

## Übung 5 - Flüge buchen

- nachdem letzte Übung hart, diese etwas leichter
- Übung beschäftigt sich mit einer Erweiterung für Dijkstras Algo
- intuition routing
- heuristiken (wichtig:  $O(1)$ )
- manhattan, euclidean distance, geodesic
- gibt es alles schon (auch in c++), bitte versuchen sie es dennoch selbst

### Aufgabe 1: Data

Dateien die in dieser Aufgabe editiert werden sollen: - SG.h

- nutzen sie die VL Graph implementierungen
- laden sie graph mit vertices von airports
  - vertices müssen attribute unterstützen
- laden sie edges aus routes
- TODO: finde werte für dauer der strecken?

### Aufgabe 2: A\* Algorithmus

Dateien die in dieser Aufgabe editiert werden sollen: - a\_star.cpp - distances.cpp - routing\_cli.cpp

- auf basis von Dijkstra aus VL (nochmal mitgeben)
- Programmieren sodass heuristik mit austauschbarer Funktion
- euclidean distance
- vergleich dijkstra mit A\* für unseren Anwendungsfall

### Aufgabe 3: Andere Distanzen

Dateien die in dieser Aufgabe editiert werden sollen: - distances.cpp - A3.md

In dieser Aufgabe wollen wir unsere Implementierung von A\* dadurch ausbauen, dass wir andere Heuristiken hinzunehmen und diese dann untereinander Vergleichen. Das euklidische Distanzmaß ist eine einfache Heuristik für unseren Anwendungsfall, hat aber den Nachteil das sie annimmt das alle Flughäfen auf einer Ebene liegen. Wie wir wissen ist die Erde nicht flach [citation], sondern ein Ellipsoid (eine gedehnte Kugel) und es ist auch nicht möglich die Oberfläche einer Kugel auf die flache Ebene ohne Verzerrung zu projizieren [citation]. Ein vermeintlich besseres Distanzmaß wäre die sogenannte Geodesische Distanz [citation], welche die Distanz als Pfad über die Oberfläche eines Ellipsoids berechnet.

- geodesic distance
- manhattan distance
- random :smile: ?
- out-degree (abuse small world phenomenon)
- comparison btw. runtime