Übungsblatt 02

Algorithmen I - Sommersemester 2024

Abgabe im ILIAS bis 03.05.2024, 18:00 Uhr

Die Abgabe erfolgt alleine oder zu zweit als eine PDF-Datei über das Übungsmodul in der Gruppe eures Tutoriums im ILIAS. Beschriftet die Abgabe deutlich mit Matrikelnummer und Name.

- Achtet bei handschriftlichen Abgaben auf Lesbarkeit.
- Achtet darauf, ob Algorithmen in Worten oder Pseudocode beschrieben werden sollen.
- Es werden immer asymptotisch möglichst effiziente Algorithmen erwartet, wenn nicht anders angegeben.
- Wenn Korrektheits- oder Laufzeitanalysen gefordert sind, behandelt diese separat von der Algorithmenbeschreibung.

Gesamtpunkte: 20

Hinweis: Schaut euch gerne die Algo-Code Aufgabe zu dynamischen Arrays an. Sie eignet sich besonders gut, um sich mehr mir amortisierten Analysen auseinanderzusetzen. `https://algo-code.iti.kit.edu/array/dynamic-array/`

Aufgabe 1 - Knoten lösen (5 Punkte)

In dieser Aufgabe wollen wir das asymptotische Verhalten von Laufzeitfunktionen mittels ihres Rekursionsbaums bestimmen. Analysiere dafür den Rekursionsbaum, indem du folgende Fragen für jede rekursive Funktion beantwortest:

- Wie viele Lagen hat der Rekursionsbaum?
- Wie viele Knoten befinden sich auf Lage i?
- Wie groß ist der Funktionsparameter auf Lage i?
- Wie viel Zeit kostet ein Knoten auf Lage i?

Stelle mithilfe deiner Antworten eine Summe auf, die die Gesamtlaufzeit der Funktion angibt. Bestimme anhand der Summe das asymptotisch Verhalten der Funktion.

a) $T_1(n) = \begin{cases} 1 & \text{falls } n = 1, \\ 27 \cdot T_1(\frac{n}{3}) + 3 \cdot n^3 & \text{sonst.} \end{cases}$ (1.5 Punkte)

b) $T_2(n) = \begin{cases} 3 & \text{falls } n = 1, \\ 32 \cdot T_2(\frac{n}{4}) + n \cdot (\log(n) + 1) & \text{sonst.} \end{cases}$ (2 Punkte)

c) $T_4(n) = \begin{cases} 1 & \text{falls } n = 1, \\ 11 \cdot T_4(\frac{n}{7}) + n \cdot \sqrt{n} & \text{sonst.} \end{cases}$ (1.5 Punkte)

Aufgabe 2 - Angriff der Klonbiber (10 Punkte)

Um seinen Geheimplan in die Tat umsetzen zu können, benötigt Dr. Meta eine große Menge an Bibern. Leider steht ihm zur Zeit nur eine kleine Zahl der flauschigen Tiere zur Verfügung. Dr. Meta ist es allerdings gelungen, eine Klonmaschine zu entwickeln, die es ihm erlaubt, seine Biber zu vervielfachen. Aufgrund technischer Schwierigkeiten kann die Maschine aktuell jeden Biber nur ein Mal klonen, daher unterscheiden wir zwischen originalen (das heißt nicht geklonten) Bibern und den geklonten Exemplaren. Wir bezeichnen die Anzahl der originalen Biber, die sich zu einem bestimmten Zeitpunkt in der Maschine befinden, mit α und analog die Anzahl der geklonten Biber mit β. Die Maschine unterstützt die folgenden Operationen (die Kosten hängen von dem Zustand der Maschine vor Aufruf der Operation ab):

- Load: Lädt einen bereits existierenden originalen Biber in die Maschine, benötigte Energie 1.
- Clone: Klont die Biber in der Maschine, indem jeder originale Biber durch zwei geklonte ersetzt wird. Bereits geklonte Biber werden <u>nicht</u> geklont, benötigte Energie $2 \cdot \alpha$.
- Empty: Entfernt alle Biber aus der Maschine, benötigte Energie $4 \cdot \alpha + \beta$.

Da die Stromkosten in dem letzten Jahr stark gestiegen sind, macht sich Dr. Meta allerdings Sorgen wegen des hohen Energieverbrauchs der Clone-Operation.

a) Gib für die folgenden beiden Folgen von Operationen die asymptotischen Gesamtkosten abhängig von k an:

(i) $k \cdot$Load
Clone
Empty

(1 Punkt)

(ii) $k \cdot$Load
Clone
$2 \cdot k \cdot$Load
Clone
Empty

(1 Punkt)

b) Zeige, dass der amortisierte Energieverbrauch für eine beliebige Abfolge von Operationen auf der zu Beginn leeren Maschine bei $\Theta(1)$ pro Operation liegt. Nutze dafür Aggregation, Charging oder die Kontomethode. (2 Punkte)

Dr. Meta ist zwar höchst erfreut über deine Ergebnisse, ist allerding mit den von dir genutzten Methoden nicht zufrieden. Er ist nicht von der Überzeugung abzubringen, dass nur mit der Potentialmethode das Potential der Maschine voll ausgeschöpft werden kann. Du sollst daher den Beweis mit dieser Methode wiederholen. Die hilfreichen Biber haben bereits etwas an dem Beweis herumgetüftelt. Dabei haben sie allerdings einiges durcheinander gebracht.

c) Bringe die untenstehenden Beweischnipsel in die richtige Reihenfolge. Außerdem haben die Bibern wohl einige Teile eines anderen Beweises mit den Schnipseln vermischt. Entferne die nicht benötigten Abschnitte. (3 Punkte)

Hinweis: Die Musterlösung enthält 14 Aussagen.

(1) Dadurch haben wir eine Potentialänderung von $\Phi(M_{\text{nach}}) - \Phi(M_{\text{vor}}) = 0 - 2 \cdot \alpha = -2 \cdot \alpha$.

(2) Wir nutzen die Potentialfunktion $\Phi(M) = \beta - \alpha$. Da die Anzahl der originalen bzw. geklonten Biber in der Maschine nicht-negativ ist, ist auch die Potentialfunktion gültig.

(3) Somit liegen die amortisierten Kosten bei $1 + 4 = 5 \in \Theta(1)$.

(4) Somit liegen die amortisierten Kosten bei $4 \cdot \alpha + \beta + \alpha - \beta = 5 \cdot \alpha \in \Theta(\alpha)$.

(5) Damit zeigen wir, dass die amortisierten Kosten jeder Operation konstant sind, indem wir die tatsächlichen Kosten einer Operation zusammen mit ihrer Potentialänderung betrachten.

(6) Wir laden einen originalen Biber ein und haben damit eine Potentialänderung von $\Phi(M_{\text{nach}}) - \Phi(M_{\text{vor}}) = (4 \cdot (\alpha + 1) + \beta) - (4 \cdot \alpha + \beta) = 4$.

(7) Wir nutzen die Potentialfunktion $\Phi(M) = 2 \cdot \alpha$. Da die Anzahl der originalen Biber in der Maschine nicht-negativ ist, ist auch die Potentialfunktion gültig.

(8) Somit liegen die amortisierten Kosten bei $2 \cdot \alpha - 2 \cdot \alpha = 0 \in \Theta(1)$.

(9) Durch diese Operation werden alle α originalen Biber und alle β geklonten Biber aus der Maschine entfernt.

(10) Somit liegen die amortisierten Kosten bei $2 \cdot \alpha - 2 \cdot \alpha = 0 \in \Theta(1)$.

(11) Clean:

(12) Wir laden einen originalen Biber ein und haben damit eine Potentialänderung von $\Phi(M_{\text{nach}}) - \Phi(M_{\text{vor}}) = 1$.

(13) Somit liegen die amortisierten Kosten bei $4 \cdot \alpha + \beta - (4 \cdot \alpha + \beta) = 0 \in \Theta(1)$.

(14) Load:

(15) Empty:

(16) Wir nutzen die Potentialfunktion $\Phi(M) = 4 \cdot \alpha + \beta$. Da die Anzahl der originalen bzw. geklonten Biber in der Maschine nicht-negativ ist, ist auch die Potentialfunktion gültig.

(17) Für die Maschine M bezeichen wir mit M_{vor} den Zustand vor und mit M_{nach} den Zustand nach einer Operation.

(18) Dadurch haben wir eine Potentialänderung von $\Phi(M_{\text{nach}})-\Phi(M_{\text{vor}}) = (0 + (2 \cdot \alpha + \beta)) - (4 \cdot \alpha + \beta) = -2 \cdot \alpha$.

(19) Durch diese Operation ersetzen wir die α originalen Biber durch $2 \cdot \alpha$ geklonte Biber.

(20) Dadurch haben wir eine Potentialänderung von $\Phi(M_{\text{nach}}) - \Phi(M_{\text{vor}}) = 0 - (4 \cdot \alpha + \beta) = -(4 \cdot \alpha + \beta)$.

(21) Clone:

(22) Dadurch haben wir eine Potentialänderung von $\Phi(M_{\text{nach}}) - \Phi(M_{\text{vor}}) = 0 - (\beta - \alpha) = \alpha - \beta$.

Während du den Beweis zusammengesetzt hast, hat Dr. Meta an der Maschine herumgeschraubt. Nun erzeugt Clone aus jedem originalen Biber drei geklonte Biber. Allerdings entfernt Empty nur noch die geklonten Biber und $\frac{4}{5}$ der originalen Biber aus der Maschine. Somit bleiben $\frac{1}{5}$ der originalen Biber nach der Ausführung der Operation in der Maschine[1]. Load und die Energiekosten der Operationen bleiben unverändert.

d) Wende nun das Gelernte an und zeige mit der Potentialmethode, dass der amortisierte Energieverbrauch jeder Operation in der von uns betrachteten Abfolge noch immer konstant ist. (3 Punkte)

Hinweis: Falls du Probleme hast eine Potentialfunktion zu finden: Du kannst die Funktion aus b) leicht abwandeln. Überlege dir dafür, was schief geht, wenn du die alte Potentialfunktion nutzt und passe diese dann an, um den Fehler zu beheben.

Aufgabe 3 - Quite sum problem (5 Punkte)

Wir wollen in dieser Aufgabe zwei Arrays A und B der Länge n von Ganzzahlen betrachten.

Für ein halboffenes Intervall $[i, j)$ mit $i \in \{0, \ldots, n-1\}$ und $j \in \{0, \ldots, n\}$, und Array $X \in \{A, B\}$ bezeichnen wir mit $\text{sum}_X(i, j) = \sum_{k=i}^{j-1} X[k]$ die Summe der Einträge zwischen i und j. Leere Summen sind gleich 0, also gilt z.B. auch $\text{sum}_X(0, 0) = 0$. In dieser Aufgabe gelten für alle Vorkommen von i, j und X die obigen Einschränkungen.

Wir wollen nun bestimmen, für welche i, j die Summen $\text{sum}_A(i, j)$ und $\text{sum}_B(i, j)$ gleich sind. Unser Algorithmus soll den größten solchen Bereich finden. Dafür bezeichnen wir mit $j - i$ die Länge des Bereichs zwischen i und j.

[1]Einfachheitshalber nehmen wir an, dass Empty nur aufgerufen wird, falls die Division ohne Rest abläuft.

a) Betrachte die beiden folgenden Arrays und bestimme das längsten Intervall $[i, j)$ mit $\text{sum}_A(i, j) = \text{sum}_B(i, j)$. Gebe hierfür die Grenzen i, j und die Länge des Intervalls an. (1 Punkt)

0	1	2	3	4	5	6
1	2	5	−3	5	0	6

0	1	2	3	4	5	6
−1	3	4	0	1	2	3

Die beiden Arrays mit Indizes.

Um das Problem zu lösen, wollen wir Präfixsummen als Hilfsmittel verwenden. Eine Präfixsumme ist die Summe $\text{sum}_X(0, j)$ für das Intervall $[0, j)$ und Array X.

b) Beschreibe in Worten einen Algorithmus, der in Zeit $\mathcal{O}(n)$ für alle Indizes die jeweilige Präfixsumme eines Arrays berechnet. Begründe warum dein Algorithmus die geforderte Laufzeit erreicht. (1.5 Punkte)

c) Nach Ausführung deines Algorithmus aus c) stehen dir die Präfixsummen eines Arrays für beliebige Intervalle $[0, j)$ beginnend bei 0 zur Verfügung. Wie kannst du mithilfe dieser Präfixsummen die Summe $\text{sum}_X(i, j)$ berechnen? (1 Punkt)

d) Beschreibe nun einen Algorithmus der in Zeit $\mathcal{O}(n^2)$ das längste Intervall mit gleicher Summe findet. (1.5 Punkte)

Aufgabe 4 - Zum Knobeln (0+3 Punkte)

Wir wollen jetzt einen Linearzeitalgorithmus für das Problem aus Aufgabe 3 finden. Im Folgenden steht dir eine magische Blackbox M zur Verfügung, die eine Folge von Werten speichern kann. Du kannst in jeweils konstanter Zeit Elemente in diese Blackbox einfügen und überprüfen, ob ein gegebenes Element in der Blackbox vorhanden ist.[2]

a*) In welchem Verhältnis stehen $\text{sum}_A(0, i) - \text{sum}_B(0, i)$ und $\text{sum}_A(0, j) - \text{sum}_B(0, j)$ zu einander, wenn:

- $\text{sum}_A(i, j) = \text{sum}_B(i, j)$?
- $\text{sum}_A(i, j) \neq \text{sum}_B(i, j)$?

(1 Punkt)

b*) Gebe einen Linearzeit Algorithmus an, der das längste zusammenhängende Intervall $[i, j)$ mit $\text{sum}_A(i, j) = \text{sum}_B(i, j)$ bestimmt. Beschreibe diesen Algorithmus in Worten. (2 Punkte)

[2]Klausurvorbereitung für den Rest des Semesters: Überlege dir bei Einführung einer neuen Datenstruktur, ob du mit dieser eine solche Blackbox implementieren kannst bzw. was die Laufzeiten der Operationen wären.

Fachschaftsveranstaltungen für Erstis

Du interessierst dich für die Arbeit der Fachschaft und möchtest dich vielleicht gerne selbst engagieren? Schau einfach vorbei am **25. April um 19:00 Uhr** beim **Semesterauftakttreffen**. Hier zeigen wir dir, wie die Fachschaft organisiert ist, was ihre Aufgaben sind und wie du dich bei uns einbringen kannst.