3, T_3'\} \dots \{T_r T_r'\}\}$$

die Reihenfolge der Paare und der Türen innerhalb eines Paars zählt nicht (ist irrelevant)

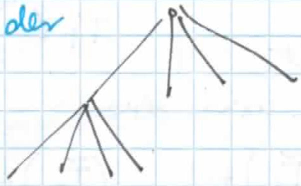
Die Bedeutung eines Besuchsmusters ist, dass eine Rundreise mit ihm consistent ist (die zu diesem speziellen Besuchsmuster passt) für jede $1 \le i \le r$ das Teilquadrat einmal über Tür $T_i$ besucht, und dann über Tür $T_i'$ verlässt, (oder andersrum).

| |
|---|
| Dynamische Programmierung für die Berechnung einer minimalen legalen Rundreise |

Wie für minimum-Gewicht Vertex Cover auf Bäumen, berechnet der Algorithmus für jeden Knoten von B (jedes Teilquadrat) die minimale Länge einer partiellen Rundreise für jedes mögliche Besuchsmuster:

sei $L_Q(\tilde{T})$ diese minimale Länge für Quadrat Q und eines seiner Besuchsmuster $\tilde{T}$

die Werte in der Tabelle



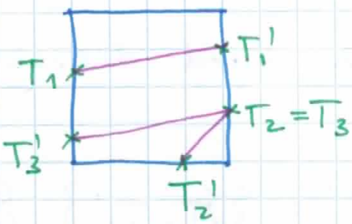
bottom-up Berechnung

A8.

## Die Rekursion:

**Basisfälle:** (sind die Blätter in B)

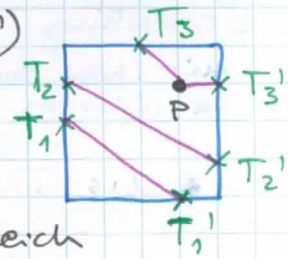
– für ein leeres Blatt-Quadrat Q ohne Eingabepunkt ist $L_Q(\tilde{T})$ einfach die Gesamtlänge der Segmente (Distanzen) $(T_i T_i')$ für jedes Tür-Paar im Besuchsmuster:



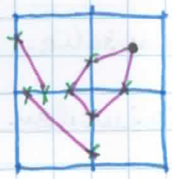

$$\boxed{L_Q(\tilde{T}) = \sum_{i=1}^{r} d(T_i T_i')}$$

$\tilde{T} = \{\{T_1, T_1'\}, \{T_2 T_2'\} \{T_3 T_3'\}\}$

– für ein Blatt-Quadrat mit einem Eingabepunkt P ähnlich aber es wird für ein Tür-Paar statt $d(T_i T_i')$ ein Umweg zu P gerechnet so dass dieser Umweg (d.h. dieses Paar) die geringste zusätzliche Länge (im Vergleich zu $\sum_i d(T_i T_i')$) verursacht.



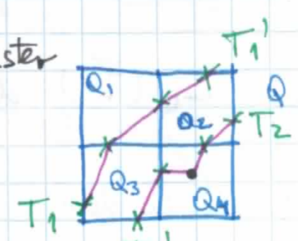

(nicht-passende Besuchsmuster)



**"Rekursionsgleichung"** (für innere Knoten)

– für ein Vater-Quadrat Q und ein Besuchsmuster $\tilde{T}$
 – betrachte alle Kombinationen von Besuchsmuster der 4 Kinder-Quadrate $Q_1\ Q_2\ Q_3\ Q_4$ $(\tilde{T}_1, \tilde{T}_2, \tilde{T}_3, \tilde{T}_4)$
 – diese nennen wir passend zueinander und zu $\tilde{T}$ falls die Türen all dieser Besuchsmuster zusammenpassen und so, dass ~~die~~ eine entsprechende partielle Rundreise mit dem Besuchsmuster $\tilde{T}$ konsistent ist. In diesem Fall addieren wir die Längen $L_{Q_1}(\tilde{T}_1) + L_{Q_2}(\tilde{T}_2) + L_{Q_3}(\tilde{T}_3) + L_{Q_4}(\tilde{T}_4)$

(Beachte dass passende Türen nicht reichen)



- schließlich ist $L_Q(\tilde{T})$ für das Vater-Quadrat das Minimum dieser Gesamtlängen über alle zu $\tilde{T}$ passenden $(\tilde{T}_1\ \tilde{T}_2\ \tilde{T}_3\ \tilde{T}_4)$

$$L_Q(\tilde{T}) = \min\left\{L_{Q_1}(\tilde{T}_1)+L_{Q_2}(\tilde{T}_2)+L_{Q_3}(\tilde{T}_3)+L_{Q_4}(\tilde{T}_4) \;\middle|\; \overset{\text{alle}}{(\tilde{T}_1,\tilde{T}_2,\tilde{T}_3,\tilde{T}_4)} \text{ passend zu } \tilde{T}\right\}$$

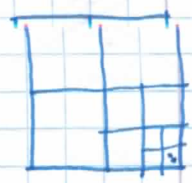
c.) Einige Details zur Laufzeitanalyse

- Die Anzahl aller möglichen Besuchsmuster $\tilde{T}$ für ein Quadrat ist $2^{O(m)} = n^{O(\frac{1}{\varepsilon})}$ (wir sehen später warum)

$$\left(2^m = 2^{\frac{\log n}{\varepsilon}} = n^{\frac{1}{\varepsilon}}\right)$$

- die Laufzeit für ein Quadrat = einen Baum-Knoten ist somit $\left[2^{O(m)}\right]^5 = 2^{O(m)}$, weil alle möglichen Kombinationen $(\tilde{T}, \tilde{T}_1, \tilde{T}_2, \tilde{T}_3, \tilde{T}_4)$ für sein eigenes Besuchsmuster und für seine 4 Teilquadrate betrachtet werden.

- B hat $O(n \cdot \log n)$ Knoten (zwei nahe Orte können für $O(\log n)$ leere Quadrate die Verantwortung tragen:



)

die Laufzeit für den ganzen Baum ist somit

$$O(n \log n) \cdot 2^{O(m)}$$

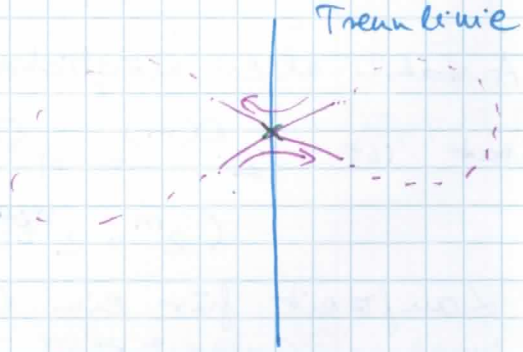
$$= \text{Poly}(n) \cdot n^{O(\frac{1}{\varepsilon})}$$

- Entscheidend für diese Analyse war die Anzahl der Besuchsmuster $2^{O(m)}$

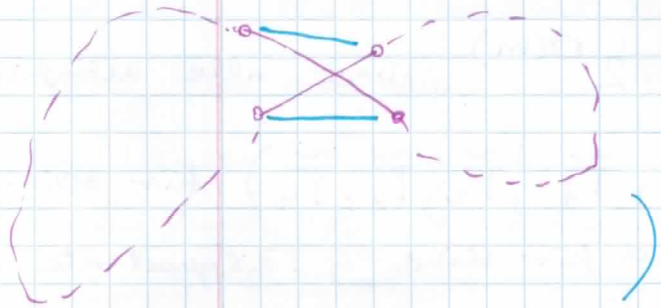
A10.

– Warum ist die Anzahl der Besuchsmuster $\tilde{T}$ für ein Quadrat $2^{O(m)}$?

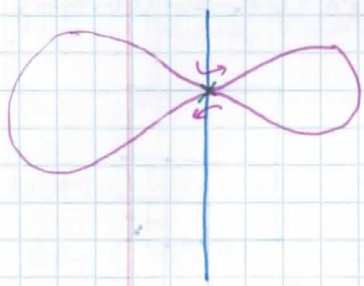
<u>Behauptung 1.</u> Es existiert eine kürzeste legale Rundreise die sich gar nicht <u>kreuzt</u>, und sich höchstens in Türen <u>trifft</u>.



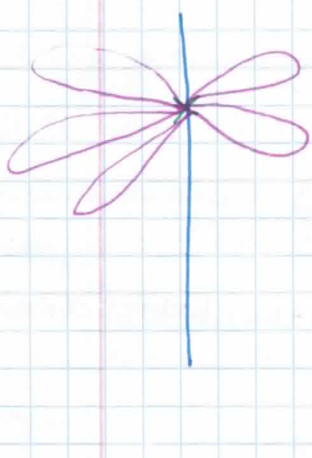

(Warum? jede Kreuzung kann aufgehoben werden so dass die Rundreise kürzer wird)



<u>Behauptung 2:</u> Eine kürzeste Rundreise besucht jede Tür höchstens zweimal.



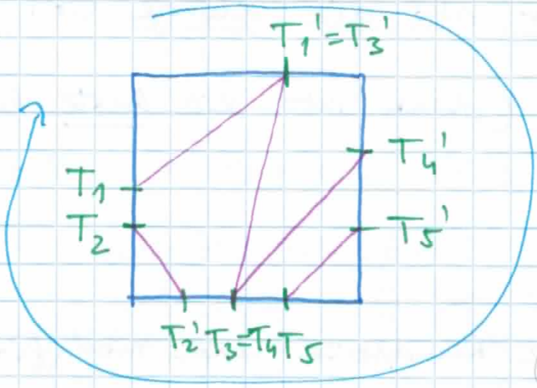
← das ist erlaubt



↙ das ist nicht erlaubt so eine Rundreise könnte gekürzt werden.

Wir können die Berechnung auf solche $\tilde{T}$ einschränken:

Korollar: Wenn wir die Tür-Paare in einem Besuchsmuster $\tilde{T}$ betrachten, die "Kanten" (Strecken) zwischen jedem Paar kreuzen sich nicht, (und jede Tür kommt maximal zweimal im Besuchsmuster vor).




$\tilde{T} = \{T_1, T_4'\}, \{T_2, T_2'\} \{T_3, T_3'\}$.

$3m$ $\binom{3m}{2}$ Paare

$2^{\binom{3m}{2}}$ Besuchsmuster

$2^{O(m^2)}$ → zu viel

$(2^{\log n})^{\log n}$ $n^{\log n}$

Laufen wir am Rand des Quadrats einmal herum, und notieren wir die Türen, und markieren wir sie mit einer öffnenden bzw. mit einer schließenden Klammer beim Treffen der ersten bzw. zweiten Tür desselben Tür-Paares

$$T_1' \; T_3' \; T_4' \; T_5' \; T_5 \; T_4 \; T_3 \; T_2' \; T_2 \; T_1$$

$$( \; ( \; ( \; ( \; ) \; ) \; ) \; ( \; ) \; )$$

→ wir erhalten einen sog. wohlgeformten Klammerausdruck der Länge $\leq 8m$ (weil jede Trennlinie max $m$ Türen hat, und eine Tür wird max. 2mal benutzt).

Thm: Die Anzahl der wohlgeformten Klammerausdrücke der Länge $k$ ist die $k$-te Catalan-Zahl

$$C_k = \frac{1}{k+1}\binom{2k}{k} = 2^{O(k)}$$

Grob: "deshalb" gibt es $2^{O(m)}$ "legale" Besuchsmuster für jedes Quadrat. (präzisere Analyse in Vazirani)

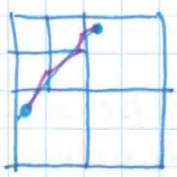
A12.

## d.) Einige Details zum Approximationsfaktor.

Der Algorithmus berechnet eine optimale legale Rundreise

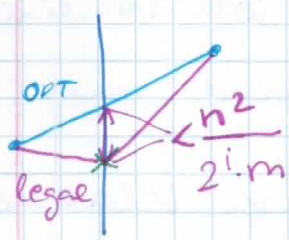
Wir brauchen: irgendeine legale Rundreise ist höchstens um $O(\varepsilon)\cdot OPT$ länger als die Länge OPT einer optimalen Rundreise (dann ist die beste legale Rundreise auch so)

– wir nehmen eine optimale Rundreise und modifizieren sie um eine legale Rundreise zu erhalten:



(wir ersetzen jede Strecke der optimalen Rundreise mit einem Polygonzug: wir verschieben jede Kreuzung mit einer Trennlinie in die nächstliegende Tür ×)

[die Verlängerung wegen einer Kreuzung mit einer Trennlinie der Schicht $i$ ist höchstens $\frac{2n^2}{2^i \cdot m}$



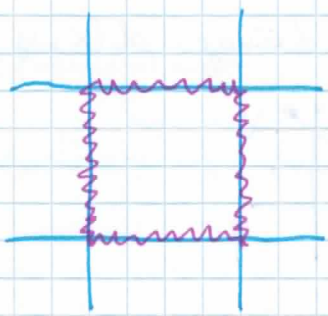

an Trennlinien der niedrigen Schichten werden diese Verlängerungen länger; aber solche Trennlinien gibt es weniger!]

Dies kann viel zu lang werden, wenn die optimale Rundreise eine Trennlinie mit wenigen Türen zu oft kreuzt!

große Umwege zu einer Tür

[ZB. die Trennlinie der Schicht 0 m-mal kreuzen könnte $\sim \frac{n^2}{m}\cdot m = \Theta(n^2)$ Verlängerung verursachen und das ist kein $\varepsilon \cdot OPT$ !]

Beispiel — alle (vereinigten) Trennlinien der Schicht 1 und eine optimale Rundreise die sie häufig kreuzt

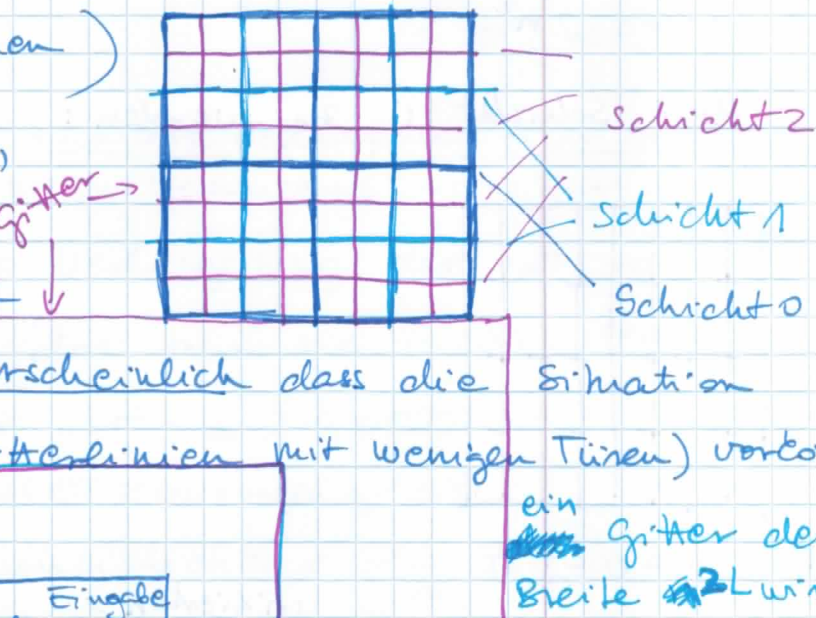


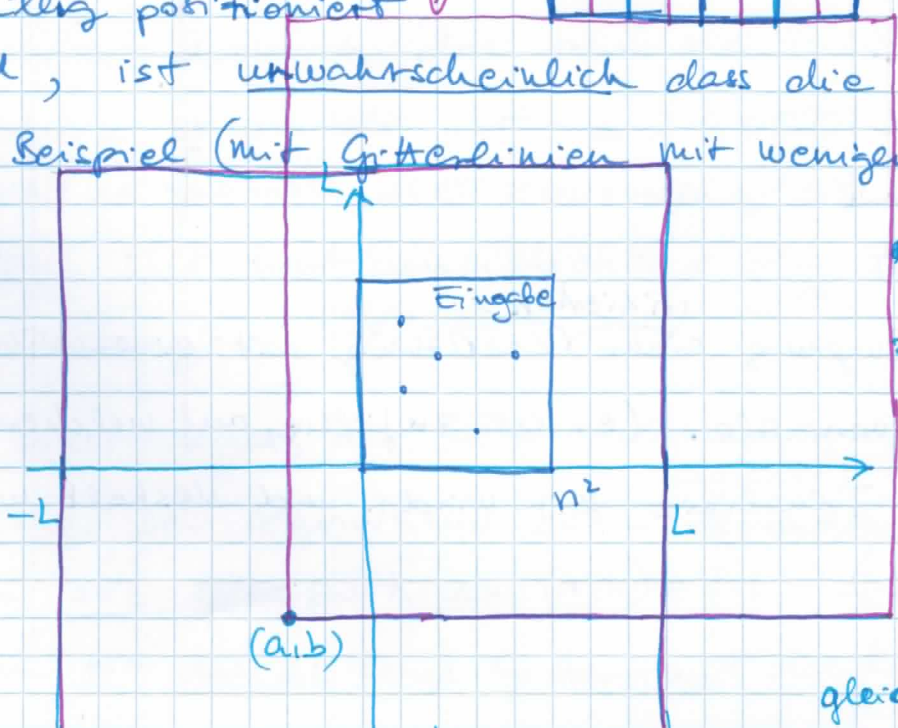
ABER in diesem Beispiel haben wir offensichtlich Pech gehabt bei der Positionierung der Trennlinien!

Wichtig: es gibt immer (viele) vorteilhafte Positionierungen der Trennlinien durch Verschiebung des ganzen Gitters.

Idee: [Die Gesamtanzahl aller Kreuzungen mit Trennlinien ist nicht mehr als 2-OPT (bitte ausprobieren wieviele (ganzzahlig positionierte) Trennlinien eine Strecke der Länge 5 kreuzen kann)]

wenn das Gitter aller Trennlinien (mit ihren Schichten zusammen) zufällig verschoben, bzw. am Anfang zufällig positioniert wird, ist unwahrscheinlich dass die Situation im Beispiel (mit Gitterlinien mit wenigen Türen) vorkommt.



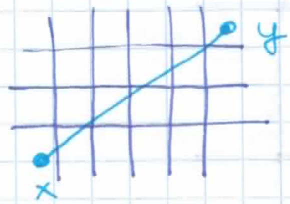

ein Gitter der Breite 2L wird zufällig positioniert (für $L > n^2$, L Zweierpotenz) s.d. seine Ecke (a,b) zufällig in $[-L,0]\times[-L,0]$ gleichverteilt fällt.

A14.

für jede $(a,b) \in [-L,0] \times [-L,0]$ enthält es die ganze Eingabe

Die präzise Analyse ist wie folgt:

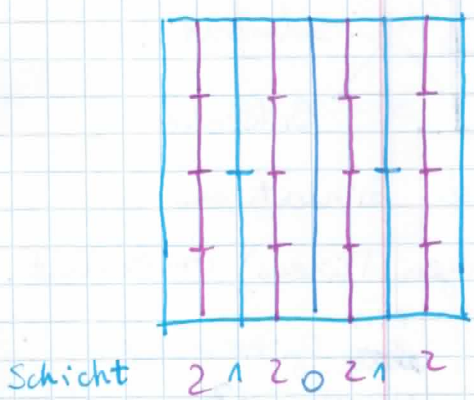
— Eine Kante der Rundreise der Länge $d(x,y)$




kreuzt maximal $d(x,y)$ horizontale und maximal $d(x,y)$ vertikale Trennlinien $\Longrightarrow$ die optimale Rundreise hat maximal 2OPT Kreuzungen mit Trennlinien insgesamt.

— Dank der zufälligen Positionierung des Gitters hat jede ganzzahlige Linie die Wahrscheinlichkeit

$\frac{2^i}{2L} \approx \frac{2^i}{2n^2}$ um eine (Vereinigung von) Trennlinie(n) der Schicht $i$ zu werden:




— Sei die Verlängerung einer fixierten Kreuzung mit einer Trennlinie eine Zufallsvariable. (es ist zufällig, auf welcher Schicht die betroffene Trennlinie sein wird, und deshalb wieviele Türen sie hat)

Da eine Linie mit Wahrscheinlichkeit $\frac{2^i}{2n^2}$ eine Trennlinie der Schicht $i$ wird, ist die Verlängerung (höchstens) $\frac{2n^2}{2^i \cdot m}$ lang mit Wahrscheinlichkeit $\frac{2^i}{2 \cdot n^2}$. Der Erwartungswert der Verlängerung wegen einer Kreuzung ist somit

$$E\left(\begin{array}{l}\text{Verlängerung einer konkreten Kreuzung} \\ \text{der opt. Rundreise mit einer} \\ \text{ganzzahliger Linie}\end{array}\right) =$$

max Anzahl der Schichten

$$= \sum_{i=0}^{2\log n} \underbrace{\frac{2n^2}{2^i \cdot m}}_{\substack{\text{Verlängerung} \\ \text{mit dieser} \\ \text{Wahrscheinlichkeit}}} \cdot \underbrace{\frac{2^i}{2n^2}}_{\text{Wahrscheinlichkeit}} = \sum_{i=0}^{2\log n} \frac{1}{m} = \frac{2\log n}{m} = \frac{2\log n \cdot \varepsilon}{2\log n} = \varepsilon$$

$$E(\text{Verlängerung für alle Kreuzungen}) \leq \varepsilon \cdot \#\text{Kreuzungen} \leq$$

$$\leq \varepsilon \cdot 2 \cdot OPT$$

es folgt: Thm: Die erwartete Verlängerung einer konkreten Kreuzung mit einer Linie wegen Verschiebung in eine Tür ist $< \varepsilon$.

**Theorem**: Die erwartete Verlängerung der optimalen Rundreise dadurch dass man sie in eine legale Rundreise verwandelt ist $\leq 2 \cdot \varepsilon \, OPT$
(weil es maximal $2OPT$ Kreuzungen gibt)

**Korollar**: Mit Wahrscheinlichkeit $\geq \frac{1}{2}$ ist die Verlängerung $\leq 4 \cdot \varepsilon \cdot OPT$

(sonst wäre der Erwartungswert größer als $2\varepsilon OPT$)

**Bemerkung**: Der Algorithmus kann derandomisiert werden indem alle Positionen $(a,b) \in [-L,0] \times [-L,0]$ ausprobiert werden für das $2L$ lange Gitter, und die kürzeste Lösung gewählt wird.

A16.

# Arora's PTAS (Zusammenfassung)

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

- verschiebe, skaliere und runde die Instanz, s.d. $[0,n^2]\times[0,n^2]$ das kleinste umschließende Quadrat ist.
- sei $L > n^2$ Zweierpotenz und $Q_0$ ein Quadrat der Länge $2L$ mit Eckpunkt $(a,b)\in[-L,0]\times[-L,0]$ zufällig (gleichverteilt) gewählt.

- definiere den 4-ären Baum $B$ der Teilquadrate von $Q_0$, und $m$ Türen auf jeder Trennlinie
$$m > \frac{2\cdot\log n}{\varepsilon}$$

- bestimme für jeden Knoten (Quadrat) $Q$ im $B$ bottom-up für jedes Besuchsmuster $\tilde{T}$
  $L_Q(\tilde{T})$ = die kürzeste Länge einer legalen partiellen Rundreise konsistent mit diesem Besuchsmuster
  ~~[illegible]~~ $L_{Q_0}$ ist die optimale Länge einer legalen Rundreise
- bestimme die optimale legale Rundreise top-down

Theorem: - Arora's Algorithmus findet eine Rundreise mit erwarteter Länge $\leq (1+2\varepsilon)\,\mathrm{OPT}$.
- Mit Wahrscheinlichkeit $\geq \frac{1}{2}$ hat die Rundreise Länge $\leq (1+4\varepsilon)\,\mathrm{OPT}$.
- Die Laufzeit ist $O(n\cdot\log n)\cdot n^{O(\frac{1}{\varepsilon})}$