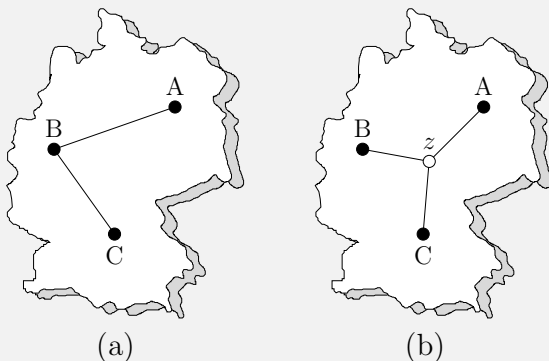


## informatiCup 2008 • Aufgabe 1

### CargoConcept

#### Einführung

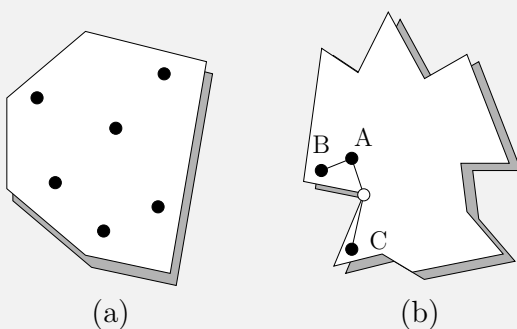
Volle Straßen und hohe Energiepreise machen der Transportindustrie zunehmend das Leben schwer. Eine Alternative zu den heute genutzten Transportwegen besteht darin, unterirdisch ein zusätzliches Transportsystem für Güter aufzubauen. Ein solches System, wie etwa das schon 2006 ausgezeichnete Projekt **CargoCap**, funktioniert ähnlich wie ein Rohrpostsystem. Waren werden an den nächsten *Startstandort* auf Europaletten angeliefert, dann in den Transportröhren möglicherweise über *Zwischenstandorte* zum *Zielstandort* befördert. Die Start- bzw. Zielstandorte sind also vergleichbar mit Verladebahnhöfen im Schienenverkehr und Zwischenstandorte gleichen Weichen.



#### Beispiel 1:

Gegeben sind drei Standorte A, B und C. In Alternative (a) werden die Standorte A, B und B, C durch Röhren verbunden und es entsteht ein gültiges System. Das ist auch bei (b) der Fall. Dort wird aber durch einen Zwischenstandort  $z$  die Gesamtlänge des Röhrensystems im Vergleich zu (a) verkürzt.

Damit Waren von einem beliebigen Startstandort zu jedem Zielstandort im System befördert werden können, müssen alle Standorte *direkt* oder *indirekt* durch *Röhren* verbunden sein. Ist dies der Fall, nennen wir das System *gültig*. Da das Verlegen dieser Transportröhren aber sehr kostenintensiv ist, versucht man möglichst wenige Meter an Röhren zu verbauen. Wie man an Beispiel 1 sieht ist das Einführen von Zwischenstandorten ein guter Ansatz, um die Gesamtlänge des Röhrensystems zu minimieren.



#### Beispiel 2:

Die Karte (a) ist ein konvexes Polygon. D.h. jedes Punktepaar innerhalb des Polygons läßt sich durch eine gerade Linie innerhalb des Polygons verbinden. Karte (b) stellt ein konkaves Polygon dar. D.h. es gibt Standorte die nicht direkt durch eine Röhre verbunden werden können, wie etwa Standort A bzw. B mit dem Standort C.

Es gibt bisher noch nicht betrachtete Einschränkungen des System. Alle Röhren sollen gerade sein und das System soll innerhalb einer Karte verlaufen. Eine Karte ist in unse-

rem Fall ein *simples Polygon*<sup>1</sup>, das entweder konvex oder konkav sein kann. Die meisten realen Karten sind konkav, etwa die Karte der Bundesrepublik Deutschland in Beispiel 1. Es gibt aber auch Beispiele von konvexen Karten, wie die der US-Staaten Colorado oder Wyoming.

## Aufgabenstellung

Entwickeln Sie für die erste Runde ein Programm, das zu einer gegebenen Menge an Standorten und einer Karte ein gültiges System berechnet, dessen Gesamtlänge möglichst klein ist. Beachten Sie folgende Hinweise:

- (a) Auf der InformatiCup-Seite finden Sie eine Beschreibung des Formats, das Ihr Programm für Testeingaben verarbeiten soll.
- (b) Zwischenstandorte können einen beliebigen Verzweigungsgrad haben und können mit Start- und Zielstandorten zusammenfallen.
- (c) Beschränken Sie sich in dieser Runde auf konvexe Karten mit bis zu 50 Standorten.
- (d) Geben Sie die Approximationsgüte (bzgl. einer effizient bestimmbaren oberen Schranke einer optimalen Lösung ) für Ihren Lösungsalgorithmus an.
- (e) Erstellen Sie zehn Testfälle, von denen Sie annehmen, dass Ihr Algorithmus im Vergleich zu denen anderer Teams gute Ergebnisse liefert.
- (f) Dokumentieren Sie Ihre Lösungsideen und die algorithmische Umsetzung.

In der zweiten Runde erweitern wir die algorithmische Problemstellung auf konkave Karten. Zudem soll eine graphische Oberfläche erstellt werden, die folgende Funktionalitäten unterstützen soll:

- (a) Import von realen, frei zugänglichen Kartendaten (etwa aus dem Projekt [OpenStreetMap](#)).
- (b) Festlegung der Standorte auf einer graphischen Ausgabe der Karte.
- (c) Darstellung der Lösungen im Verlauf des Algorithmus auf der graphischen Karte.

Erstellen Sie eine Bedienungs- und Installationsanleitung. Schicken Sie uns Ihre Implementierung, Anleitungen sowie einen Bericht über die von Ihnen gemachten Entwurfsentscheidungen in der Software-Entwicklung.

## Hinweise

- Informationen über CargoCap finden Sie unter <http://www.cargocap.de>.

---

<sup>1</sup>Die Linien eines simplen Polygons dürfen sich nicht überschneiden. Dadurch erhalten wir ein wohl definiertes Inneres des Polygons.

- Informationen zum „Projekt OpenStreetMap“ finden Sie unter <http://www.openstreetmap.de>.