Algorithmen und
Berechnungskomplexität I
Wintersemester 2022/23

UNIVERSITÄT BONN

Prof. Dr. Anne Driemel
Frederik Brüning, Jan Eube
Institut für Informatik

Abgabe: 16.01.2023 bis 10.00 Uhr

Übungsblatt 12

Aufgabe 12.1: Sortieren durch Vertauschen (6 Punkte)

Betrachten Sie den folgenden Sortieralgorithmus **GreedySort**.

GreedySort(int[ ] a)

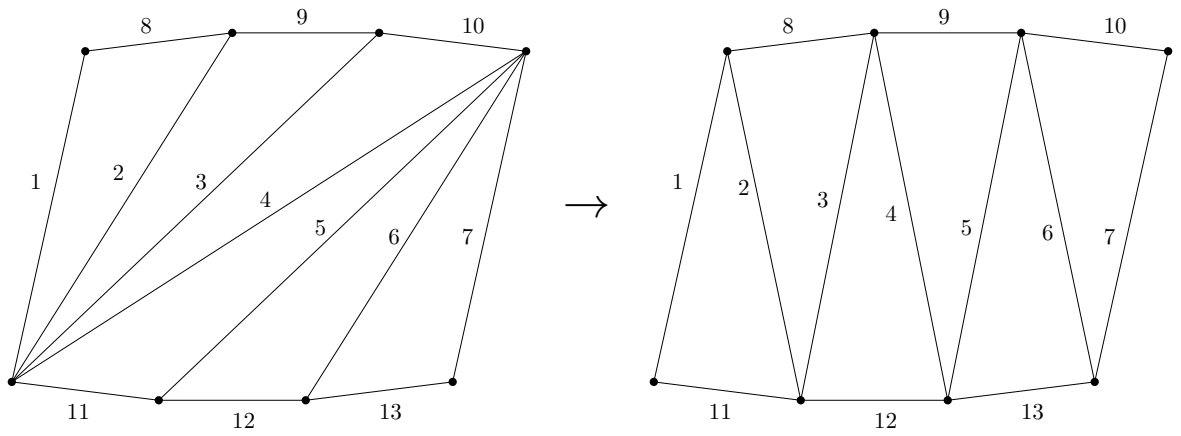
```
1 while (∃ i, j mit i < j, sodass a[i] > a[j]){
2       int x = a[j];
3       a[j] = a[i];
4       a[i] = x;
5 }
6 return a
```

Zeigen Sie, dass **GreedySort** nach weniger als n^2 Vertauschungen terminiert, wobei n die Länge des Feldes a ist. Zeigen Sie dies unabhängig davon, nach welchem Kriterium i und j in Zeile 1 des Algorithmus gewählt werden.

Hinweis: Untersuchen Sie, wie sich die Konfliktfunktion $\Phi(a) = |\{(i,j) \mid i < j, a[i] > a[j]\}|$ in jeder Iteration des Algorithmus verändert.

Aufgabe 12.2: Delaunay-Triangulation (4+4=8 Punkte)

(a) Wenden Sie den GreedyFlips-Algorithmus von Fortune auf das folgende Beispiel an, um die gegeben Triangulation in die daneben gegebene Delaunay-Triangulation zu überführen. Geben Sie hierzu in Reihenfolge der durchgeführten Delaunay-Flips die Nummern der durch die Flips ersetzen Kanten an. Gehen Sie dabei davon aus, dass Kanten mit kleinerer Nummer zuerst geflipt werden, falls mehrere Kanten gleichzeitig einen Delaunay-Flip erlauben. Eine durch einen Delaunay-Flip hinzugefügte Kante erhält die Nummer der durch den Delaunay-Flip entfernten Kante.



(b) Wieviele Flips werden vom Greedy-Algorithmus im schlimmsten Fall durchgeführt? Geben Sie eine Funktion f an, sodass die Laufzeit in $\Omega(f(n))$ ist, wobei n die Anzahl der Knoten der Triangulation ist. Begründen Sie ihre Antwort. Ein formaler Beweis ist nicht nötig.

Aufgabe 12.3: Breitensuche auf Halbkanten-Datenstruktur (6 Punkte)

Beschreiben Sie einen Algorithmus durch Pseudocode, der die Halbkanten-Datenstruktur H einer Triangulation und einen Startknoten s als Eingabe erhält und eine Breitensuche auf dieser Triangulation ausgehend von s ausführt. Nehmen Sie dabei an, dass jeder Knoten u der Halbkanten-Datenstruktur zusätzlich die Variable $u.\text{color}$, $u.\pi$ und $u.d$ speichern kann, und zu Begin für jeden Knoten u gilt $u.\text{color} = \text{weiß}$, $u.\pi = \text{null}$ und $u.d = \infty$. Geben Sie im Algorithmus explizit an, wie die adjazenten Knoten eines gegebenen Knotens ermittelt werden. Ihr Algorithmus sollte Laufzeit $O(n)$ haben.