

4. Übungsblatt zur Vorlesung Entwurf und Analyse von Algorithmen, WS 12/13

Abgabe: Bis Donnerstag, 08.11.2012, 12:00 Uhr, Abgabekasten vor 48-694.

17. Aufgabe

Track ϵ 2
Track AI [2]

Seien X und Y zwei Zufallsvariablen über \mathbb{N}_0 mit Wahrscheinlichkeiten-Erzeugendenfunktionen P_X bzw. P_Y . Sei außerdem $Z = X + Y$ mit P_Z . Zeigen oder widerlegen Sie:

$$X, Y \text{ unabhängig} \implies P_Z(z) = P_X(z)P_Y(z).$$

18. Aufgabe

a) b) c)
Track ϵ 3 5 [5]
Track AI 3 5 [5]

Wir betrachten Arrays und entsprechende Speicherabbildungsfunktionen:

- a) Finden Sie eine sequentielle Speicherabbildungsfunktion, welche zu einer beliebigen $n \times n$ Matrix eine Erweiterung zu einer $(n+k) \times (n+k)$ -Matrix, $k \geq 1$, ohne Umstellung der vorhandenen Elemente zulässt.
- b) Beschreibe $A : [0 : m] \times [0 : m] \rightarrow \mathbb{Z}$ ein 2-dimensionales Feld mit den Elementen $A[i_1, i_2]$, $0 \leq i_1 + i_2 \leq m$.

1. Bestimmen Sie eine sequentielle Speicherabbildungsfunktion

$$\text{LOC}_A : \{(i_1, i_2) \in [0 : m] \times [0 : m] \mid 0 \leq i_1 + i_2 \leq m\} \rightarrow [N : M].$$

Beweisen Sie, dass Ihre Lösung sequentiell ist! Wie ist M zu wählen?

2. Angenommen, man ließe $0 \leq i_1 + i_2 \leq 2m$ zu. Wie würde sich diese Erweiterung auf die Eigenschaften Ihrer Speicherabbildungsfunktion aus 1. auswirken?
- c) Sei $A : [1 : m] \times \dots \times [1 : m] \rightarrow \mathbb{Z}$ ein k -dimensionales Feld mit den Elementen $A[i_1, i_2, \dots, i_k]$, $1 \leq i_k \leq i_{k-1} \leq \dots \leq i_2 \leq i_1 \leq m$ (Verallgemeinerung der unteren Dreiecksmatrix).

Bestimmen Sie eine lexikographische Speicherabbildungsfunktion

$$\text{LOC}_A : \{(i_1, \dots, i_k) \in [1 : m] \times \dots \times [1 : m] \mid 1 \leq i_k \leq \dots \leq i_1 \leq m\} \rightarrow [N : M].$$

Wie ist M zu wählen?

19. Aufgabe

	a)	b)	c)
Track ϵ	2	[5]	[5]
Track AI	2	5	[5]

- a) Es sei eine $n \times n$ -Matrix $A : [1 : n] \times [1 : n]$ gegeben, in der jedes Element $A[i, j]$ die Werte **Wasser** oder **Ente** annehmen kann. Zwei Enten gehören zu derselben Entenfamilie, wenn sie entweder in horizontal, vertikal oder diagonal benachbarten Feldern "schwimmen". Entwerfen Sie einen Algorithmus, der folgendes leistet: Bei Eingabe (i, j) , $1 \leq i, j \leq n$, berechne die Größe der Entenfamilie, zu der die Ente an Position (i, j) gehört, sofern $A[i, j] = \text{Ente}$; ansonsten gib 0 zurück.

Was ist die Zeitkomplexität Ihres Algorithmus?

- b) Wir wollen herausfinden, wie stark je zwei Enten verwandt sind. Den Verwandtschaftsgrad modellieren wir mit
- der kleinsten Zahl k an Schritten (von einem Feld zu einem benachbarten, wie in a)), um von Ente (i_1, j_1) zu Ente (i_2, j_2) zu gelangen, *wenn die beiden zu einer Familie gehören*, und sonst
 - der kleinsten Zahl k an Schritten über Wasser¹, um von einem Verwandten der Ente (i_1, j_1) zu einem Verwandten von Ente (i_2, j_2) zu gelangen. In diesem Fall wollen wir $-k$ zurückgeben.

Entwerfen Sie einen Algorithmus, der den Verwandtschaftsgrad für zwei gegebene Enten(indizes) berechnet!

Was ist die Zeitkomplexität Ihres Algorithmus?

- c) Martha möchte die Entenfamilie nicht mehr in ihrem Teich haben. Sie beauftragt Ihren Mann Kurt damit, der Plage ein Ende zu bereiten. Kurt hat Skrupel und möchte so wenige Enten wie möglich töten. Er recherchiert und findet heraus, dass eine Entenfamilie die Heimat wechselt, wenn man sie in (mindestens zwei) kleinere Familien zerschlägt.

Entwerfen Sie einen Algorithmus für Kurt! Gegeben eine Teichmatrix **A** mit genau einer Entenfamilie soll dieser die Mindestanzahl Enten zurückgeben, die Kurt töten muss (und idealerweise deren Positionen).

Was ist die Zeitkomplexität Ihres Algorithmus?

Hinweis: In allen Algorithmenentwurfsaufgaben gilt: Stellen Sie zu Beginn die Kernidee(n) Ihres Algorithmus heraus; damit erleichtern Sie dem Korrektor das Verständnis ungemein. Der Algorithmus selbst sollte in *klarem, definiten* Pseudocode angegeben werden. Die Faustregel ist, dass ein Programmierer ohne Kenntnis der Problemstellung in der Lage sein sollte, den Algorithmus anhand Ihres Pseudocodes ohne Nachdenken zu implementieren.

¹Schritte über Enten sind erlaubt, zählen aber nicht. Intuition: es müssen noch (mindestens) k Enten zielsicher in die Welt gesetzt werden, um die Familien zu vereinen.

20. Aufgabe

	a)	b)
Track ε	2	2
Track AI	2	[2]

- a) Nehmen Sie an, Sie dürfen ausschließlich die Datenstruktur **Stack** verwenden, ohne daran Veränderungen vorzunehmen. Wie können Sie dann unter Verwendung zweier Stacks *effizient* eine Queue implementieren?

Effizienz heißt hier, dass jedes Element zwischen Eintritt in die Queue bis zum Verlassen der Queue nur konstant oft *gepusht* oder *gepoppt* wird, und zwar unabhängig davon, wie viele Elemente dazwischen in die Queue eingefügt oder aus der Queue entfernt werden.

Für die Lösung dieser Aufgabe soll *kein* Programm geschrieben werden. Es genügt, wenn Sie die Idee einer Implementierung detailliert ausarbeiten. Dabei ist zu zeigen, dass Ihre Lösung tatsächlich effizient ist.

- b) Lässt sich auch umgekehrt ein Stack durch zwei Queues effizient implementieren? Wie bzw. warum nicht?