Hamiltonkreise und Eulertouren

- Sei G = (V, E) ein Graph.

- **Eulertour:**

  - Ein geschlossener Weg in G, der jede Kante genau einmal enthält.

- **Hamiltonkreis:**

  - Ein Kreis in G, der jeden Knoten genau einmal enthält.

Eulertour: Charakterisierung

Satz: Ein zusammenhängender Graph G = (V, E) enthält eine Eulertour

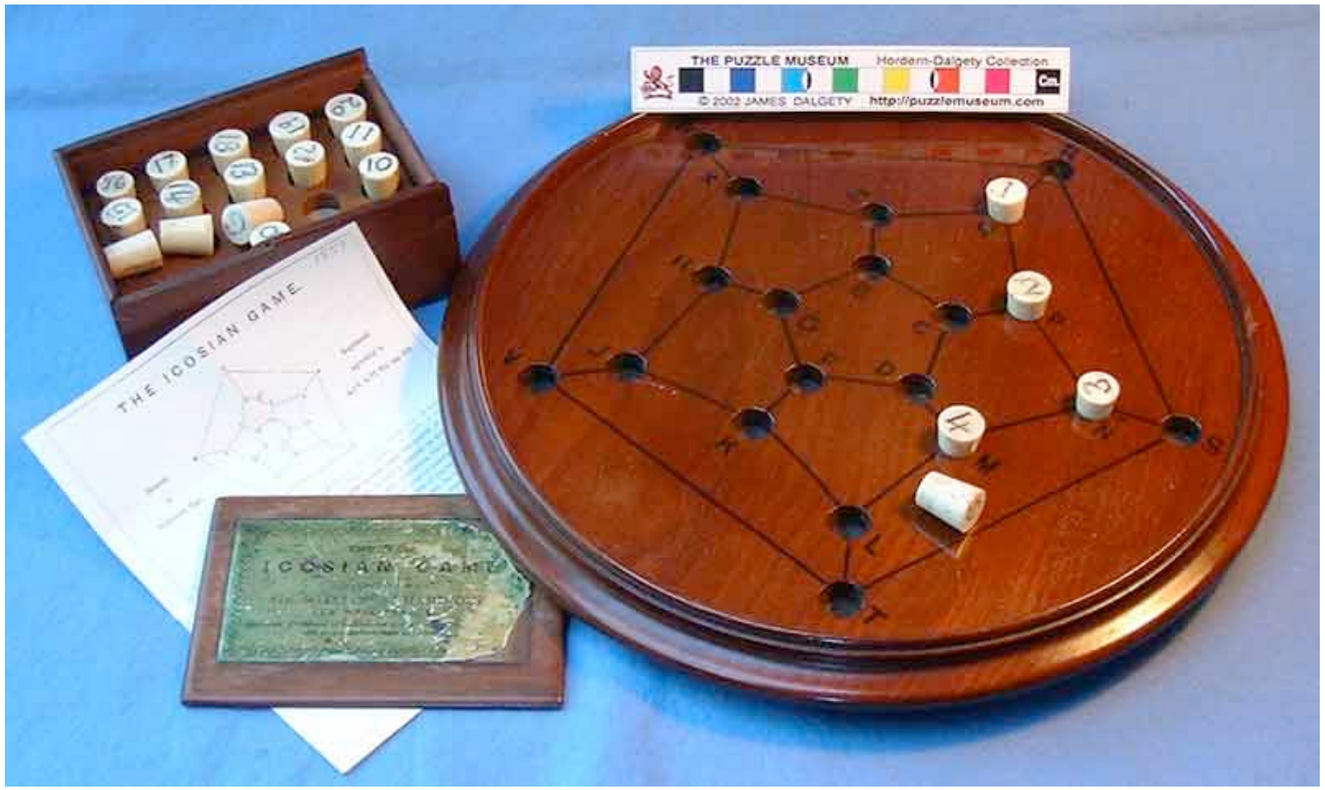
gdw. der Grad jedes Knotens gerade ist.

… und eine solche kann man in O(|E|) Zeit finden

Bem: Sind in einem zusammenhängenden Graphen G = (V, E) alle *bis auf zwei* Knotengrade gerade, so enthält der Graph einen Eulerweg

(Ein Eulerweg ist ein Weg, der alle Kanten des Graphen genau einmal enthält — aber nicht notwendigerweise im gleichen Knoten startet und endet.)

Hamiltonkreis

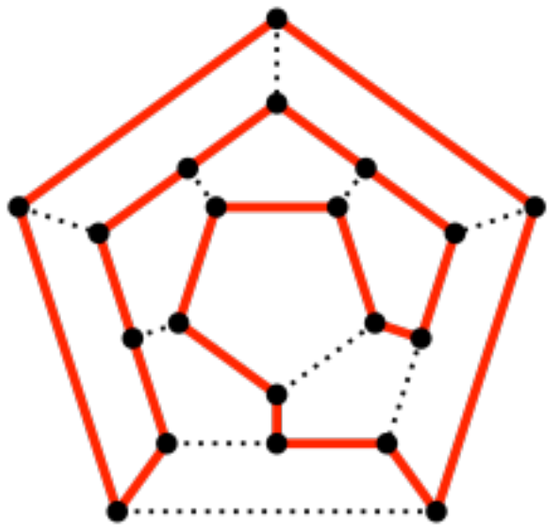


Ikosaeder Spiel

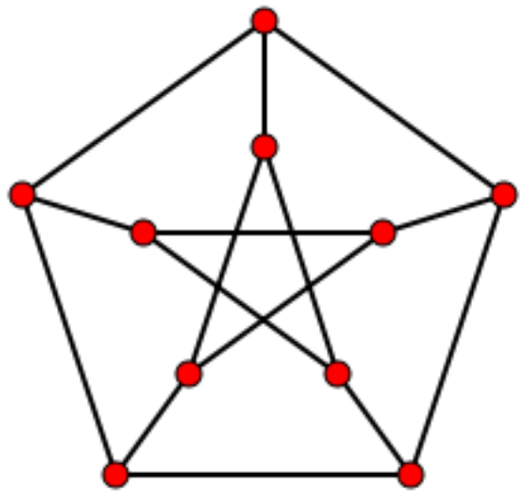
Sir William Hamilton
(1805 - 1865)

Gesucht: **Hamiltonkreis**

Hamiltonkreis - Beispiele

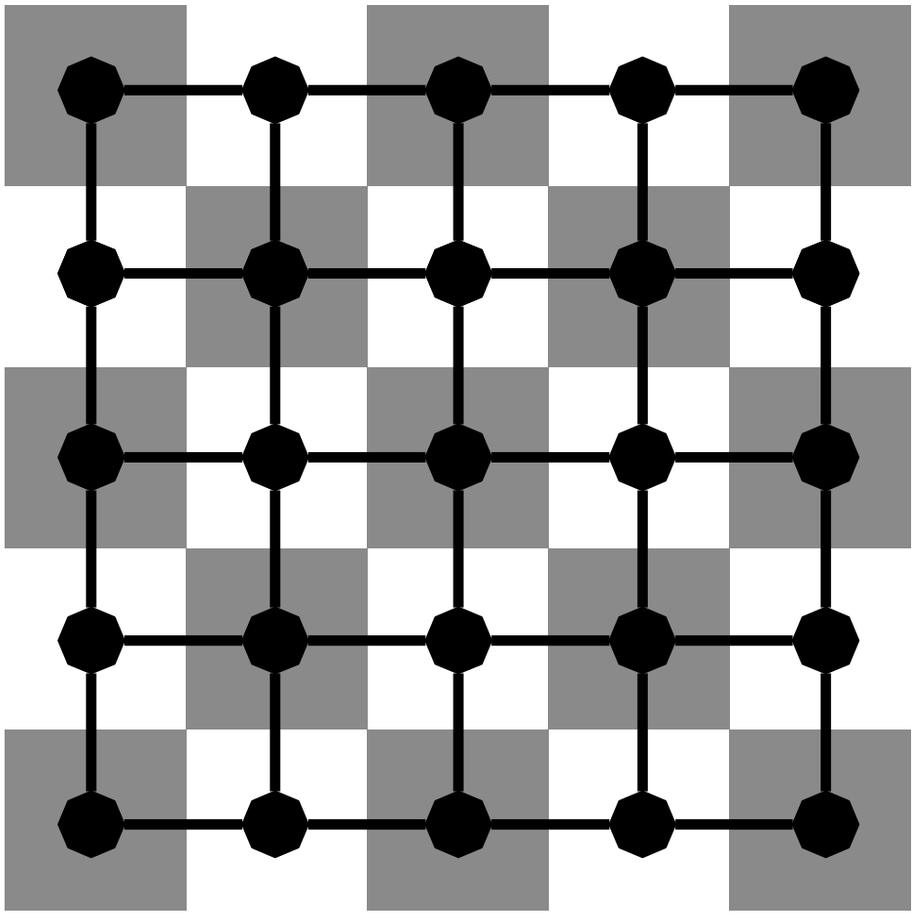


Ikosaeder



Petersengraph

Hamiltonkreis - Beispiele



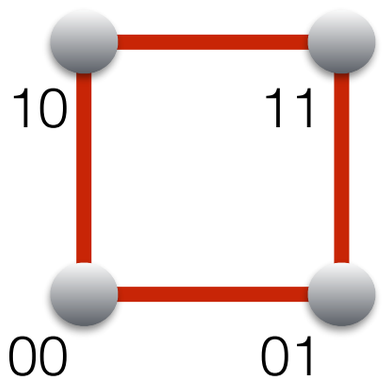
Satz: Seien m,n ≥ 2.

Ein n×m Gitter enthält einen Hamiltonkreis **gdw** n*m gerade ist.

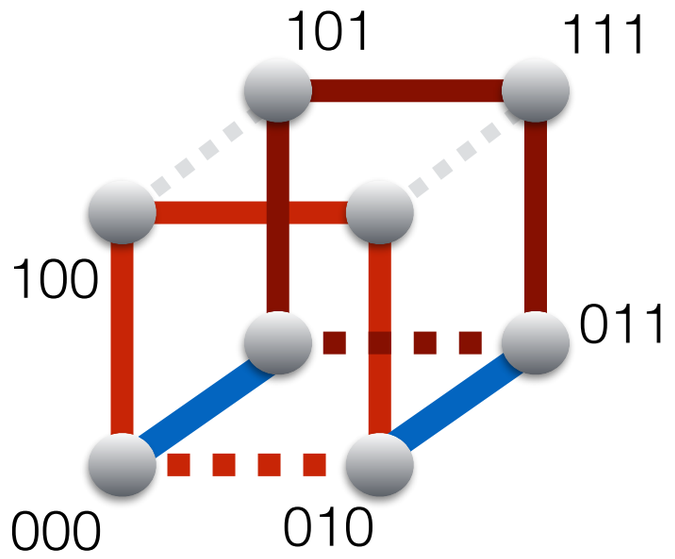
Satz: Ein bipartiter Graph $G = (A \cup B, E)$ mit |A| ≠ |B| kann keinen Hamiltonkreis enthalten.

Hamiltonkreis im Hyperwürfel: Gray-Code

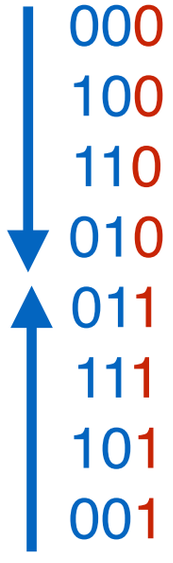
d=2:



d=3:



00
10
11
01



analog für d ≧ 4

Satz von Dirac

Satz: (Dirac 1952)

Jeder Graph $G = (V, E)$ mit $|V| \geq 3$ und Minimalgrad

$\delta(G) \geq |V|/2$ enthält einen Hamiltonkreis.

Paul Dirac
(1902-1984)

Hamiltonkreis

Wie entscheidet man (effizient),
ob ein Graph einen Hamiltonkreis enthält?

Brute Force: $\approx n!$

Ziel für heute: $\approx 2^n$

Zum Vergleich:

$$\log_2(n!) = \Theta(n \log n)$$

$$\log_2(2^n) = n$$

Hamiltonkreis

Wie entscheidet man (effizient),
ob ein Graph einen Hamiltonkreis enthält?

Brute Force: ≈ n!

Ziel für heute: ≈ 2^n

European Commission

For now, general purpose AI models that were trained using **a total computing power of more than 10^25 FLOPs** are considered to carry systemic risks

10^25 ≈

25! ≈ 1.6 e25

2^84 ≈ 1.9 e25

Hamiltonkreise mit DP

Für alle $S \subseteq [n]$ mit $1 \in S$ und alle $x \in S$ mit $x \neq 1$:

„Idee“

$$P_{S,x} := \begin{cases} 1, & \text{es gibt einen 1-x-Pfad, der genau die Knoten aus S enthält} \\ 0, & \text{sonst} \end{cases}$$

Dynamische Programmierung:
Laufzeit: $\approx n^2 2^n$
Speicherplatz: $\approx n\, 2^n$

Initialisierung:

$\forall x \in \{2,\ldots,n\}$: für $S = \{1,x\}$ setze $P_{S,x} = 1$ gdw. $\{1,x\} \in E$

Schleife:

for all s = 3 to n
  for all $S \subseteq [n]$ mit $1 \in S$ und $|S| = s$:
    for all $x \in S$ mit $x \neq 1$:

Rekursion

$$P_{S,x} = \max \{ P_{S\setminus\{x\},y} : y \in N(x) \cap S,\ y \neq 1 \}$$

Ausgabe: G enthält Hamiltonkreis gdw $\exists\, x \in N(1)$ mit $P_{[n],x} = 1$

How to brand your research?

The 1950s were not good years for mathematical research. [The Secretary of Defense] had a pathological fear and hatred of the word “research".

What title, what name, could I choose? […] I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying.
[…] It also has a very interesting property as an adjective, and that is it's impossible to use the word dynamic in a pejorative sense.

— Richard Bellman, Eye of the Hurricane: An Autobiography (1984)

Eulertour vs Hamiltonkreis

Satz: Ein zusammenhängender Graph $G = (V, E)$ enthält eine Eulertour

gdw. der Grad jedes Knotens gerade ist.

… und eine solche kann man in $O(|E|)$ Zeit finden

Satz: Das Problem

„Gegeben ein Graph $G = (V, E)$, enthält G einen Hamiltonkreis?“

kann man in Zeit $O(|V|^2 \cdot 2^{|V|})$ entscheiden und, falls ja, einen solchen finden.

Eulertour vs Hamiltonkreis

Satz: Ein zusammenhängender Graph G = (V, E) enthält eine Eulertour

gdw. der Grad jedes Knotens gerade ist.

… und eine solche kann man in O(|E|) Zeit finden

Satz: (Karp 1972)

Das Problem

„Gegeben ein Graph G = (V, E), enthält G einen Hamiltonkreis?“

ist NP-vollständig.

Exkurs

(Vorgriff auf Vorlesung Theoretische Informatik)

Komplexitätstheorie

polynomiell

Eulertour

Hamiltonkreis

P = *effizient entscheidbare Probleme*

NP = *(einseitig) effizient verifizierbare Probleme*

nichtdeterministisch polynomiell

$P \stackrel{?}{=} NP$ → *1 Million US-$ (Clay-Foundation)*

[eines von sieben Millennium-Problemen]

Ein Problem π aus NP heisst NP-vollständig, falls gilt

$$\pi \in P \quad \Rightarrow \quad P = NP$$

P und NP

Es gibt sehr viele NP-vollständige Probleme:

- Hamiltonkreis
- Rucksackproblem
- Clique: Gibt es in einem Graphen k paarweise benachbarte Knoten?
- Nullstelle mod n: Hat ein Polynom mod n eine Nullstelle?
- Satisfiability: hat eine logische Formel eine Lösung?
- ....

... sehr sehr viele:

Classic Nintendo Games are (Computationally) Hard

Greg Aloupis* Erik D. Demaine† Alan Guo†‡ Giovanni Viglietta§

February 10, 2015

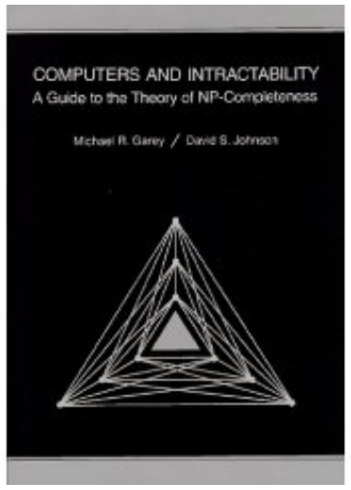
Abstract

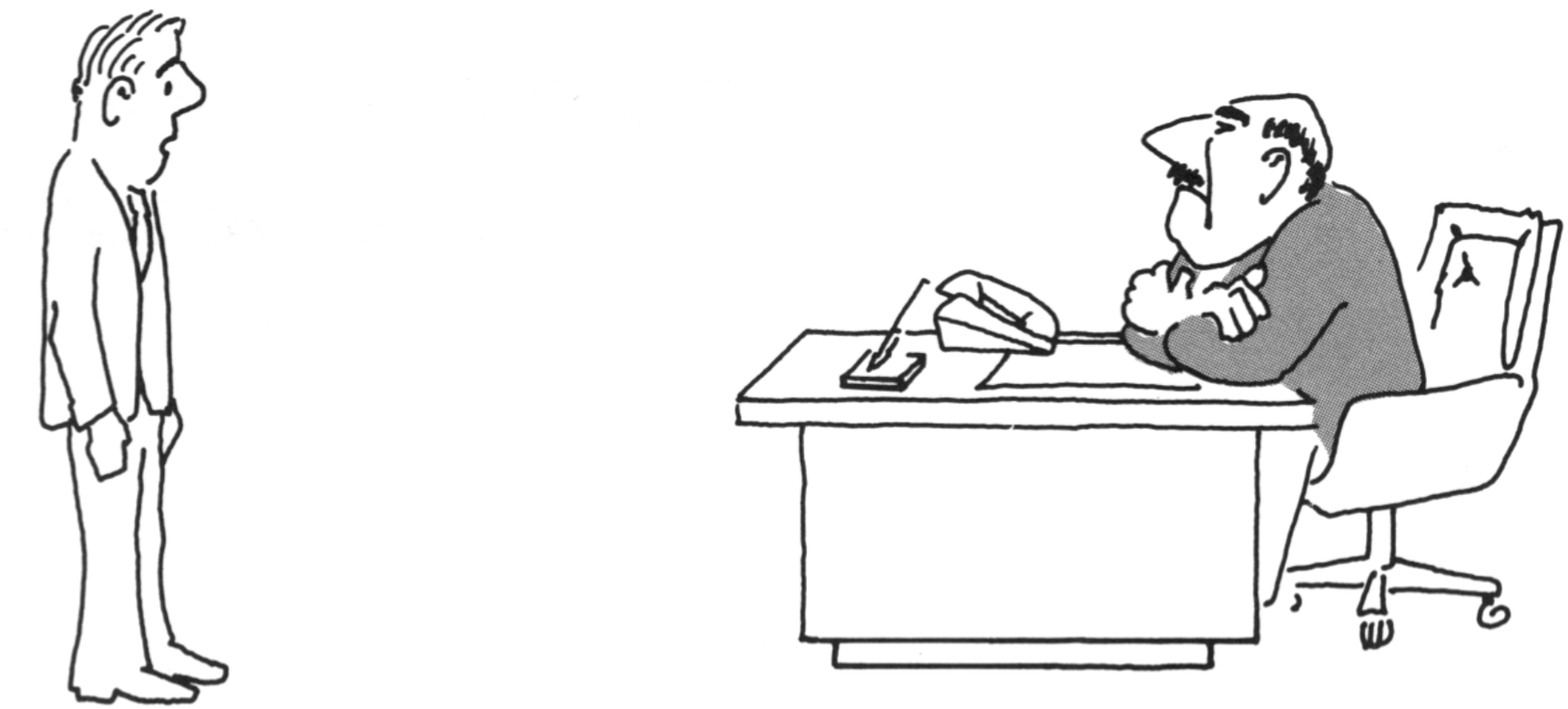
We prove NP-hardness results for five of Nintendo's largest video game franchises: Mario, Donkey Kong, Legend of Zelda, Metroid, and Pokémon. Our results apply to generalized versions of Super Mario Bros. 1–3, The Lost Levels, and Super Mario World; Donkey Kong Country 1–3; all Legend of Zelda games; all Metroid games; and all Pokémon role-playing games. In addition, we prove PSPACE-completeness of the Donkey Kong Country games and several Legend of Zelda games.

1 Introduction

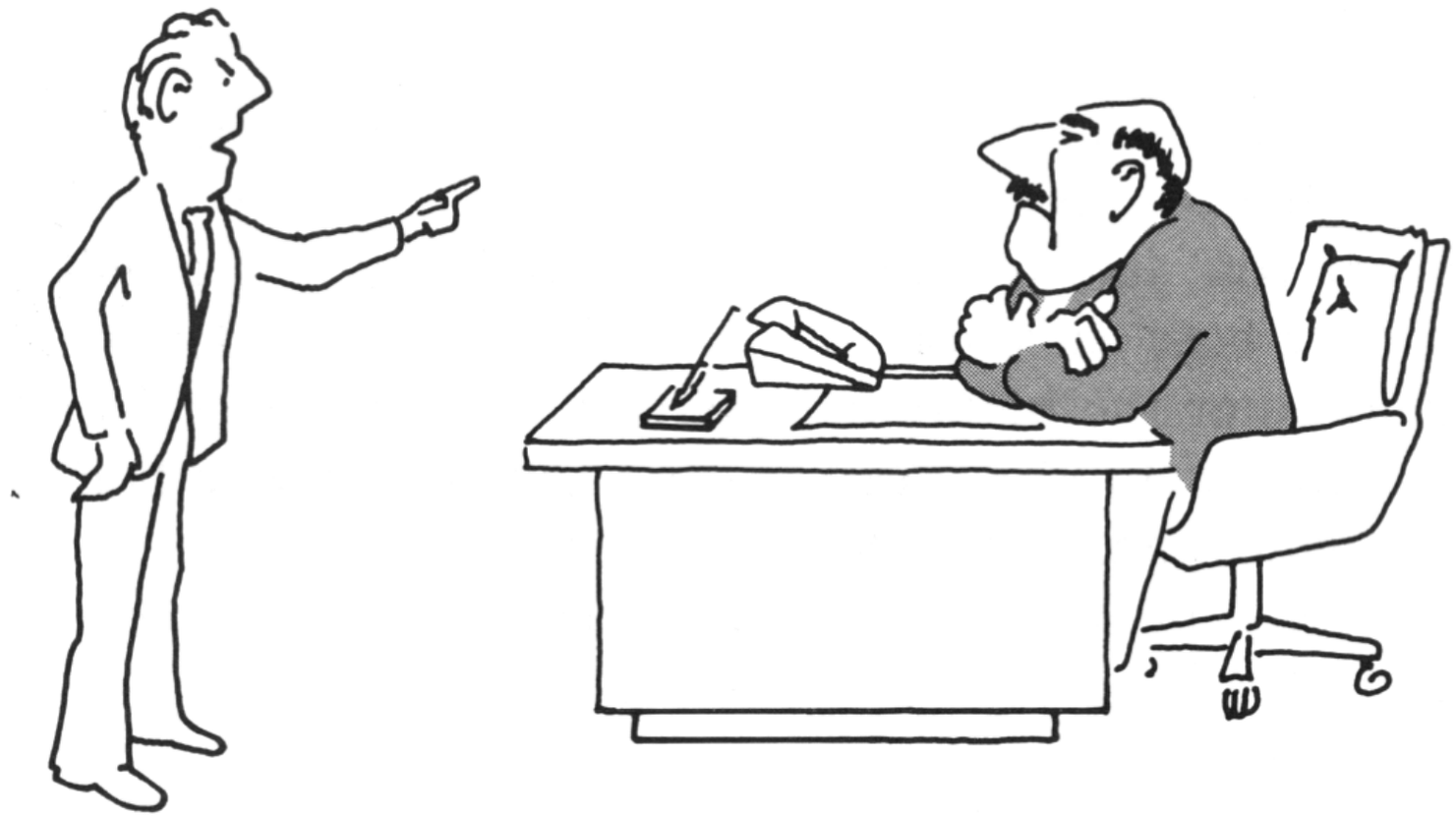
A series of recent papers have analyzed the computational complexity of playing many different video games [1, 4, 5, 6], but the most well-known classic Nintendo games have yet to be included among these results. In this paper, we analyze some of the best-known Nintendo games of all time: Mario, Donkey Kong, Legend of Zelda, Metroid, and Pokémon. We prove that it is NP-hard, and in some cases PSPACE-hard, to play generalized versions of most games in these series. In

Komplexitätstheorie





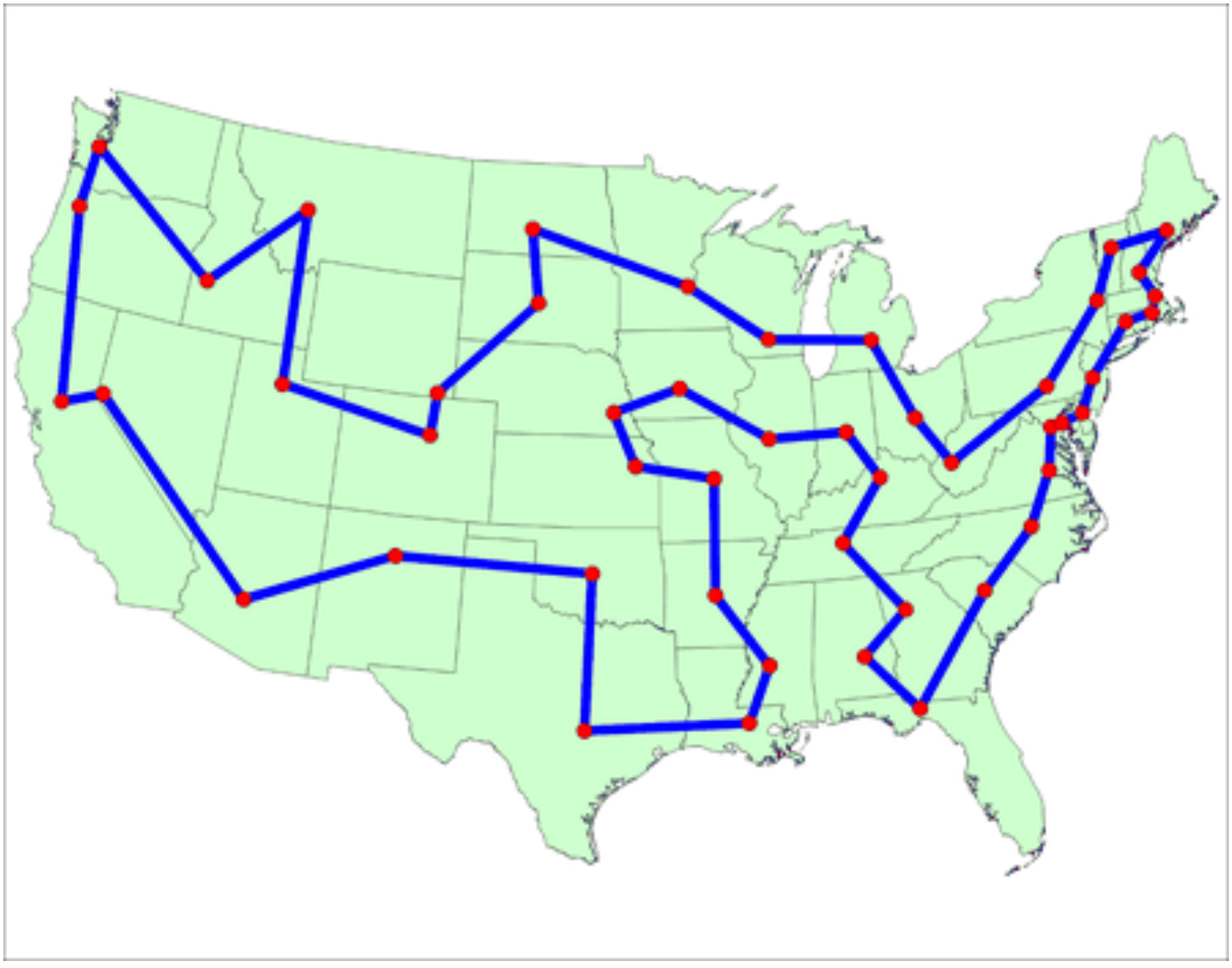
"I can't find an efficient algorithm, I guess I'm just too dumb."



"I can't find an efficient algorithm, because no such algorithm is possible!"

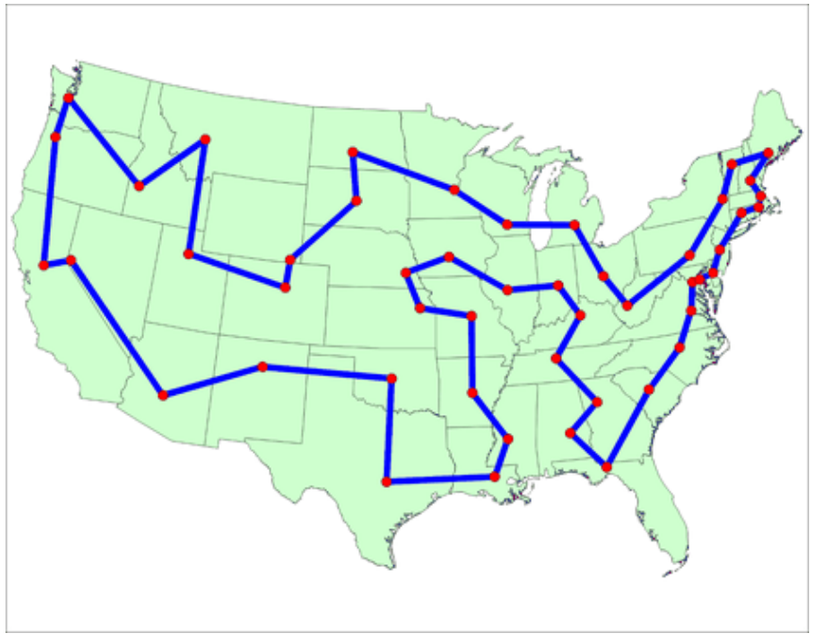
"I can't find an efficient algorithm, but neither can all these famous people."

Hamiltonkreis - Beispiele



The Traveling Salesman Problem — Das Problem des Handlungsreisenden

Traveling Salesman Problem



Gegeben: vollständiger Graph auf n Knoten,
Distanzen zw. je zwei Knoten: $\ell : \binom{[n]}{2} \to \mathbb{R}$

Gesucht? Kürzeste Rundreise:

$$\min_{H:\ \text{Hamiltonkreis}} \sum_{e \in E(H)} \ell(e)$$

Gesucht: Rundreise kurzer als X (gegeben X)

Traveling Salesman Problem

Das Traveling Salesman Problem (TSP) ist NP-vollständig:

Graph $G=([n], E)$ $\Longleftrightarrow$ vollst Graph $K_n=([n],\binom{[n]}{2})$

mit Längenfunktion $\ell : \binom{[n]}{2} \to \{0, 1\}$

so dass $\ell(\{x, y\}) = \begin{cases} 0, & \text{falls } \{x, y\} \in E \\ 1, & \text{sonst.} \end{cases}$

Dann gilt:

G enthält Hamiltonkreis $\Longleftrightarrow$ $\text{opt}(K_n,\ell) = 0$

Hieraus folgt auch: für kein $C > 0$ kann es einen polynominellen C-Approximationsalgorithmus geben (ausser es gilt P=NP).

List of NP-complete problems

3 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

This is a dynamic list and may never be able to satisfy particular standards for completeness. You can help by adding missing items with reliable sources.

This is a list of some of the more commonly known problems that are NP-complete when expressed as decision problems. As there are hundreds of such problems known, this list is in no way comprehensive. Many problems of this type can be found in Garey & Johnson (1979).

Graphs and hypergraphs [edit]

Graphs occur frequently in everyday applications. Examples include biological or social networks, which contain hundreds, thousands and even billions of nodes in some cases (e.g. Facebook or LinkedIn).

- 1-planarity[1]
- 3-dimensional matching[2][3]:SP1
- Bandwidth problem[3]:GT40
- Bipartite dimension[3]:GT18
- Capacitated minimum spanning tree[3]:ND5
- Route inspection problem (also called **Chinese postman problem**) for mixed graphs (having both directed and undirected edges). The program is solvable in polynomial time if the graph has all undirected or all directed edges. Variants include the rural postman problem.[3]:ND25,ND27
- Clique cover problem[2][3]:GT17
- Clique problem[2][3]:GT19
- Complete coloring, a.k.a. achromatic number[3]:GT5
- Cycle rank
- Degree-constrained spanning tree[3]:ND1
- Domatic number[3]:GT3
- Dominating set, a.k.a. domination number[3]:GT2
  - NP-complete special cases include the edge dominating set problem, i.e., the dominating set problem in line graphs. NP-complete variants include the connected dominating set problem and the maximum leaf spanning tree problem.[3]:ND2
- Feedback vertex set[2][3]:GT7
- Feedback arc set[2][3]:GT8
- Graph coloring[2][3]:GT4
- Graph homomorphism problem[3]:GT52
- Graph partition into subgraphs of specific types (triangles, isomorphic subgraphs, Hamiltonian subgraphs, forests, perfect matchings) are known NP-complete. Partition into cliques is the same problem as coloring the complement of the given graph. A related problem is to find a partition that is optimal terms of the number of edges between parts.[3]:GT11,GT12,GT13,GT14,GT15,GT16,ND14
- Grundy number of a directed graph.[3]:GT56
- Hamiltonian completion[3]:GT34
- Hamiltonian path problem, directed and undirected.[2][3]:GT37,GT38,GT39
- Graph intersection number[3]:GT59
- Longest path problem[3]:ND29
- Maximum bipartite subgraph or (especially with weighted edges) maximum cut.[2][3]:GT25,ND16
- Maximum common subgraph isomorphism problem[3]:GT49
- Maximum independent set[3]:GT20
- Maximum Induced path[3]:GT23
- Minimum maximal independent set a.k.a. minimum independent dominating set[4]
  - NP-complete special cases include the minimum maximal matching problem,[3]:GT10 which is essentially equal to the edge dominating set problem (see above).
- Metric dimension of a graph[3]:GT61
- Metric k-center
- Minimum degree spanning tree
- Minimum k-cut
- Minimum k-spanning tree
- Steiner tree, or Minimum spanning tree for a subset of the vertices of a graph.[2] (The minimum spanning tree for an entire graph is solvable in polynomial time.)
- Modularity maximization[5]
- Monochromatic triangle[3]:GT6
- Pathwidth,[6] or, equivalently, interval thickness, and vertex separation number[7]
- Rank coloring
- k-Chinese postman
- Shortest total path length spanning tree[3]:ND3
- Slope number two testing[8]

Wir kennen Hunderte (Tausende?) von Problemen die NP-vollständig sind!

Deswegen ein sehr wichtiges Werkzeug um zu erklären warum wir für viele Probleme keine effiziente Algorithmen haben

"I can't find an efficient algorithm, but neither can all these famous people."

Komplexitätstheorie

Haskell Program A

```
foldl (+) 0 [1,2,3,4,5]
```

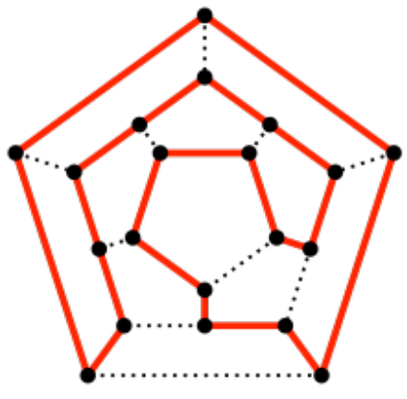
A kann als B kodiert werden

Python Program B

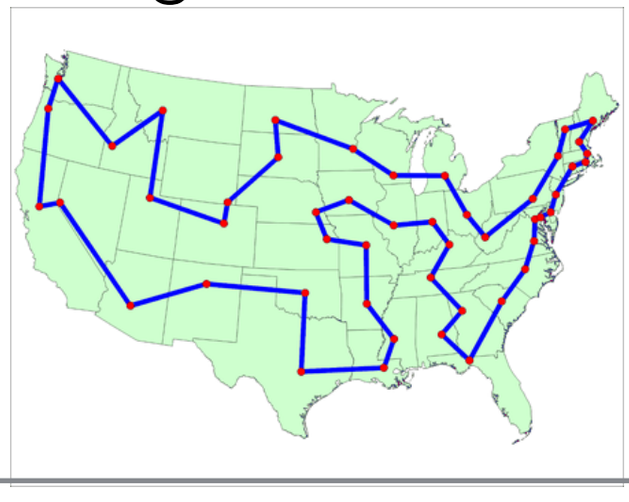
```
from functools import reduce
xs = [1, 2, 3, 4, 5]
reduce(lambda x, y: x+y, [1, 2, 3, 4, 5],0)
```

0+1+2+3+4+5 = 15

Hamiltonkreis “Program A”



TSP “Program B”



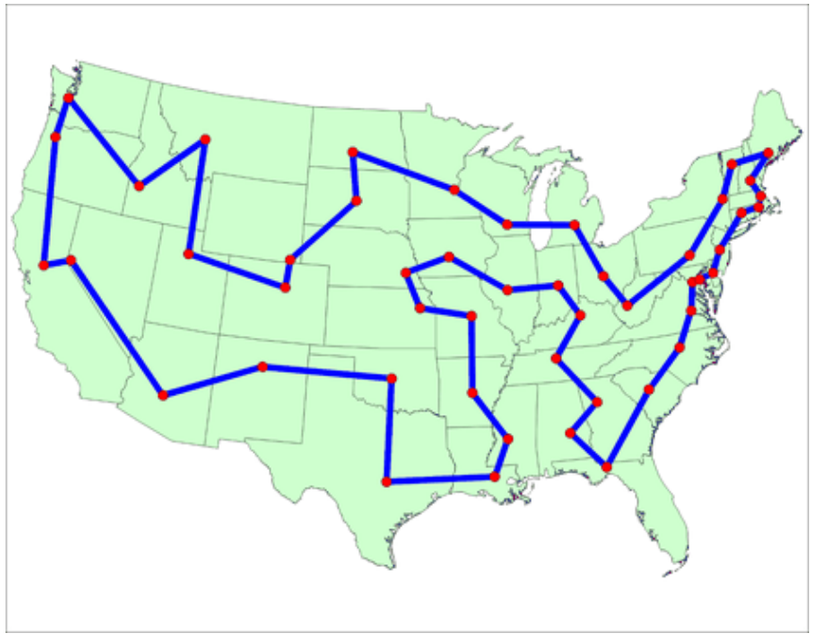
Deswegen:
Weil das Hamiltonkreis-Problem NP-vollständig ist,
ist auch das TSP-Problem

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt

Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



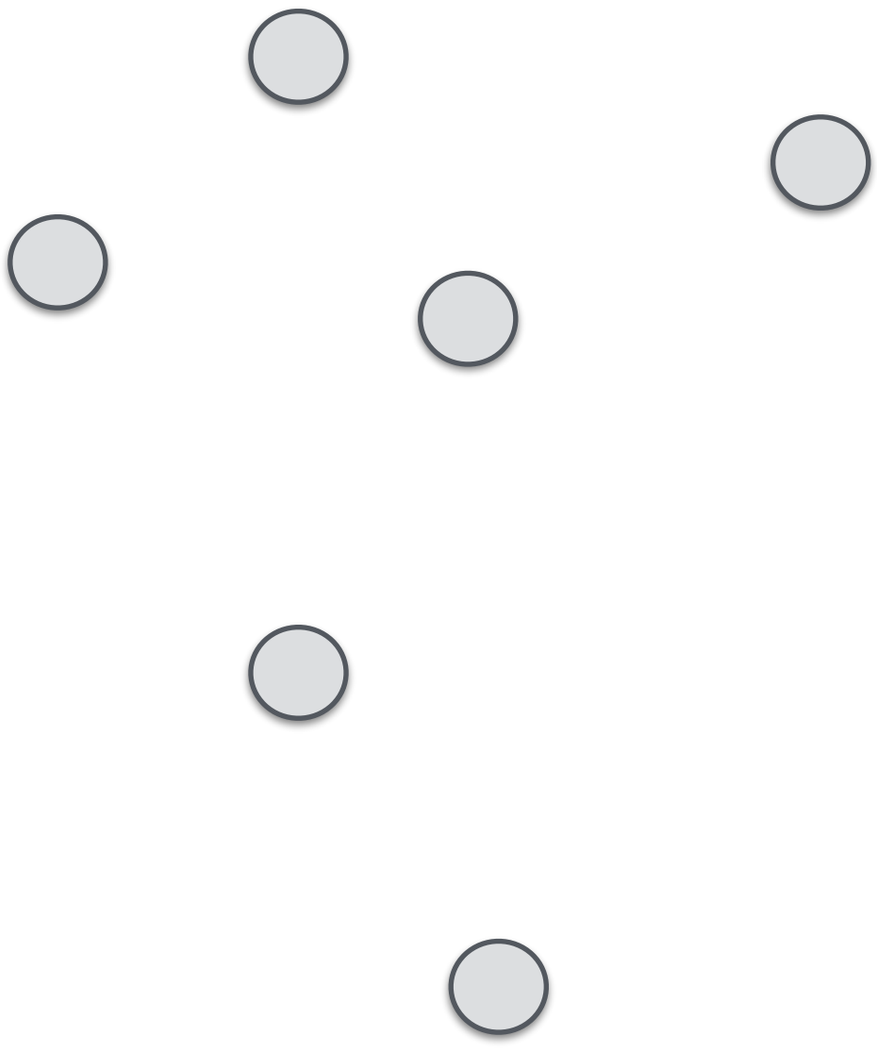
Bsp: $\ell(e) \in \{1,2\}$ für alle $e \in \binom{[n]}{2}$

aus MST kann man eine 2-Approximation ableiten

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

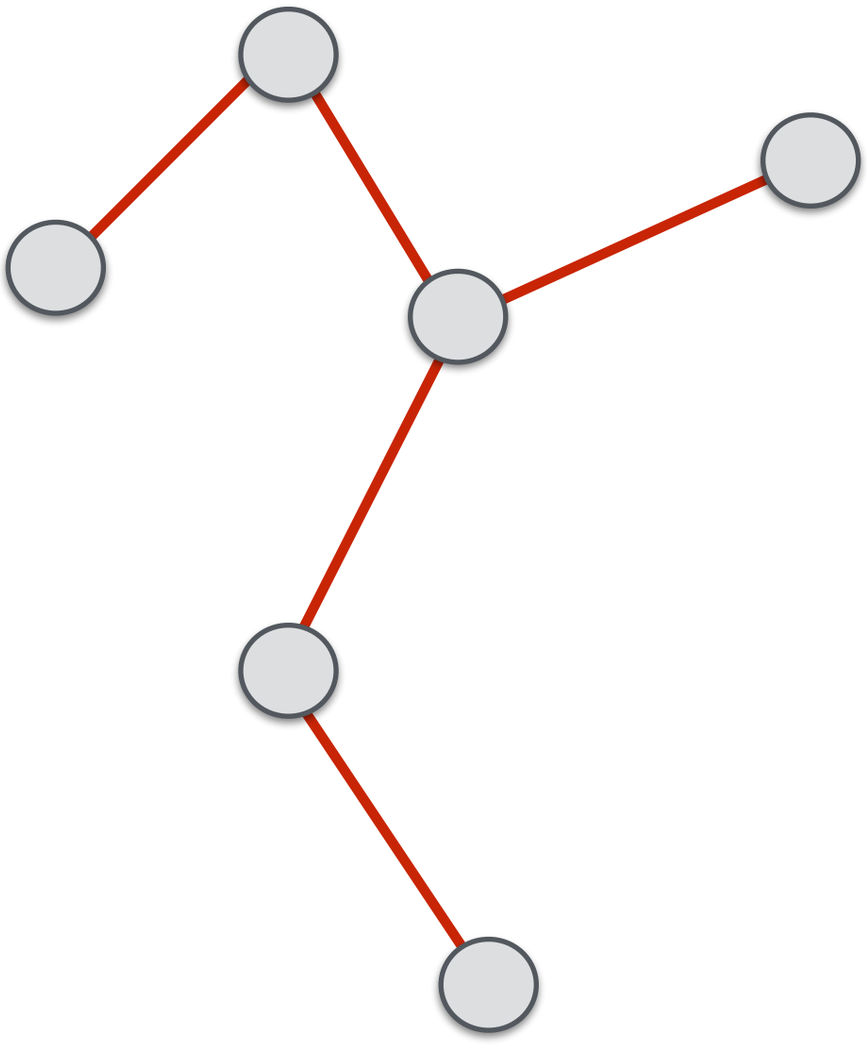
$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



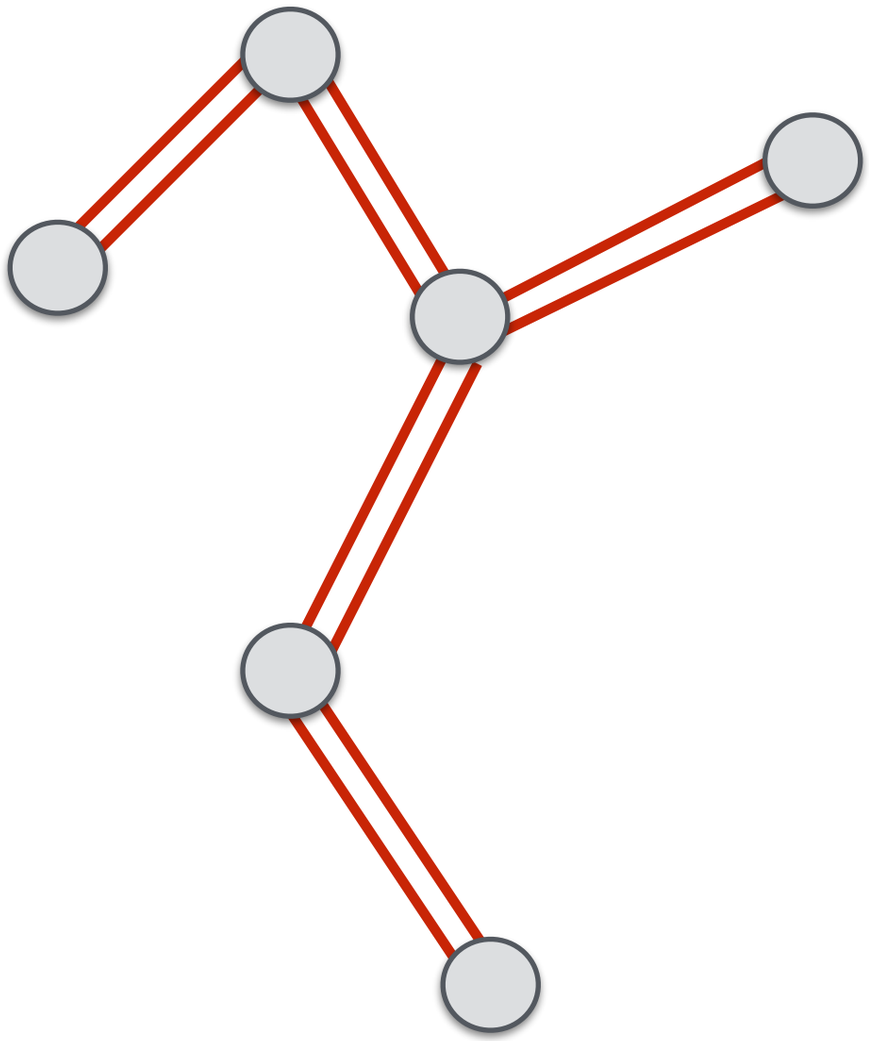
1. Bestimme minimalen Spannbaum **T**

es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

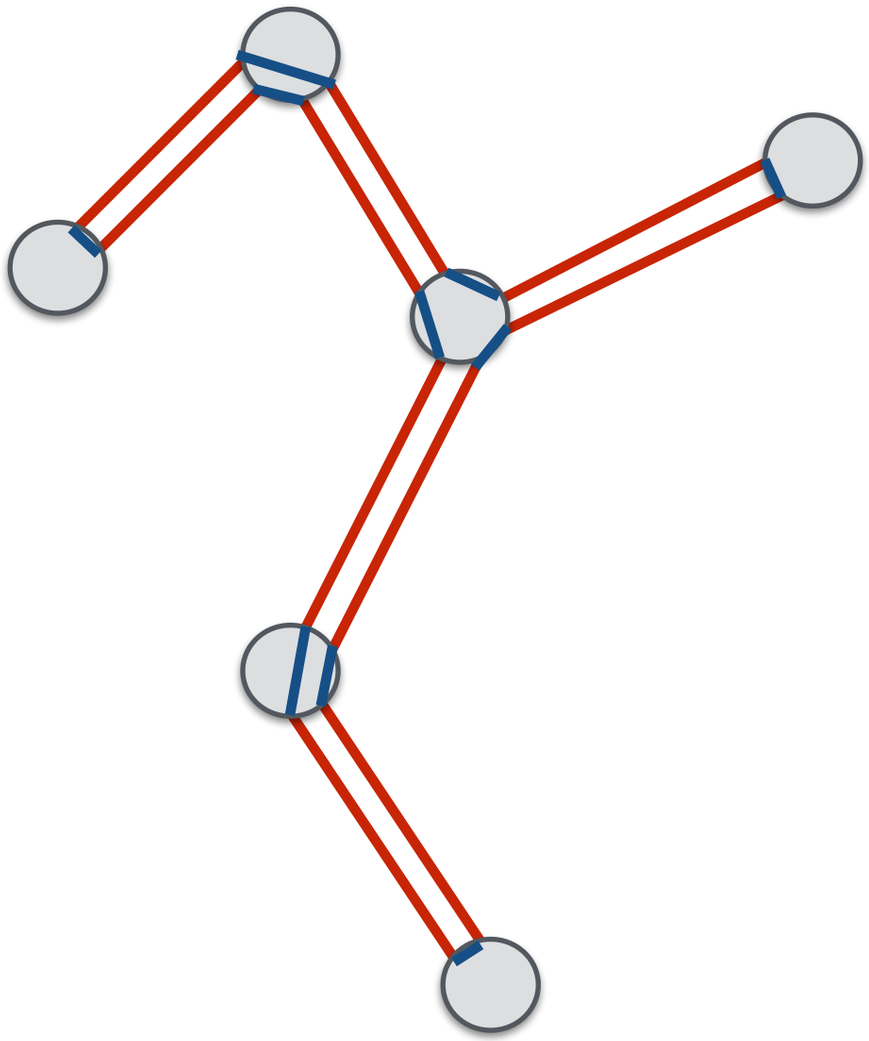
2. verdopple alle Kanten von **T**

es gilt: $2\ell(T) \leq 2\text{opt}(K_n, \ell)$

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

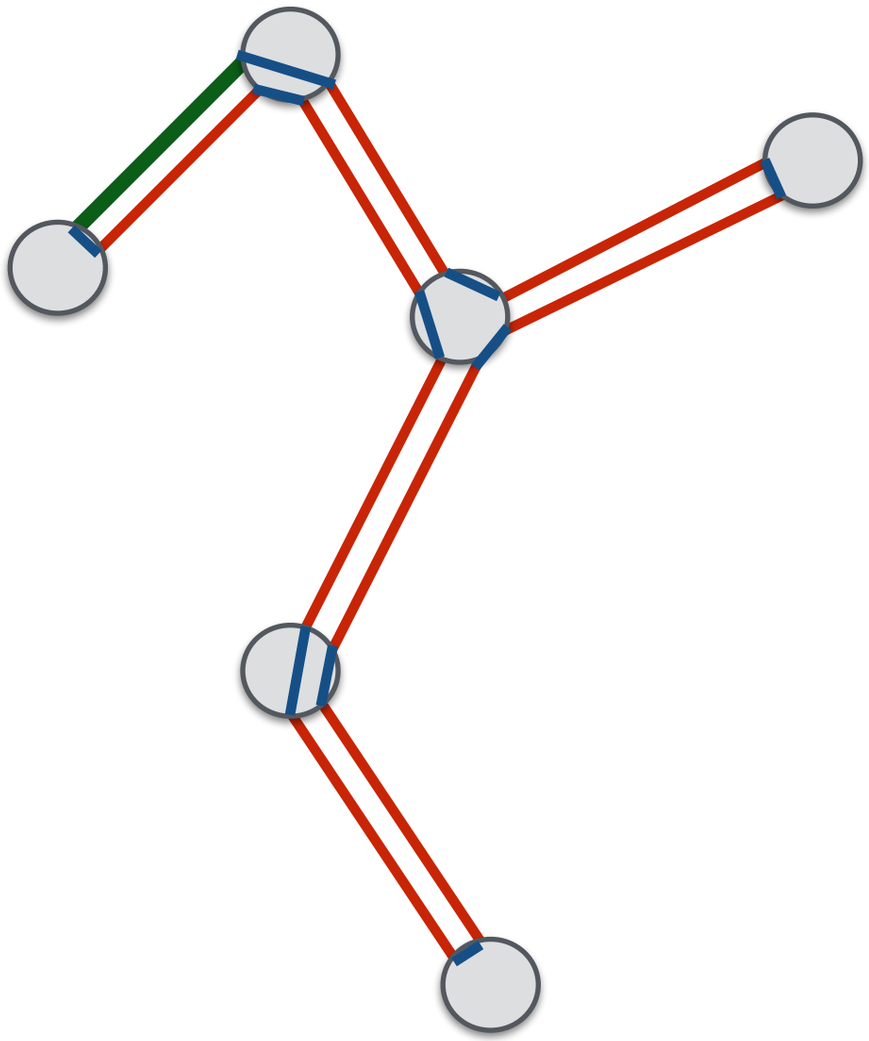
3. bestimme Eulertour **W**

   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2\text{opt}(K_n, \ell)$

3. bestimme Eulertour **W**

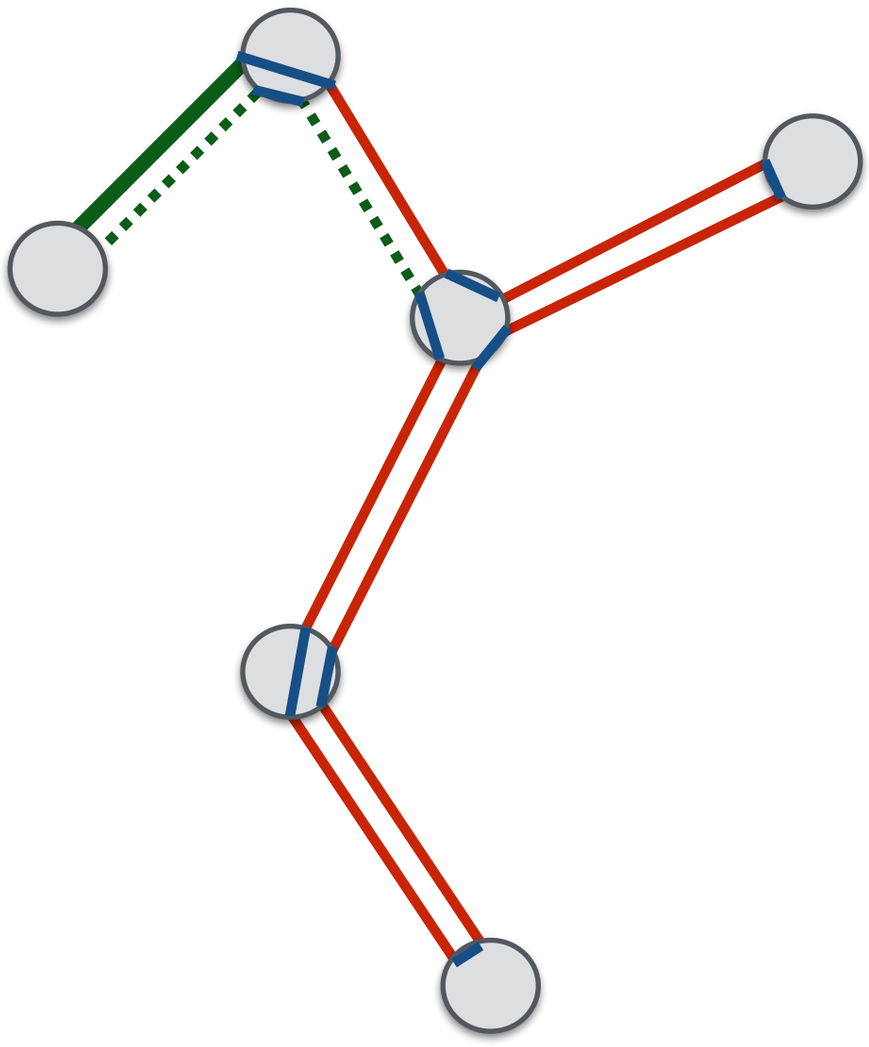
   es gilt: $\ell(W) = 2\ell(T) \leq 2\text{opt}(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

3. bestimme Eulertour **W**

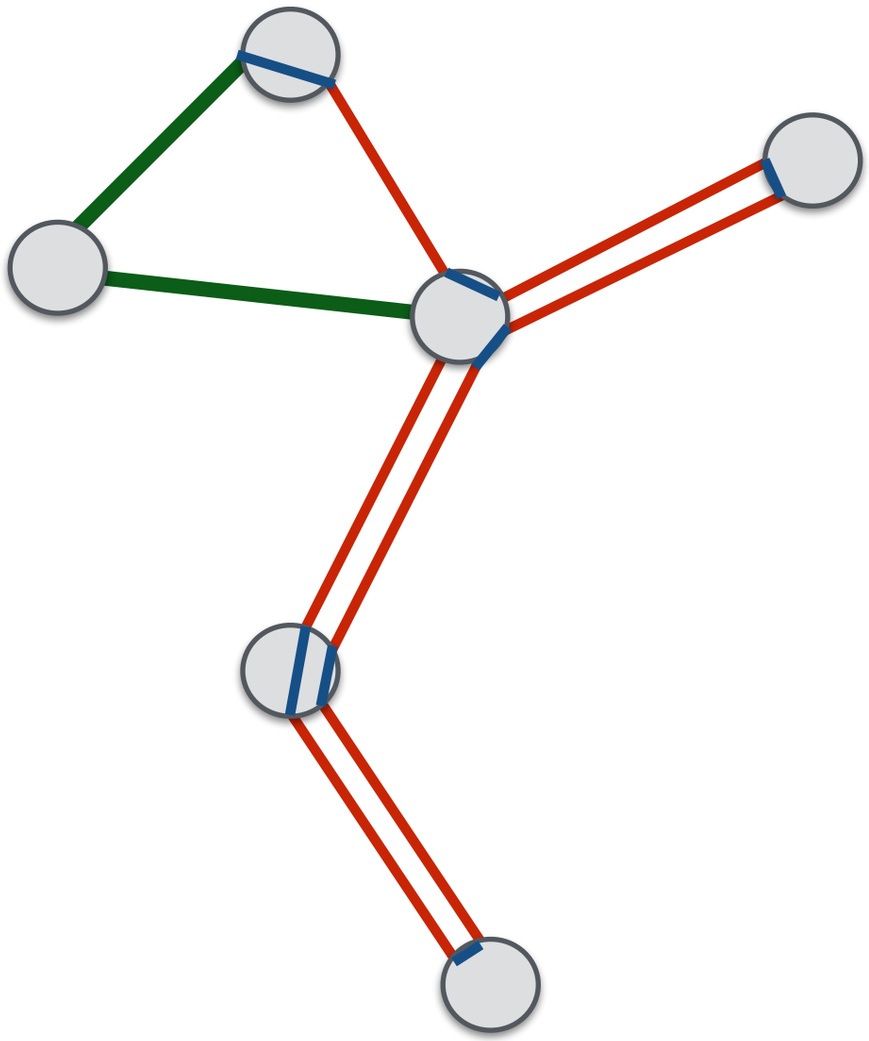
   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird ⇒ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2\text{opt}(K_n, \ell)$

3. bestimme Eulertour **W**

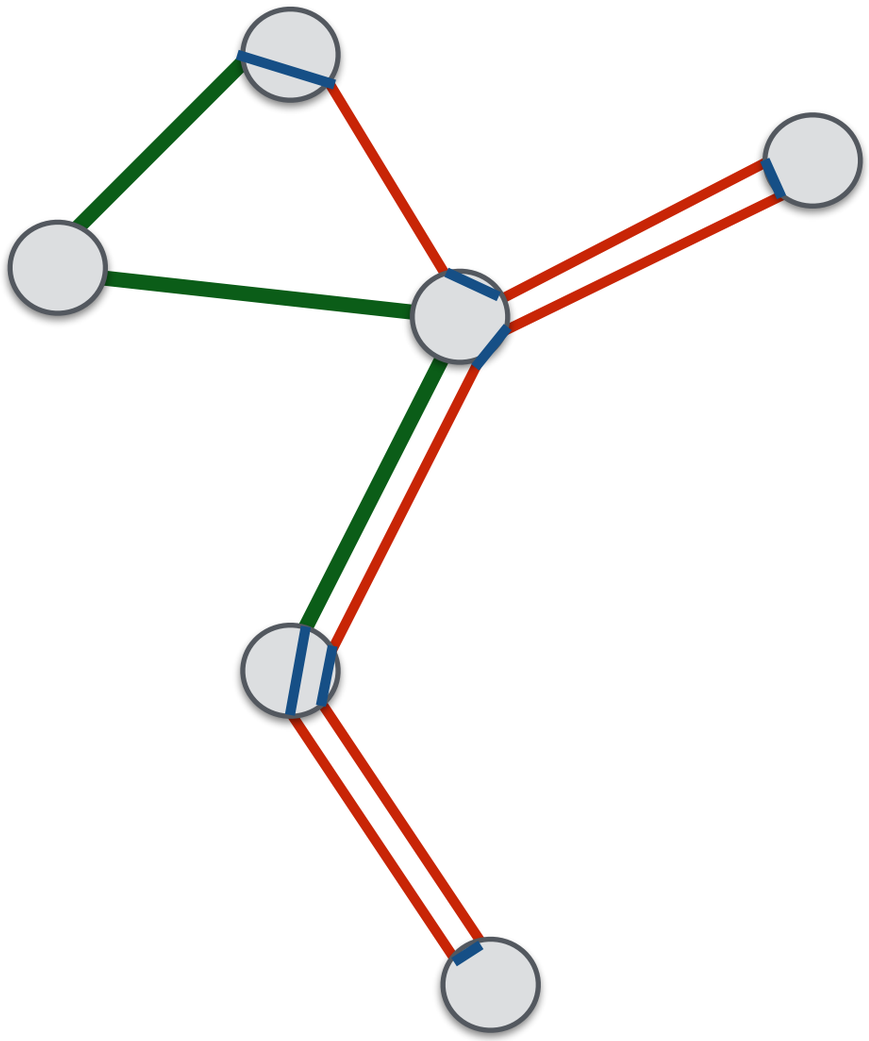
   es gilt: $\ell(W) = 2\ell(T) \leq 2\text{opt}(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

3. bestimme Eulertour **W**

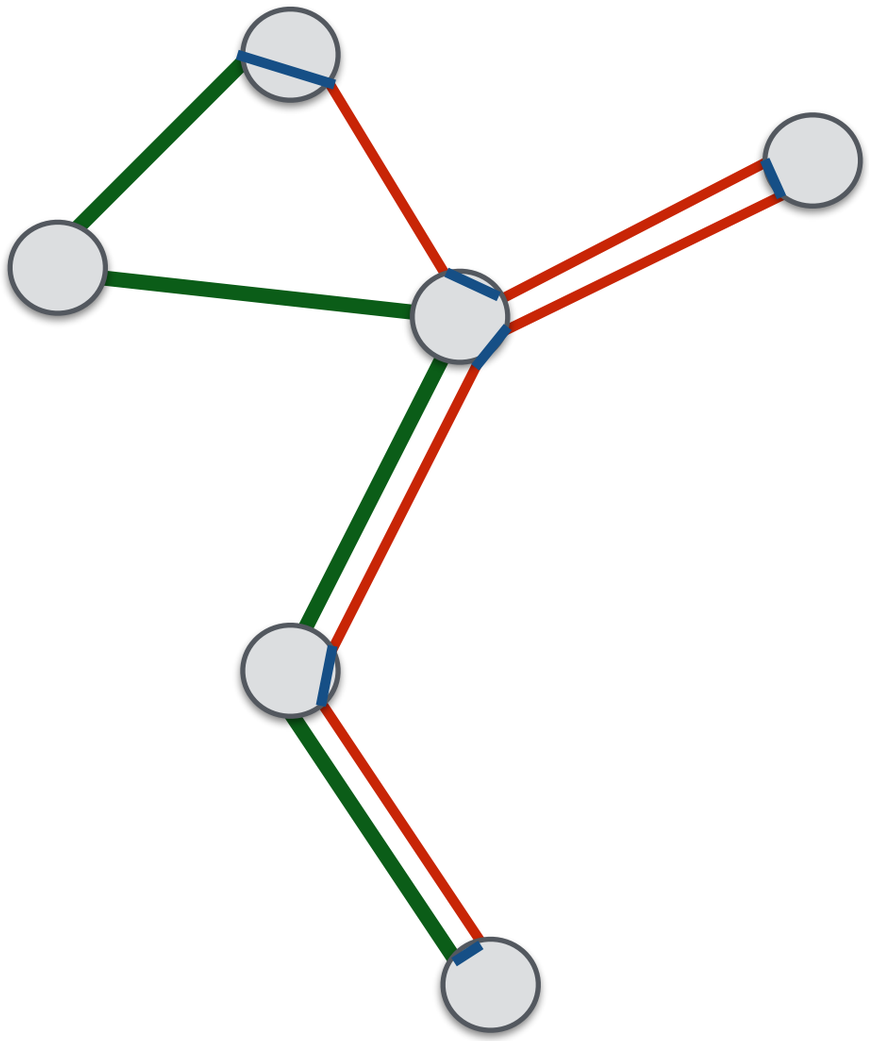
   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird ⇒ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2\text{opt}(K_n, \ell)$

3. bestimme Eulertour **W**

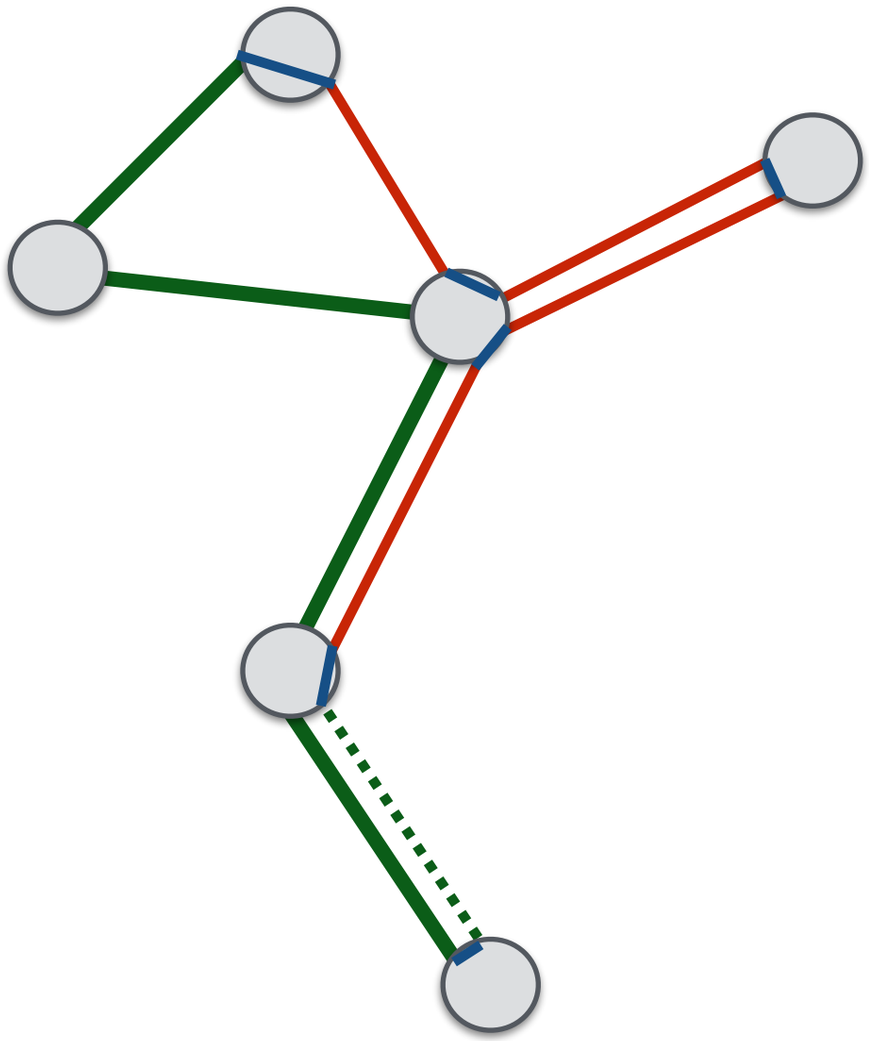
   es gilt: $\ell(W) = 2\ell(T) \leq 2\text{opt}(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

3. bestimme Eulertour **W**

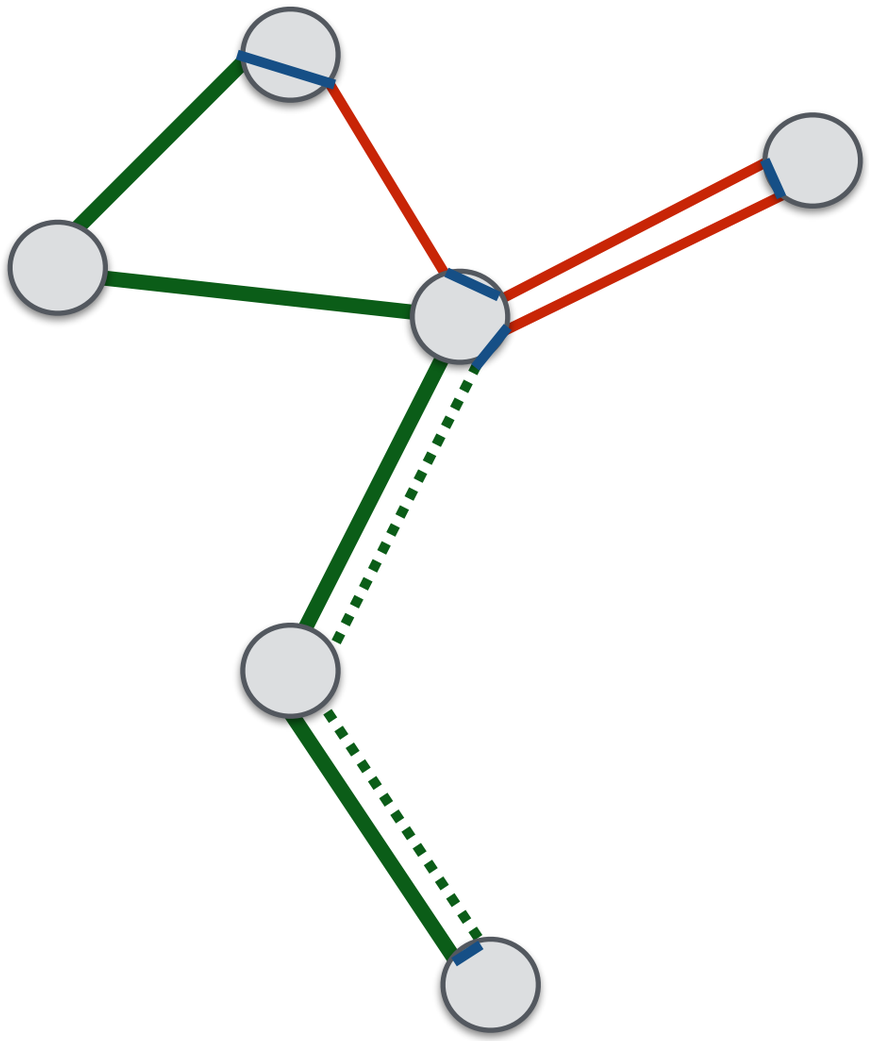
   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird ⇒ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

3. bestimme Eulertour **W**

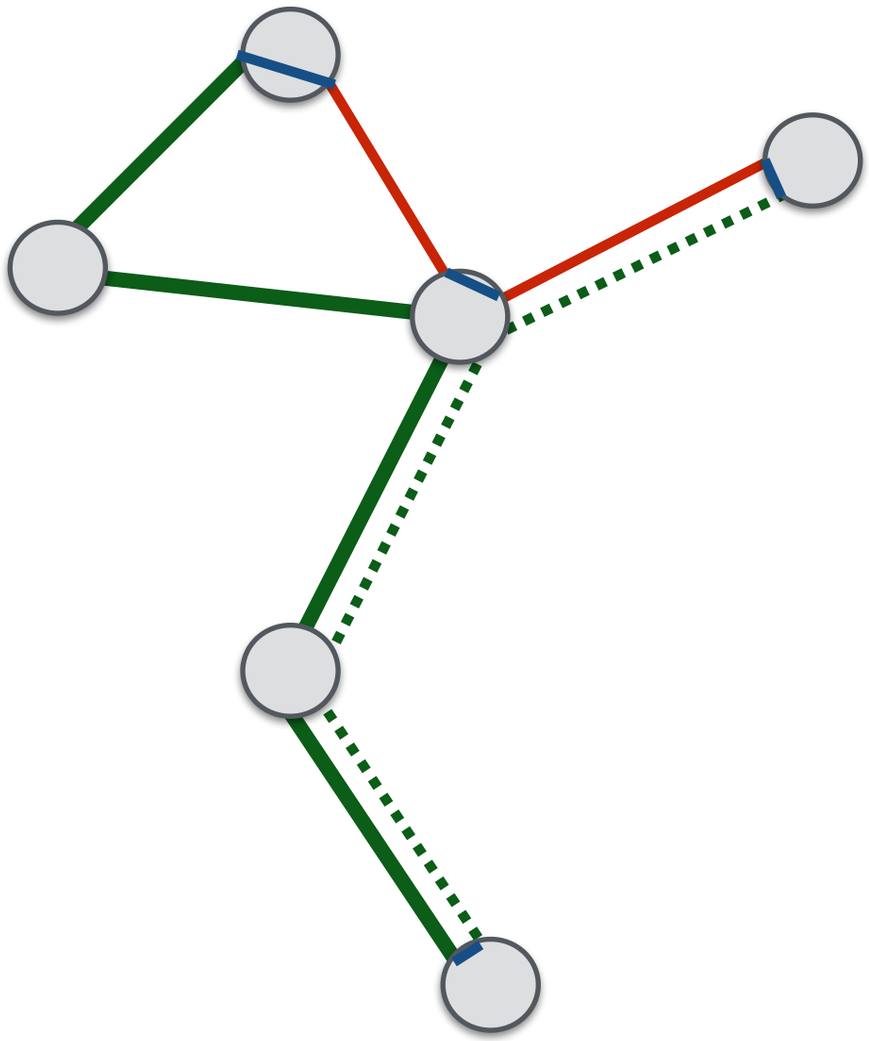
   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird ⇒ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2\text{opt}(K_n, \ell)$

3. bestimme Eulertour **W**

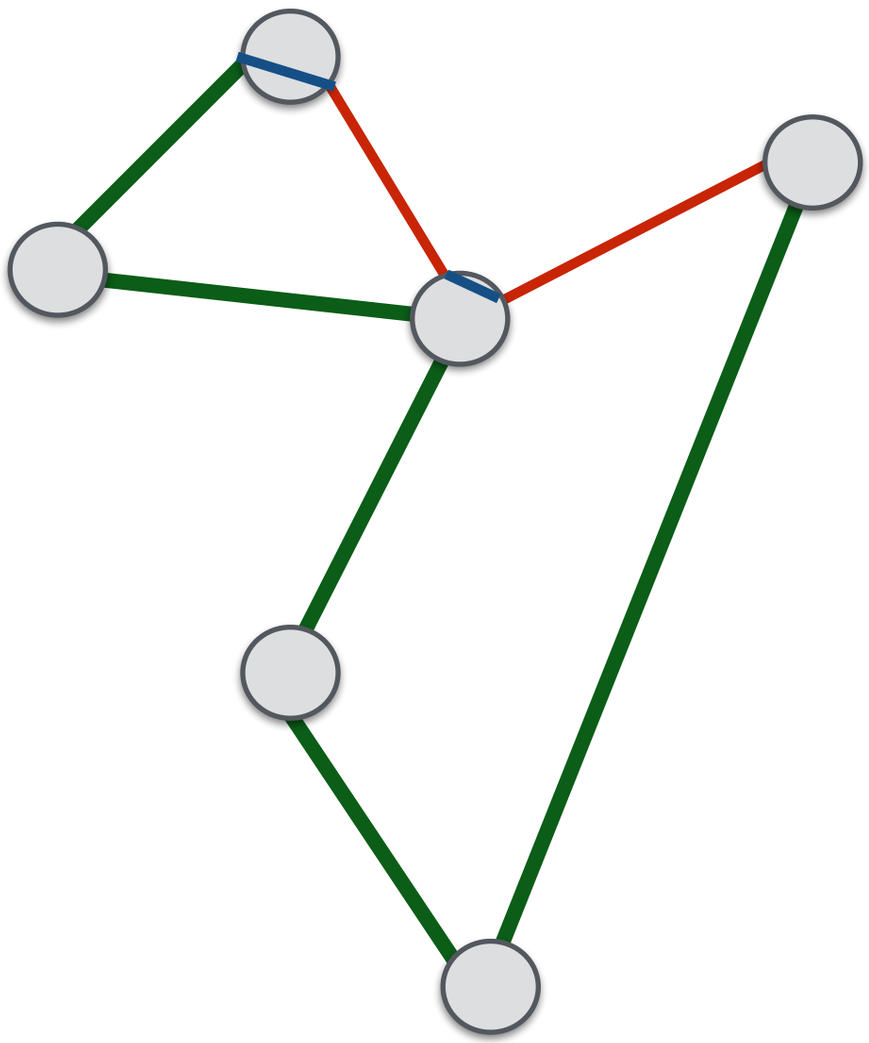
   es gilt: $\ell(W) = 2\ell(T) \leq 2\text{opt}(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird ⇒ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

3. bestimme Eulertour **W**

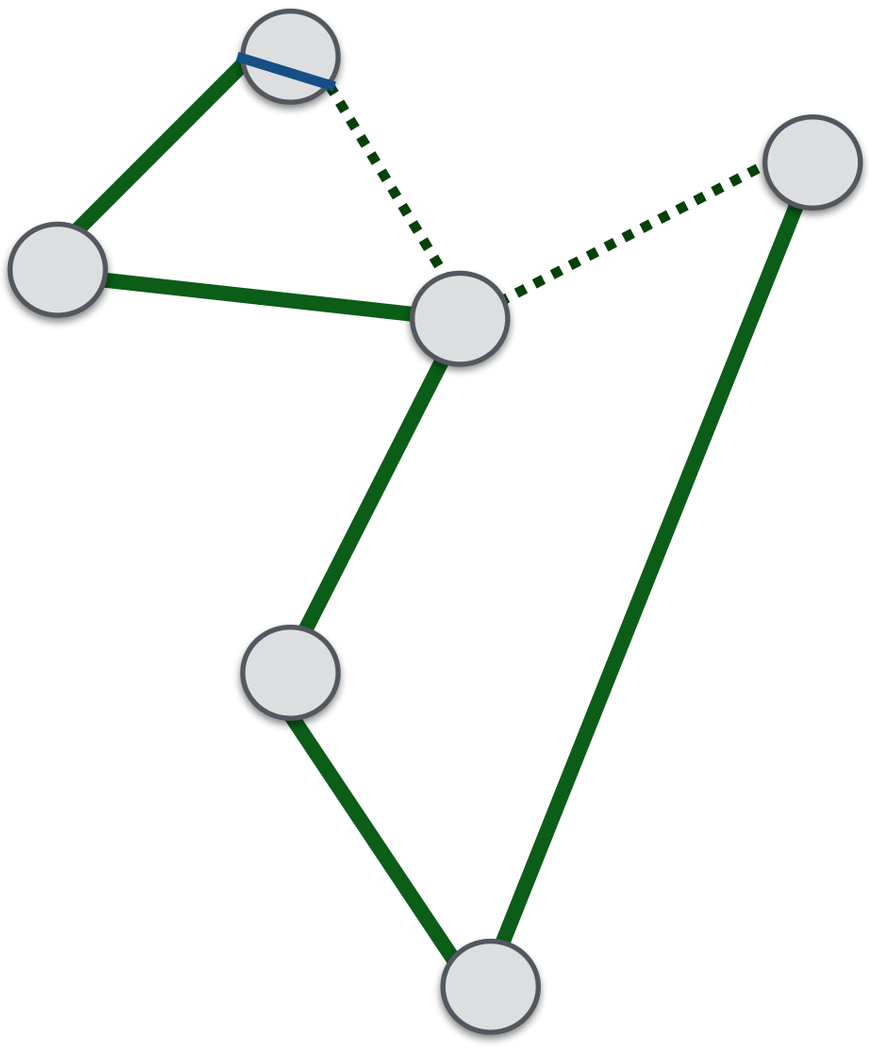
   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird ⇒ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq opt(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2opt(K_n, \ell)$

3. bestimme Eulertour **W**

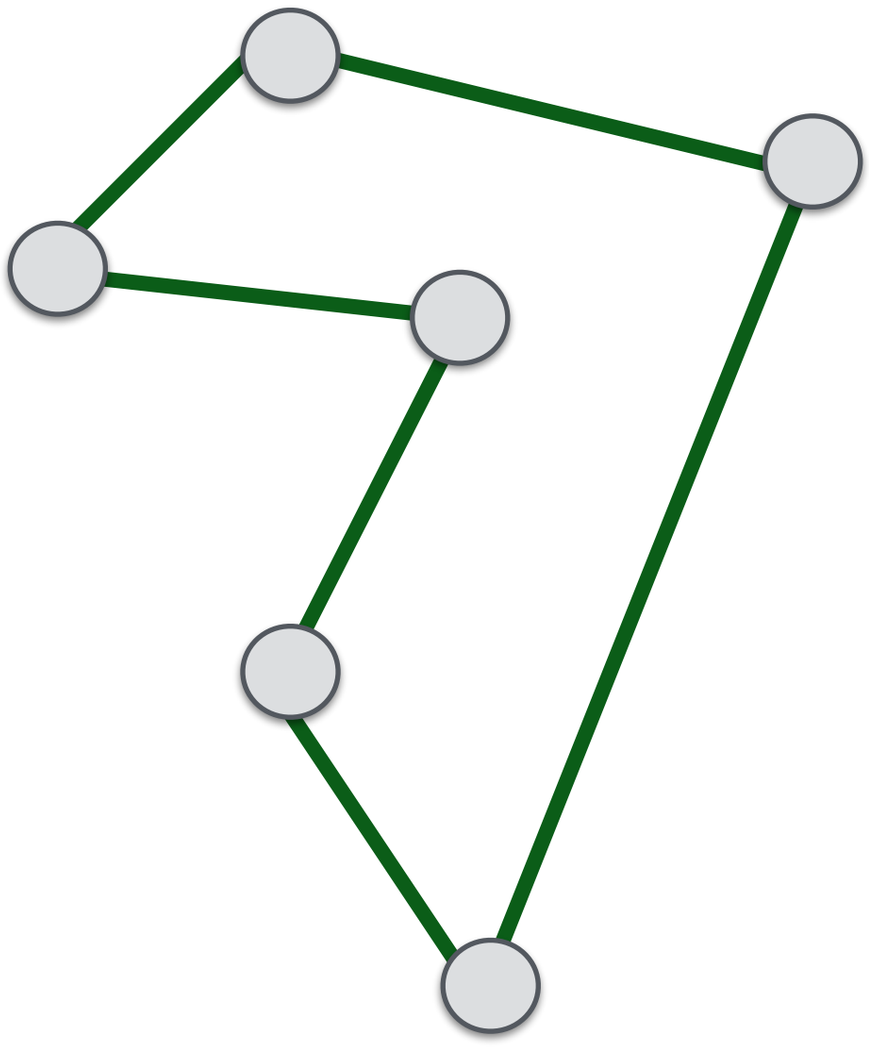
   es gilt: $\ell(W) = 2\ell(T) \leq 2opt(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis **C**

Metrisches Traveling Salesman Problem

Funktion ℓ erfüllt Dreiecksungleichung:

$$\ell(x,z) \leq \ell(x,y) + \ell(y,z) \qquad \forall x,y,z \in [n]$$



1. Bestimme minimalen Spannbaum **T**

   es gilt: $\ell(T) \leq \text{opt}(K_n, \ell)$

2. verdopple alle Kanten von **T**

   es gilt: $2\ell(T) \leq 2\text{opt}(K_n, \ell)$

3. bestimme Eulertour **W**

   es gilt: $\ell(W) = 2\ell(T) \leq 2\text{opt}(K_n, \ell)$

4. durchlaufe **W**, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis **C**

es gilt: $\ell(C) \leq \ell(W) = 2\ell(T) \leq 2\text{opt}(K_n, \ell)$

wegen Dreiecksungleichung