

GeoMagic Testaufgabe

Filip Mazug

Kommentar

Im Folgenden möchte ich einen kurzen Überblick über den Algorithmus und das Konzept geben, damit das Einlesen leichter fällt. Aber zunächst noch ein kleiner Kommentar zur Aufgabenstellung. Ich hatte den Eindruck, der Satz: "Treffen sich mehr als zwei Linien in einem Punkt, dann wird an dieser Stelle nicht zusammengefügt.", kann auf verschiedene Weisen interpretiert werden.

Option 1: "In diesem Punkt wird nicht zusammengefügt", bedeutet nichts wird dort zusammengefügt. Somit gehen von diesem Punkt keine Linien aus.

Option 2: Mehr als zwei Linien können in einem Punkt zusammenlaufen, aber in diesem Punkt werden dann die Linien nicht verbunden. D.h. alle Linien, die in diesem Punkt zusammenlaufen, enden dort. Oder fangen dort an, je nachdem wie man es betrachtet. Wenn sie dort anfangen, können sie von dort aus in andere Richtungen weiterlaufen. Diese Option bedeutet, dass in einem Punkt mehrere Linien quasi übereinander liegen können, die dort ihren Start oder Endpunkt haben, aber eben untereinander nicht verbunden sein können.

Den Lösungsweg zu Option 1 hatte ich bereits ausgearbeitet und abgegeben zusammen mit der Überlegung, warum ich mich zuerst für Option 1 entschieden habe. Diese Abgabe befasst sich mit dem Lösungsweg zu Option 2. Und da Option 2 doch die besseren Ergebnisse liefert, weil wie in der Aufgabe gefordert, mehr Linienzüge erstellt werden können, schien es mir am Ende doch richtiger und sicherer, auch Option 2 mit abzugeben.

Die Code-Dokumentation ist recht ausführlich, aber für ein einfaches Einlesen möchte ich gerne nochmal auf das wichtigste zum Konzept und der Implementierung eingehen.

Zum Konzept:

- Eine einfache Datenstruktur Point speichert ein (x,y) Koordinatenpaar für besseres Handling
- Es gibt 3 mögliche Klassen von Points, welche durch die Aufgabenstellung vorgegeben sind.
 - SinglePoints (SP), die genau einmal vorkommen
 - Connectoren (CN), kommen genau zweimal vor und verbinden zwei Linien
 - Terminatoren (TER), kommen 3 oder mehr als dreimal vor und verbinden 3 oder mehr Linien
- Ein Linienzug wird erstellt, indem ein Punkt gewählt wird (SP, CN, TER) und von diesem Punkt aus ein nächster Punkt angehängt wird, sodass diese zwei Punkte eine Linie bilden, wie sie auch in der input.txt Datei vorkommt.
- Je nachdem welche Punkte zum Verlängern der Linien gewählt werden, wächst der Linienzug. Ein SP (falls vorhanden) bildet immer den Anfang einer Linie oder eines Linienzuges oder das Ende, da von diesem Punkt nur eine Linie ausgeht und er mit höchstens einem weiteren Punkt verbunden ist.
- In einem TER können mehrere Linien zusammenlaufen, aber diese Linien werden dann in dem Punkt nicht verbunden. (Nach Definition von Option 2.) Daher kann ein TER ein Anfangs oder Endpunkt einer Linie oder eines Linienzuges sein, aber nicht in der Mitte eines Linienzuges vorkommen, da ein TER keine Linien verbindet.
- Ein CN befindet sich somit immer in der Mitte eines Linienzuges da er genau zwei Nachbarpunkte hat.
- Jeder Punkt hat ein Attribut type, in welchem codiert ist, wie oft der Punkt in der input.txt Datei vorkommt und somit, wie oft er insgesamt verwendet werden darf. Über den type wird außerdem bestimmt zu welcher Klasse ein Punkt gehört (SP, CN, TER). Durch die Kontrolle und das Aktualisieren der types beim Bilden von Linienzügen wird sichergestellt, dass alles nach den durch das Konzept vorgegeben Rahmenbedingungen abläuft und nur per Definition erlaubte Linienzüge gebildet werden.
- Neben der input.txt Datei befinden sich im Projektordner noch zwei weitere Input Dateien, um den Algorithmus auf besondere Bedingungen zu testen und somit sicherzustellen, dass die Funktion nicht auf die Aufgabe hardgecoded ist. Die eine Datei bildet eine zirkuläre Struktur ab, die nur aus CNs besteht. Die andere Datei enthält einen Terminator in dem nicht nur 3, sondern auch 5 Linien zusammenlaufen.

Vereinfachter Ablauf zur Orientierung

- SPandCNandTER = Liste aller Punkte (mit ihren types) in input.txt
- starterList = Liste der Linien aus input.txt, wobei die Linien durch Punkte definiert sind
- Linienzüge = Liste aller bisher gefundenen Linienzüge
- Linienzug = aktueller Linienzug, der bearbeitet wird. Oder temporärer Container für diesen Linienzug, der im rekursiven Aufruf als Parameter übergeben wird. Ist der Container als Inputparameter für den rekursiven Aufruf leer, wird ein neuer Linienzug begonnen. Ist der Container nicht leer, wird auf diesem Linienzug weitergearbeitet.

Rekursiver Algorithmus

1. Basisfall: Wenn keine Punkte mehr übrig sind, um einen Linienzug zu bilden, beende die Rekursion und gib die aktualisierte Liste der Linienzüge zurück.
2. Nehme Punkt aus SPandCNandTER
3. Wenn der aktuelle Punkt ein SP oder TER ist:
 - Suche nach einem zweiten Punkt (SP oder CN oder TER), um eine Linie zu bilden.
 - Wenn ein zweiter Punkt gefunden wird (weil diese Kombination in starterList existiert), bilde Linie zwischen den beiden Punkten.
 - Wenn eine Linie erstellt werden kann:
 - Falls die Kombination (SP, SP oder SP, TER oder TER, SP oder TER, TER) war: Füge die Linie zu den Linienzügen hinzu
 - Durch entsprechende Übergabe der Parameter erkennt die Funktion im rekursiven Aufruf, dass der Linienzug zu Ende war und ein neuer beginnt.
 - Falls der zweite Punkt ein CN war, muss der Linienzug weitergebaut werden: Füge die Linie zum Linienzug (temporärer Speicher) hinzu
 - Funktion erkennt im rekursiven Aufruf, dass der Linienzug weitergebaut werden muss.
 - Aktualisiere Point types (in SPandCNandTER und starterList)
 - Rufe die Funktion rekursiv auf, um den nächsten Linienzug zu erstellen, oder den aktuellen zu erweitern.
4. Wenn der aktuelle Punkt ein CN ist:
 - Suche nach einem weiteren Punkt (SP, CN, TER), um den Linienzug fortzusetzen.
 - Solange ein weiterer CN gefunden wird, setze den aktuellen Punkt auf den gefundenen CN und such weiter.
 - Funktion erkennt im rekursiven Aufruf, dass der Linienzug weitergebaut werden muss.
 - (Sonderfall: es gibt nur noch CNs, dann ist es ein Kreis
-> füge diesen zu Linienzügen hinzu)
 - Wenn kein weiterer CN gefunden wird, such nach einem abschließenden SP oder TER, um den Linienzug zu beenden.
 - Wenn ein abschließender SP oder TER gefunden wird, füg den Linienzug zu den Linienzügen hinzu.
 - Aktualisiere Point types (in SPandCNandTER und starterList)
 - Rufe die Funktion rekursiv auf, um den nächsten Linienzug zu erstellen, oder den aktuellen zu erweitern.
5. Rekursion: Teilmenge von verfügbaren Punkten (definiert durch die types) in SPandCNandTER wird weitergegeben, sowie eine Teilmenge der noch nicht verwendeten Linien (starterList), während die Menge der Linienzüge wächst.
6. Rückgabe der Linienzüge: Am Ende der Rekursion wird die Liste der Linienzüge zurückgegeben, nachdem alle Punkte und Linien ordnungsgemäß verwendet wurden.