

Problem des Handlungsreisenden

- Gesucht ist eine Rundreise durch eine gegebene Menge von Orten, so dass
 - jeder Ort mit Ausnahme des Startorts nur ein Mal besucht wird
 - die Rundreise möglichst “günstig” ist
- Modellierung als Graph:
 - Orte als Knoten
 - Verbindungen zwischen Orten als gewichtete Kanten
 - Meist: Vollständiger Graph, d. h. alle Knoten sind paarweise verbunden

Ausprobieren:

- TSP-Spiel

Definition

- Ein geschlossener Kantenzug, der alle **Knoten** eines Graphen genau ein Mal enthält, heißt **hamiltonscher** Kreis
- Ein Graph, der einen hamiltonschen Kreis enthält, heißt **hamiltonscher Graph**
- Es ist kein effizienter Algorithmus bekannt, der für jeden Graphen entscheiden kann, ob er hamiltonsch ist oder nicht
- Zur Erinnerung: Ein geschlossener Kantenzug, der alle **Kanten** eines Graphen genau ein Mal enthält, heißt **eulerscher** Kantenzug
- Zur Entscheidung, ob ein Graph eulersch ist oder nicht, reicht die Betrachtung des Grads aller Knoten

- In der theoretischen Informatik gibt es zwei große Problemklassen, P und NP
 - P enthält alle Probleme, für die eine **deterministische Turingmaschine** in polynomieller¹ Zeit eine Lösung findet
 - NP enthält alle Probleme, für die eine **nichtdeterministische Turingmaschine**² in polynomieller Zeit eine Lösung findet
- Es gilt $P \subseteq NP$, aber es ist nach wie vor unklar, ob $P = NP$

¹**Polynomielle Laufzeit** bedeutet, dass die Laufzeit in Abhängigkeit der Problemgröße durch eine ganzrationale Funktion beschrieben werden kann, z. B. $O(n^2)$ oder $O(n^{1000})$

²Eine **nichtdeterministische Turingmaschine** hat mehrere mögliche Übergänge, die sie alle (parallel) abarbeitet

- Das “Problem des Handlungsreisenden” erfordert die Suche nach einem hamiltonschen Kreis mit minimalen Kosten (Summe der Kantengewichte)
- Das Problem gehört (mindestens) der Problemklasse NP an
 - Eine exakte Lösung für große Graphen kann nicht in vertretbarer Zeit gefunden werden
- Ist es wenigstens möglich, gute Näherungslösungen zu finden?

Wiederholung: Greedy-Algorithmen

- Als Greedy-Algorithmen bezeichnet man Algorithmen, die
 - schrittweise eine Lösung suchen
 - dabei immer den momentan besten Folgeschritt wählen
- Ein Greedy-Algorithmus findet nicht notwendigerweise die bestmögliche Lösung
- Prim und Kruskal sind Beispiele für Greedy-Algorithmen
 - beide finden auch die optimale Lösung

- Greedy-Algorithmus für das Problem des Handlungsreisenden:
 1. Wähle einen beliebigen Startknoten
 2. Gehe zum nächstgelegenen unbesuchten Nachbarknoten
 3. Setze das Verfahren fort, bis alle Knoten besucht sind
- Liefert in der Regel **keine** optimale Lösung