Übungsblatt 11 - Konvexe Hüllen im $\mathbb{R}^3$

1 Worst Case Laufzeit

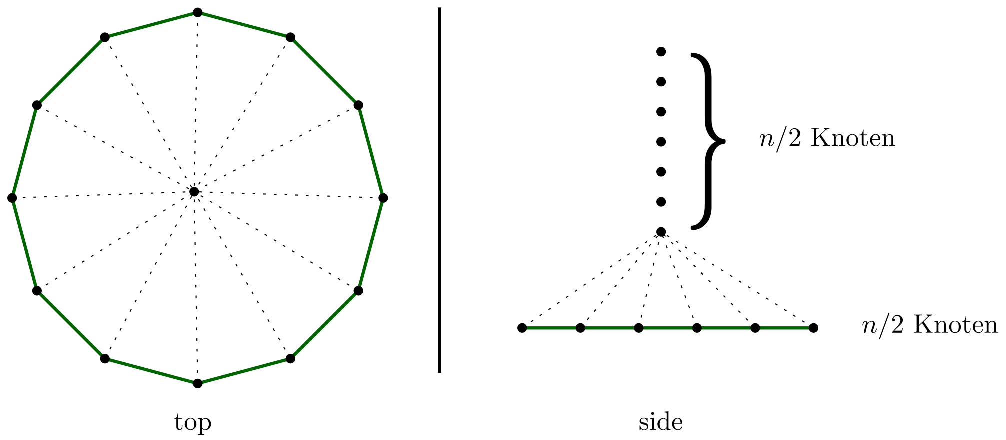
In der Vorlesung wurde der Algorithmus ConvexHull zur Berechnung der konvexen Hülle von einer Punktmenge im $\mathbb{R}^3$ vorgestellt.

a) Zeige, dass die worst-case Laufzeit von ConvexHull in $\mathcal{O}(n^3)$ liegt.

b) Zeige, dass es Punktmengen gibt für die ConvexHull tatsächlich eine Laufzeit von $\Theta(n^3)$ benötigt, wenn eine "schlechte" Permutation erzeugt wird.

Lösung:
Zu a): Drei Schleifen ineinander geschachtelt. Jede Schleife wird $\mathcal{O}(n)$ mal ausgeführt. Der Aufwand für die innerste Schleife ist in $\mathcal{O}(1)$. Insgesamt in $\mathcal{O}(n^3)$.

Zu b): Konstruiere reguläres $n/2$-gon und positioniere geschickt die restlichen $n/2$ Knoten (s. Abbildung). Die 'zufällige' Permutation sorgt zunächst dafür das die konvexe Hülle des $n/2$-gons berechnet wurd und danach $n/2$ mal $\mathcal{O}(n)$ viele neue Facetten hinzugefügt und gelöscht werden.



2 Alternativer Algorithmus

Die konvexe Hülle einer Menge P von n Punkten im 3-dimensionalen Raum kann auch berechnet werden, indem man eine Ebene um bereits bekannte Kanten der konvexen Hülle "rotieren" lässt und auf diese Weise neue Seitenflächen entdeckt. Gebe eine detaillierte Beschreibung dieses Verfahrens an. Achte darauf, dass dein Algorithmus zur Berechnung der konvexen Hülle insgesamt eine Laufzeit von $O(n^2)$ hat und beweise diese Schranke.

Lösung:
Der Algorithmus ist im Grunde der 'Gift-Wrapping'-Algorithmus den wir schon aus dem 2-dimensionalen kennen. Bestimme zwei Punkte die auf jeden Fall Teil der konvexen Hülle sein müssen (ähnlich wie im 2-dimensionalen Fall). Berechne dann für jeden Knoten aus P die Ebene die durch ihn und die schon bekannte Kante definiert wird. Wähle dann die Ebene die, ähnlich dem 2-dimensionalen Fall, den Winkel mit den bereits bestehenden Facetten maximiert [Für den initialen Fall muss man eine Pseudo-Facette einfügen; s. dazu Folien zur ersten Übung]. Der Aufwand liegt in $\mathcal{O}(f \cdot n)$ wobei f die Anzahl der Facetten der konvexen Hülle ist.

3 Zufallszahlen

Bei den in der Vorlesung vorgestellten randomisierten Algorithmen wurde immer ein Zufallszahlengenerator für die Erzeugung einer zufälligen Permutation benötigt. Wir haben biser immer angenommen, dass der Zeitaufwand dafür in $\mathcal{O}(1)$ liegt. Angenommen, der einzige Zufallszahlen-Generator auf den wir zugreifen können erzeugt zufällig nur ein einzelnes Bit (0 oder 1) in konstanter Zeit.

a) Beschreibe ein Verfahren um mit einem solchen Zufallszahlen-Generator eine zufällige Permutation von n Zahlen zu bestimmen.

b) Was is die Laufzeit von dem in a) beschriebenem Verfahren?

Lösung:
Zu a): s. Übungsfolien (Seite 15/38 bzw. 9-4)

Zu b): Ein Durchlauf des Algorithmus $\textsc{RandomInt}(k)$ benötigt (abgesehen von der möglichen Rekursion) $\mathcal{O}(\log k)$ Zeit. Die Wahrscheinlichkeit, dass $\textsc{RandomInt}(k)$ in einem Durchlauf erneut aufgerufen wird ist höchsten $1/2$. Damit ergibt sich für den Erwartungswert der Laufzeit $\sum_i 1/2^i \log k = \mathcal{O}(\log k)$.