

Übungsblatt 3

Abgabe bis 6. Dezember 2021

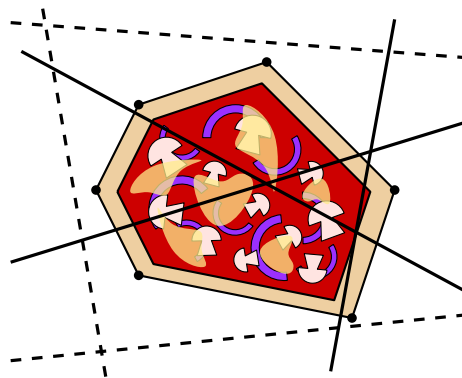
Aufgabe 1: Pizza-Guillotine Teil 1

5 Punkte

Viele Studenten beschwerten sich darüber, dass die örtliche Pizzeria ihre Backwaren ungeschnitten ausliefert. Um die Kundenzufriedenheit zu verbessern, soll das Pizzaschneiden automatisiert werden. Ein erster Prototyp ist eine Pizza-Guillotine. Sie enthält eine unendlich lange Klinge, die prinzipiell frei in der Ebene verschoben und rotiert werden kann und dann auf die Pizza herabfällt. Da der Prototyp noch im Anfangsstadium ist, kann die Ausrichtung der Klinge jedoch nur mit dem Befehl *Reset* geändert werden, welcher die Klinge in eine zufällig gewählte Ausrichtung überführt.

Um trotzdem mit dem Prototypen arbeiten zu können, hat man sich auf folgendes Vorgehen geeinigt. Es wird getestet, ob die aktuelle Ausrichtung der Klinge ungünstig ist oder nicht. Falls ja, wird ein Reset ausgeführt und der Vorgang wiederholt. Andernfalls wird geschnitten. Da dieses Vorgehen extrem ineffizient ist, muss der (womöglich häufig ausgeführte) Test sehr schnell sein!

Unglücklicherweise kann es passieren, dass die Klinge in ihrer aktuellen Ausrichtung sehr ungünstig oder sogar neben der Pizza landen würde. In diesem Fall muss mittels Reset eine neue Ausrichtung probiert werden. Eine Pizza ist dabei durch ein konvexes Polygon mit n Punkten gegeben. Gebt einen Algorithmus an, der nach $\mathcal{O}(n)$ Vorberechnungszeit jedes Mal in $\mathcal{O}(\log n)$ Zeit bestimmen kann, an welchen Punkten die Klinge in ihrer aktuellen Ausrichtung auf den Rand der Pizza fallen würde.



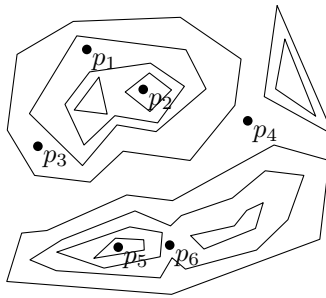
Aufgabe 2: Skihütten

5 Punkte

Ihr seid dabei den nächsten Skiurlaub zu planen. Das wichtigste Kriterium für die Wahl eines Skigebiets ist (für verschiedene x) die Anzahl Skihütten die auf x Metern Höhe liegen. Um einfacher bewerten zu können wie interessant ein Skigebiet ist, sucht ihr deshalb einen effizienten Algorithmus der derartige Informationen aus den Karten der Skigebiete extrahieren kann.

Die Karten bestehen aus einer Menge von Polygonen (welche Höhenlinien repräsentieren) und einer Menge von Punkten (den Skihütten), wobei sich die Kanten der Polygone nicht schneiden, aber trotzdem Polygone in anderen Polygonen enthalten sein können. Der Algorithmus soll für ein Skigebiet bestimmen wie viele der Skihütten in genau x Polygonen enthalten sind.

Beispiel:



Hier befindet sich eine Hütte auf Höhe 0 (diese ist p_4), eine Hütte auf Höhe 1, zwei Hütten auf Höhe 2, sowie zwei Hütten auf Höhe 4.

Aufgabe 3: 2D-LP Vervollständigen

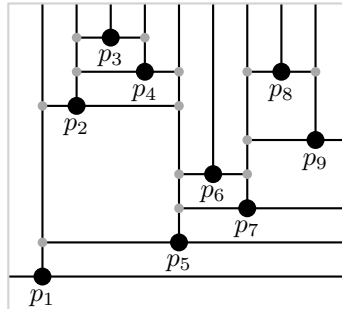
5 Punkte

Gebt einen Algorithmus an, der als Eingabe ein 2-dimensionales lineares Programm (LP) mit n Nebenbedingungen erhält und in $\mathcal{O}(n)$ Zeit bestimmt, ob es unbeschränkt ist. Wie kann euer Algorithmus verwendet werden, um den Algorithmus zur Lösung eines 2D-LP aus der Vorlesung zu vervollständigen?

Aufgabe 4: Halbe 2D-Bereichsanfragen

5 Punkte

Gegeben seien die nach x -Koordinaten sortierten Punkte $p_1, \dots, p_n \in \mathbb{R}^2$. Eine Datenstruktur, mit der halbe Bereichsanfragen schnell beantwortet werden können, unterteilt die Ebene in Zellen. Dabei wird von jedem Punkt p_i ein Strahl vertikal nach oben geschossen und anschließend in beiden Richtungen eine horizontale Kante von p_i zum nächsten Strahl eingefügt, dessen Startpunkt eine kleinere y -Koordinate hat, als p_i selbst.



Gebt einen Algorithmus an, der in $\mathcal{O}(n)$ Zeit den geometrischen Graphen erstellt, der diese Datenstruktur repräsentiert. Beweist, dass euer Algorithmus die vorgegebene Laufzeitschranke nicht überschreitet.