UE DGA WS2020-2021
Übungsblatt 9

Aufgabe 49:

a Erzeugt die folgende Modifikation des Fisher-Yates-Algorithmus ebenfalls eine zufällige Permutation von A? Weshalb oder weshalb nicht?

PERMUTE-WITH-ALL($A[1,\ldots,n]$)
$n = A$.lenght
for $i = 1$ to n **do**
  vertausche $A[i]$ mit $A[RANDOM(1,n)]$
end for

b Eine weitere Abwandlung des Fisher-Yates-Algorithmus: Diesmal wird das Elememt $A[i]$ in jedem Schritt mit einem zufälligen Element aus dem Teilfeld $A[i+1,\ldots,n]$ vertauscht, also

PERMUTE-WITHOUT-IDENTITY($A[1,\ldots,n]$)
$n = A$.lenght
for $i = 1$ to n **do**
  vertausche $A[i]$ mit $A[\mathrm{RANDOM}(i+1,n)]$
end for

Man könnte zunächst meinen, dass dieser Algorithmus alle von der Identität verschiedenen Permutaionen zufällig erzeugt. Weisen Sie nach, dass dies aber nicht der Fall ist. Überlegen Sie sich, welche Klasse von Permutationen von diesem Algorithmus tatsächlich zufällig erzeugt werden.

Aufgabe 50:
Bestimmen Sie eine Rekursion fur die mittlere Anzahl der Schlüsselvergleiche beim (nicht-randomisierten) Quicksort, wenn anstelle des letzten Elements $A[n]$ das der Größe nach mittlere der drei Elemente $A[1], A[\lfloor \frac{n}{2} \rfloor]$ und $A[n]$ als Pivot-Element verwendet wird. Das Lösen dieser Rekursion ist nicht verlangt.

Aufgabe 51:
Quickselect ist ein Algorithmus zum Auffinden des j-kleinsten Elements eines Feldes, der auf einer ähnlichen Idee wie Quicksort basiert: Wähle ein zufälliges Pivotelement und bringe es wie bei Quickselect an die richtige Stelle, sodass im Teilfeld links davon nur kleinere Elemente und im Teilfeld rechts davon nur größere Elemente sind. Ist das Pivotelement genau an Stelle j, so sind wir fertig. Hat das linke Teilfeld mehr als j Elemente, so suche dort weiter, ansonsten im rechten Teilfeld (anstatt wie bei Quicksort beide Seiten rekursiv zu betrachten, macht man dies jetzt nur mit der Seite, wo das Element ist).
Schreiben Sie einen Pseudocode für den Algorithmus Quickselect.

Aufgabe 52:

Zur Analyse der erwarteten Laufzeit von Quickselect: Bezeichne $T(n)$ die Laufzeit von Quickselect bei einem Feld der Größe n und sei X_k das Ereignis, dass das Pivotelement an k-ter Stelle steht.

a Weisen Sie nach, dass $\mathbb{E}(X_k) = \frac{1}{n}$ für $k = 1, \ldots, n$ und dass für $T(n)$ gilt, dass

$$T(n) \leq \sum_{k=1}^{n} X_k \cdot T(\max(k-1, n-k)) + O(n).$$

b Da X_k und $T(\max(k-1, n-k))$ unabängig sind, lässt sich also folgern, dass

$$\mathbb{E}(T(n)) \leq \sum_{k=1}^{n} \frac{1}{n} \cdot \mathbb{E}(T(\max(k-1, n-k))) + O(n).$$

Zeigen Sie (mittels Betrachtung von $\max(k-1, n-k)$), dass

$$\mathbb{E}(T(n)) \leq \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n} \mathbb{E}(T(k)) + O(n).$$

c Folgern Sie daraus nun mittels Substituionsmethode, dass $\mathbb{E}(T(n)) = O(n)$ (d.h. $\mathbb{E}(T(n)) \leq cn$ für passendes $c > 0$).

Aufgabe 53:

Quick-Hull ist ein auf einer ähnlichen Idee wie Quicksort basierender Divide-and-Conquer-Algorithmus zum Auffinden der konvexen Hülle einer endlichen Punktmenge im $\mathbb{R}^2$. Er arbeitet wie folgt: Zunächst werden die beiden Punkte mit der größten und der kleinsten x-Koordinate gesucht (sollte es mehrere Punkte mit kleinster/größter x-Koordinate geben, so wähle denjenigen mit kleinster y-Koordinate). Da diese Punkte Extremalpunkte sind, sind sie Bestandteil der konvexen Hülle.
Die beiden gefundenen Punkte bilden eine Gerade, die die Punktmenge in zwei Teilmengen teilt: die Punkte rechts von der Gerade und die Punkte links davon (links und rechts ergeben sich aus dem Winkel zwischen dem Richtungsvektor der Geraden und dem Vektor zwischen Anfangspunkt der Gerade und dem betrachteten Punkt). Diese beiden durch die Gerade getrennten Punktmengen werden nun von Quick-Hull rekursiv betrachtet. Innerhalb der zu betrachtenden Punktmenge wird der Punkt P gesucht, der die maximale Distanz zur Geraden hat. Dieser ist ebenfalls Teil der konvexen Hülle. Das Dreieck bestehend aus den Endpunkten der Gerade und dem Punkt P besteht aus drei Punkten, die alle zur konvexen Hülle gehören. Also können wir alle Punkte im Inneren des Dreiecks bei weiteren Aufrufen des Algorithmus ignorieren.
Die Seiten des Dreiecks fungieren nun als neue Trenngeraden. Der Algorithmus wird so lange wiederholt, bis nur noch die Endpunkte der Trenngerade Teil der zu untersuchenden Punktmengen sind.

a Illustrieren Sie die Arbeitsweise des Algorithmus anhand einer selbst gewählten Punktmenge mit mindestens 10 Punkten.

b Begründen Sie, warum $O(n^2)$ die Worst-Case-Laufzeit von Quick-Hull ist.

Aufgabe 54:
Wiederholen Sie an Hand des folgenden linearen Programms, wie allgemeine lineare Programme in Standardform gebracht werden können.

$$
\begin{aligned}
&\text{Minimiere } 2x_1 + 7x_2 + x_3 \text{ unter den Nebenbedingungen}\\
&\quad x_1 - x_3 = 7,\\
&\quad 3x_1 + x_2 \geq 24,\\
&\quad x_2 \geq 0,\\
&\quad x_3 \leq 0.
\end{aligned}
$$