

Übungsblatt 9

Abgabe bis Dienstag, den 30. Juni um 12:00 Uhr

Aufgabe 1 (10 Punkte)

Schreiben Sie ein Programm, das die Datei *cities.txt* vom letzten Übungsblatt einliest und am Ende die k Städte mit der größten Einwohnerzahl ausgibt, sortiert nach der Einwohnerzahl (die größte Stadt zuerst). Das k sollte dem Programm dabei als Argument in der Kommandozeile übergeben werden. Das Programm soll eine Prioritätswürgeschlange (PW) benutzen und in Zeit $O(n \cdot \log k)$ laufen, wobei n die Gesamtanzahl der Städte ist. Sie dürfen für die PW auf die Implementierung zurückgreifen, die Python, Java, C++ zur Verfügung stellt.

Aufgabe 2 (10 Punkte)

Implementieren Sie eine *BucketQueue*, wie in Vorlesung 9b erklärt, die die folgenden Operationen unterstützt: *insert*, *getMin*, *deleteMin*, *changeKey*. Eine Folge von n Operationen soll dabei $O(n + M)$ Zeit benötigen, wenn die Keys aus dem Bereich $0..M - 1$ sind. Alle Operationen außer *deleteMin* sollten dabei in $O(1)$ Zeit laufen. Die Datenstruktur soll zu jedem Zeitpunkt $O(n + M)$ Platz benötigen. Sie brauchen für diese Aufgabe kein *Main* Programm, es reichen die Unit Tests, die sind dafür hier umso wichtiger. Eine geeignete *LinkedList* Implementierung für Python finden Sie auf dem Wiki.

Zusatzaufgabe (5 Punkte): Erweitern Sie Ihre Implementierung so, dass der benötigte Platz $O(n + \Delta)$ ist, wenn die Menge der gespeicherten Schlüssel zu jedem Zeitpunkt aus dem Bereich $minKey..minKey + \Delta - 1$ kommen, für ein gegebenes Δ . Wenn Sie die Zusatzaufgabe machen, zählt das auch als Aufgabe 2.

SVN, Unit Tests, Jenkins und *erfahrungen.txt* wie gehabt. Warum hat die Vorlesung dieses Jahr doppelt so viele aktive Teilnehmer wie in jedem der letzten beiden Jahre? Hinweis: an der Dozentin liegt es nicht, sie hatte die Vorlesung auch im SS 2013 gehalten und auch nicht (viel) schlechter. Da waren zu diesem Zeitpunkt nur noch ca. 50 Leute dabei (im SS 2014 genauso), jetzt sind es noch über 100.