

Auszuarbeiten bis 11.10.17

1. O-Notation (4 Punkte)

In der Vorlesung wurden die unterschiedlichen Komplexitätsklassen besprochen. Suchen Sie in „echten“ Quellen für die Komplexitätsklassen $O(1)$, $O(n)$, $O(\log(n))$ und $O(n^2)$ jeweils zwei Algorithmen. Überlegen Sie sich auch eine Begründung für die Einordnung in die jeweilige Klasse.

2. Grobanalyse (5 + 3 + 3 Punkte)

Entwickeln Sie die folgenden Funktionen und analysieren Sie diese infolge.

Bestimmen Sie den Best, Worst und Average-Case (sofern möglich)

`int findCharLeft(String str, char ch)` bestimmt die Position des ersten Zeichens von links mit dem Wert `ch` in der Zeichenkette `str`.

`int findCharRight(String str, char ch)` bestimmt die Position des ersten Zeichens von rechts mit dem Wert `ch` in der Zeichenkette `str`.

`int findCharRandom(String str, char ch)` bestimmt die Position des ersten Zeichens mit dem Wert `ch` in der Zeichenkette `str` anhand zufällig ausgewählter Positionen.

Implementieren Sie die beiden Funktionen Java. Achten Sie auf eine möglichst effiziente Implementierung.

3. (De-)komprimierung eines Textes (5 Punkte)

Wie auf jedem Amt, fallen auch in der Stadtverwaltung von Konstantinopel täglich Unmengen von Dokumenten an. Der neue Informatikleiter der Stadtverwaltung hat, gezwungen durch die Misswirtschaft seines Vorgängers, für das Jahr 2016 jeglichen Ankauf von neuem Festplattenplatz auf den Servern untersagt. Prof. Mayar Derwisch von der Fachhochschule Konstantinopel wurde daher von der Stadtverwaltung mit einem Spezialauftrag, der Entwicklung eines (De-)komprimierungsverfahrens, betraut.

Prof. Mayar hat sich die Sache so vorgestellt: Wiederholt sich ein Zeichen mindestens drei Mal, so wird das Zeichen nur einmal ausgegeben. Nach dem Zeichen wird die Häufigkeit ausgegeben, mit der das Zeichen im Originaltext hintereinander vorkommt. z.B.: `abbccdddeeffggg` \Rightarrow `abbc3d4eefg3`. Im Text können Buchstaben und Zahlen vorkommen!

Entwickeln Sie einen Komprimierungs- und Dekomprimierungs-Algorithmus.

Hinweise

Vergessen Sie nicht auf die Lösungsidee und testen Sie Ihre Programme ausführlich und automatisiert!