



Übungsblatt P1

Abgabe via Moodle.
Deadline Fr. 10ter Juni

Problembeschreibung:

Gegeben ist eine Sequenz s von paarweise verschiedenen Elementen. Zu einer Gewichtsfunktion $\omega : E \rightarrow \mathbb{R}_{\geq 0}$, die jedem Element der Sequenz s ein nicht negatives Gewicht zuordnet. Zusätzlich gegeben ist eine Zahl $q \in (0, 1]$. Gesucht ist das gewichtete q -Perzentil $\phi \in s$, genau das Element mit den folgenden Eigenschaften:

$$\sum_{e \in s: e \leq \phi} \omega(e) \geq q \cdot \sum_{e \in s} \omega(e)$$

und

$$\sum_{e \in s: e \geq \phi} \omega(e) > (1 - q) \cdot \sum_{e \in s} \omega(e)$$

Ziel dieser Programmieraufgabe ist es einen Algorithmus zu entwerfen und zu implementieren, der das gewichtete q -Perzentil einer Sequenz in erwarteter linearer Laufzeit bestimmt. Die Implementierung des entworfenen Algorithmus zur Bestimmung des gewichteten q -Perzentils soll in der Programmiersprache C++ erfolgen.

Beispiel:

Das gewichtete 0.5-Perzentil der folgenden Sequenz ist das Element 73 mit Gewicht 63.

Wert e	60	29	15	91	19	77	73	95	84	41
$\omega(e)$	49	48	54	62	55	78	63	45	76	21

Aufgaben:

Benutzen Sie das in Moodle zur Verfügung gestellte Rahmenprogramm (Framework). Um das Programm zu kompilieren führen sie die Datei `compile_with_cmake.sh` aus. Im Anschluss wird die ausführbare Datei `main` in einem neuen Ordner `deploy` erzeugt. Sie können die Datei `main` ohne Argumente oder mit eigenen Parametern, z.B. $q = 0.3$, $n = 4$, $steps = 10$ und $arrazSize = 3$ ausführen:

```
./deploy/main 0.3 4 10 3
```

Um den Algorithmus zu implementieren gehen Sie wie folgt vor:

Aufgabe 1 (4 Punkte)

Entwerfen Sie zunächst einen Algorithmus, der ähnlich ist zu dem QuickSelect Algorithmus der Vorlesung und das obige Problem löst (d.h. zu einer vorgegebenen Sequenz s , einer Gewichtsfunktion ω und einer Zahl q das gewichtete q -Perzentil bestimmt). Der von Ihnen entworfene Algorithmus soll erwartete lineare Laufzeit haben. Begründen Sie die Laufzeit kurz.

Hinweis: Sie müssen das nicht formal beweisen.

Aufgabe 2 (16 Punkte)

Implementieren Sie den in Aufgabe 1 entworfenen Algorithmus, so dass eine erwartete Laufzeit von $O(n)$ erreicht wird. Die Datei `src/WeightedSelectionAlgorithm.h` enthält schon die nach außen sichtbare Signatur.

Aufgabe 3 (1 Punkte)

Führen Sie das Rahmenprogramm `main` aus, was eine tabellarische Ausgabe in Textform erzeugt. In dieser Tabelle kann man die Laufzeit in Sekunden des von Ihnen implementierten Algorithmus (Spalte 3) mit der vorgegebenen Implementierung (Spalte 2) auf eine zufälligen Eingabesequenz unterschiedlicher Länge (Spalte 1) vergleichen. Speichern Sie diese Ausgabe in die Daten `gnuplot/random.data` ab und geben Sie sie mit Ihrer Implementierung ab. Mit Hilfe des Programms Gnuplot und der beigefügten Datei `gnuplot/random.gnuplot` lässt sich ein Schaubild `gnuplot/random.png` der Ausgabe erzeugen. Bitte geben Sie auch das Schaubild ab.

Aufgabe 4 (1 Punkte)

Interpretieren Sie die Ergebnisse der Ausgaben. Ist Ihre Implementierung besser oder schlechter als der vorgegebene Algorithmus? Betrachten Sie dabei verschiedene Eingabegrößen und Eingabedaten.

Aufgabe 5 (2 Punkte)

Erläutern Sie kurz, wie man ihren Algorithmus implementieren kann, so dass er die Inplace-Eigenschaft hat.

Hinweise:

- Das Rahmenprogramm überprüft Ergebnisse auf Korrektheit. Um diesen Test für den von Ihnen geschriebenen Algorithmus zu verwenden müssen sie in der Datei `main.cpp` die Zeile 85 auskommentieren. Das ist ein guter Indikator dafür, ob Ihr Algorithmus korrekt ist, allerdings kein Beweis.
- Die Abgabe von Code beschränkt sich auf den von Ihnen erstellten Code. Das Rahmenprogramm muss nicht hochgeladen werden.
- Es wird eine zweite Programmieraufgabe mit gleicher Punktzahl geben. Zum Erhalt des Übungsscheins ist das Erreichen von 25% der Punkte pro Aufgabe notwendig.
- Täuschungsversuche werden mit 0 Punkten bestraft. Gerade bei Programmieraufgaben ist es nicht schwer, die Ergebnisse anderer Studenten zu kopieren, deswegen werden die Tutoren die abgegebenen Lösungen auch untereinander vergleichen.