

ÜBUNG 4

Arbeiten Sie in 2er Gruppen an folgendem Programmierbeispiel. Die Wahl der Programmiersprache bleibt Ihnen überlassen (C, C++, C#, Java).

Aufgabenstellung

TSP Enumeration und Nearest Neighbour Heuristic

Implementieren Sie die vorgestellten Algorithmen vollständige Enumeration und Nearest Neighbour Heuristic zur Suche einer Rundreise für das TSP. Der Algorithmus (Enumeration oder Nearest Neighbour Heuristic) soll als Option auf der Kommandozeile angegeben werden (-e bzw. -n).

Das Hauptprogramm liest aus einer Datei (Dateiname wird dem Programm als Parameter übergeben) n zweidimensionale Punkte (= Städte) ein. Bauen Sie nun für diese Punkte eine Adjazenzmatrix eines vollständigen, ungerichteten, gewichteten Graphen auf, wobei die Kantengewichte als Entfernung der Punkte in der Ebene definiert sind. Geben Sie die Adjazenzmatrix aus.

Anschließend berechnen Sie auf Basis der Adjazenzmatrix mittels vollständiger Enumeration und mittels Nearest Neighbour Heuristic Lösungen für das TSP. Geben Sie die von Ihren Algorithmen gefundenen Lösungen aus, indem Sie die einzelnen Punkte in der besuchten Reihenfolge angeben sowie die Gesamtlänge der Rundreise (bei der Enumeration genügt die Ausgabe der optimalen Tour). Um die Algorithmen miteinander vergleichen zu können, soll zusätzlich auch die benötigte CPU-Zeit zur Berechnung der Rundreisen ausgegeben werden.

Der Programmaufruf sollte wie folgt aussehen: **tsp -e|-n filename**
Die Ausgabe erfolgt auf die Standardausgabe (stdout).

Hinweise File-Format

- Zeile 1 enthält die Anzahl n der einzulesenden Punkte ($n > 0$).
- Zeilen 2 bis 2+n-1 enthalten jeweils einen 2-dim Punkt

Beispiel:

File:

```
5
1.0 1.0
5.5 0.9
3.0 1.5
0.5 4.3
4.0 3.0
```

Alternativ können auch das Fileformat und Testdaten der TSPLIB <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> verwendet werden.

Protokoll

Erstellen Sie ein Protokoll mit Vergleichsmessungen für Enumeration und Nearest Neighbour Algorithmus für verschiedene Testfiles und vergleichen Sie die gefundene optimale Lösung der Enumeration mit der Näherungslösung des Nearest Neighbour Verfahren (Laufzeit, Tourlänge).

Beschreiben und begründen Sie auch den Aufwand Ihrer Algorithmen mittels O-Notation.

Abgabe

Die Abgabe erfolgt über Moodle. Abzugeben ist eine zip/tgz Datei mit:

- a) Quellcode mit ausführlichen Kommentaren
- b) Ausführbares Programm
- c) Protokoll

Die Abgabe muss beim dritten Code Review präsentiert werden, es können für diese Übung maximal 25 Punkte erreicht werden (20 Punkte für das Programm und 5 Punkte für das Protokoll).

Punkteschlüssel

Implementierung 20 Punkte

- Einlesen und Aufbau der Adjazenzmatrix 5
- Vollständige Enumeration 5
- Nearest Neighbour 5
- Effizienz und Codeverständnis 5

Protokoll 5 Punkte

- Zeitmessungen 2
- Aufwandsabschätzung 3

Hinweis: Unsaubere Implementierung und daraus folgende Crashes in der zentralen Funktionalität führen zu Punkteabzügen!