Universität
des
Saarlandes
FR 6.2 Informatik

Kurt Mehlhorn und Corinna Coupette WiSe 2021/22

Übungen zu Ideen der Informatik

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter21/ideen/

Blatt 5: Websuche **Abgabeschluss: 29.11.2021**

Aufgabe 1 Terminologie *(5 Punkte)*

a) Beantworten Sie *in Ihren eigenen Worten*:

(1) Was ist Suchen? (*1 Punkt*)

Lösung:
Siehe Slides. Mögliche Antwort: Gegeben eine Menge von Daten und ein gesuchtes Datum, beantworte die Frage, ob das gesuchte Datum in der Menge von Daten enthalten ist.

(2) Was ist Sortieren? (*1 Punkt*)

Lösung:
Siehe Slides. Mögliche Antwort: Gegeben eine Menge von Daten und ein Kriterium, nach dem die zwei Daten miteinander verglichen werden können, ordne die Menge von Daten nach diesem Kriterium.

(3) Wie hängen Suchen und Sortieren zusammen? (*1 Punkt*)

Lösung:
Siehe Slides. Mögliche Antwort: In einer Menge von Daten, die sortiert ist, kann man effizienter suchen.

b) Nennen Sie die in der Vorlesung eingeführten Algorithmen zur Suche in Listen mit ihrer jeweiligen Laufzeit und den Anforderungen an die Eingabe. (*1 Punkt*)

Lösung:
Siehe Slides. Lineare Suche (keine Anforderungen): $\mathcal{O}(n)$; Binärsuche (Anforderung: Liste sortiert): $\mathcal{O}(\log n)$.

c) Nennen Sie die in der Vorlesung eingeführten Algorithmen zur Sortierung von Listen mit ihrer jeweiligen Laufzeit. (*1 Punkt*)

Lösung:
Siehe Slides. Sortieren durch Mischen: $\mathcal{O}(n \log n)$; Quicksort: $\mathcal{O}(n^2)$ (schlechtester Fall), $\mathcal{O}(n \log n)$ (bester Fall).

Aufgabe 2 Buchstabenliste *(15 Punkte)*

Betrachten Sie die folgende Buchstabenliste.

`buchstaben` =	B	E	Z	N	G	H	S	X	D	Q	A	C	P	W	K
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

a) Sie möchten prüfen, ob der Buchstabe S in der Liste enthalten ist. Ihr Kommilitone schlägt vor, die Binärsuche zu verwenden.

(1) Mit welchen Buchstaben würde der Buchstabe S verglichen, wenn Sie dem Vorschlag des Kommilitonen folgten? Geben Sie die Buchstaben in der Reihenfolge an, in der die Vergleiche stattfinden. (*2 Punkte*)

Lösung:
X, N, H, S.

(2) Würde der Vorschlag des Kommilitonen in diesem konkreten Fall funktionieren (mit Begründung)? (*1 Punkt*)

Lösung:
Ja, denn S wird gefunden, siehe voriger Aufgabenteil.

(3) Würde der Vorschlag des Kommilitonen im Allgemeinen funktionieren (mit Begründung)? (*2 Punkte*)

Lösung:
Nein, denn es ist hier Glück, dass die vorgenommenen Vergleiche immer richtig ausfallen. Würde man z.B. nach A suchen, würde man fälschlicherweise nicht fündig werden, da X > A, A aber in der Liste rechts von X steht. Die Korrektheit der Binärsuche setzt voraus, dass die Liste sortiert ist.

b) Betrachten Sie Algorithmus 1.

Algorithmus 1 : Ordnung von Buchstaben

```
1 buchstaben ← Eingabe (in Form einer Liste)
2 n ← Länge von buchstaben
3 for runde von 0 bis n − 1 do
4     for position von 0 bis n − 2 do
5         if buchstaben[position] < buchstaben[position+1] then
6             temp ← buchstaben[position];
7             buchstaben[position] ← buchstaben[position+1];
8             buchstaben[position+1] ← temp;
```

(1) Führen Sie eine Runde der äußeren for-Schleife des Algorithmus (das ist die for-Schleife ab Z. 3) mit der oben angegebenen Liste als Eingabe aus und geben Sie an, wie die Liste nach dieser Runde aussieht. (*2 Punkte*)

Lösung:

`buchstaben` =	E	Z	N	G	H	S	X	D	Q	B	C	P	W	K	A
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

(2) Wie sieht die Liste aus, wenn Sie den Algorithmus ganz ausgeführt haben? (*2 Punkte*)

Lösung:
Die Liste ist absteigend sortiert.

(3) Was können Sie über die Liste mit Sicherheit sagen, wenn Sie die äußere for-Schleife x Mal ausgeführt haben? (*2 Punkte*)

Lösung:
Die kleinsten x Elemente befinden sich am Ende der Liste.

(4) Wie viele Vergleiche (Z. 5) führt der Algorithmus für eine Liste der Länge n aus (mit Begründung)? (*2 Punkte*)

Lösung:
Die innere Schleife bewirkt, dass in jeder Runde (man sagt auch: in jeder *Iteration*) der äußeren Schleife $n-1$ Vergleiche ausgeführt werden. Die äußere Schleife wird n Mal ausgeführt. Der Algorithmus führt also $n \cdot (n-1) \in \mathcal{O}(n^2)$ Vergleiche durch.

(5) Wie müssten Sie die Binärsuche aus der Vorlesung anpassen, damit Sie auf einer mit Algorithmus 1 sortierten Liste funktioniert (mit Begründung)? (*2 Punkte*)

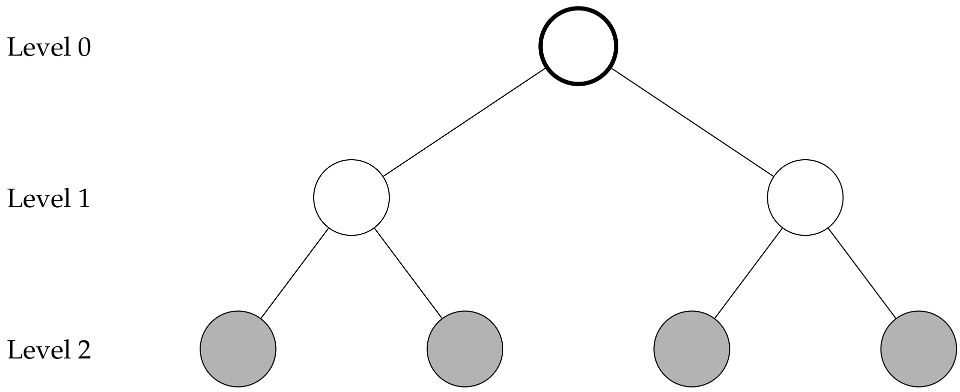
Lösung:
Da Algorithmus 1 die Liste absteigend sortiert, müssen Sie die Binärsuche so anpassen, dass Sie von einer absteigend sortierten Liste ausgeht. Das können Sie z.B. erreichen, indem Sie die Verwendung der Vergleichsoperatoren $<$ und $>$ umdrehen.

Aufgabe 3 Bäume *(10+3 Punkte)*

Betrachten Sie die nachfolgende Zeichnung. Das ist ein *Baum*, genauer gesagt ein *vollständiger Binärbaum*. In der Informatik wachsen Bäume (sofern sie eine Wurzel haben) von oben nach unten, ähnlich der Darstellung von Stammbäumen. Die Kreise heißen *Knoten* und die Linien heißen *Kanten*. Gewöhnen Sie sich an diese Terminologie; wir werden sie noch häufiger benötigen.

Der Knoten auf Level 0 heißt *Wurzel* und ist in der Abbildung dick umrandet; die Knoten auf dem untersten Level (hier: Level 2) heißen *Blätter* und sind in der Abbildung grau markiert. Für einen Knoten v auf Level l heißen die Knoten auf Level $l + 1$, mit denen v direkt verbunden ist, *Kinder* von v.

Hinweis: Der vorige Satz war abstrakter als die ihm vorangehenden Sätze. Lesen Sie ihn ggf. noch einmal und stellen Sie sicher, dass Sie verstehen, was er bedeutet. In der Informatik ist es ganz normal, dass man manchmal Dinge mehrfach lesen muss, um sie zu verstehen.



a) Tragen Sie die Zahlen 1 bis 7 so in den Baum ein, dass für jeden Knoten v mit linkem Kind c_L und rechtem Kind c_R gilt: *Die Zahl, die c_L zugeordnet ist, ist kleiner als die Zahl von v, und die Zahl, die c_R zugeordnet ist, ist größer als die Zahl von v.* (*2 Punkte*)

   Lösung:
   Von links nach rechts: 1–7.

   Hinweis: Die Aufgabe war nicht gut gestellt; die Lösung ist hier nicht eindeutig. Die angegebene Regel müsste lauten: Alle Zahlen in c_L und darunter sind kleiner als die Zahl von v, und alle Zahlen in c_R und darunter sind größer als die Zahl von v. *Dies hat Konsequenzen aber nur für die Aufgabenteile a), b) und f).*

b) Welchen Wert hat das rechte Kind des Knotens, dem Sie die 4 zugeordnet haben? (*1 Punkt*)

   Lösung:
   6.

c) Wie viele Blätter hat der oben abgebildete vollständige Binärbaum? (*1 Punkt*)

   Lösung:
   4.

d) Ein vollständiger Binärbaum ist dadurch gekennzeichnet, dass alle Blätter auf demselben Level liegen und jeder Knoten, der kein Blatt ist, exakt zwei Kinder hat. Wie viele *Blätter* hat ein Binärbaum, dessen unterstes Level das Level 3 ist (mit Begründung)? (*1 Punkt*)

Lösung:
$2^3 = 8$.

e) Auf welchem *Level* liegen die Blätter eines vollständigen Binärbaums, der 32 Blätter hat (mit Begründung)? (*2 Punkte*)

Lösung:
$\log 32 = \log 2^5 = 5$.

f) Angenommen, Ihnen wird ein vollständiger Binärbaum gegeben mit der Garantie, dass die unter a) formulierte Bedingung zutrifft. Argumentieren Sie, dass es einen Algorithmus gibt, der die Frage beantwortet, ob eine Zahl im Baum enthalten ist, und dabei $l+1$ Vergleiche durchführt, wobei l das Level der Blätter ist. (*3 Punkte*)

Lösung:
Sie beginnen bei der Wurzel v (Level 0) und fragen, ob die gesuchte Zahl größer oder kleiner als die Zahl von v ist. Falls größer, folgen Sie der rechten Kante, anderenfalls der linken Kante. Dann stellen Sie mit dem neuen Knoten dieselbe Frage, usw., bis Sie den gesuchten Wert gefunden haben oder an einem Blatt angekommen sind, was spätestens nach $l+1$ Vergleichen der Fall ist. Dieses Vorgehen entspricht genau dem Vorgehen bei der Binärsuche in einer Liste.

g) (Bonus) Wie viele *Knoten* hat ein vollständiger Binärbaum, dessen Blätter auf Level l liegen (mit Begründung)? (*3 Punkte*)

Lösung:
$2^0 + 2^1 + \cdots + 2^l = \sum_{i=0}^{l} 2^i = 2^{l+1} - 1$.

Aufgabe 4 Werfen *(10 Punkte)*

Eine Münze hat zwei Seiten: Kopf (0) und Zahl (1). Ein normaler Würfel hat sechs Seiten: 1, 2, 3, 4, 5 und 6. Nehmen Sie an, dass Münze und Würfel immer exakt auf einer ihrer Seiten landen, wenn Sie sie werfen. Mehrere Münzwürfe (Würfelwürfe) hintereinander bilden eine Wurfsequenz, z.B. „0110“ („165433“).

a) Geben Sie alle möglichen Wurfsequenzen an, die Ihnen begegnen können, wenn Sie eine Münze werfen:

(1) einmal (*1 Punkt*)

Lösung:
„0“, „1“

(2) zweimal (*1 Punkt*)

Lösung:
„00“, „01“, „10“, „11“

(3) dreimal (*1 Punkt*)

Lösung:
„000“, „001“, „010“, „011“, „100“, „101“, „110“, „111“

b) Jeweils mit Begründung: Wie viele mögliche Wurfsequenzen gibt es, wenn Sie

(1) eine Münze n-mal werfen? (*1 Punkt*)

Lösung:
2^n

(2) einen normalen Würfel n-mal werfen? (*1 Punkt*)

Lösung:
6^n

(3) einen Würfel mit x Seiten n-mal werfen? (*1 Punkt*)

Lösung:
x^n. Allgemein ist x^n die Anzahl der Sequenzen der Länge n, die man aus x verschiedenen Elementen bilden kann, wenn die einzelnen Elemente mehrfach vorkommen können.

c) Jeweils mit Begründung: Wie hoch ist die Wahrscheinlichkeit, dass Sie

(1) die Wurfsequenz „111“ erhalten, wenn Sie eine Münze dreimal werfen? (*1 Punkt*)

Lösung:
$\frac{1}{8}$, denn es gibt 8 Wurfsequenzen der Länge 3 und eine davon ist „111“.

(2) eine Wurfsequenz mit mindestens zwei Einsen erhalten, wenn Sie eine Münze dreimal werfen? (*1 Punkt*)

Lösung:
$\frac{4}{8} = \frac{1}{2}$, denn von den 8 Wurfsequenzen der Länge 3 enthalten 4 mindestens zwei Einsen (s. Aufgabenteil a)(3)).

(3) eine Wurfsequenz erhalten, die mindestens eine 6 enthält, wenn Sie einen Würfel sechsmal werfen? (*2 Punkte*)
Hinweis: Sie können sich die Arbeit erleichtern, indem Sie sich zuerst die Wahrscheinlichkeit überlegen, in der gegebenen Situation eine Wurfsequenz zu erhalten, die keine 6 *enthält.*

Lösung:
Die Wahrscheinlichkeit, *keine* 6 zu werfen, ist bei jedem Wurf $\frac{5}{6}$. Die Würfe sind voneinander unabhängig, sodass die Wahrscheinlichkeit, sechsmal hintereinander *keine* 6 zu werfen, $(\frac{5}{6})^6$ ist. Die hier gesuchte *Gegenwahrscheinlichkeit* ist 1 minus die Wahrscheinlichkeit, sechsmal hintereinander *keine* 6 zu werfen: $1 - (\frac{5}{6})^6 \approx 1 - \frac{1}{3} = \frac{2}{3}$.

Ich habe für die Videos, die Nachbereitung und das Übungsblatt etwa ☐ Stunden gebraucht.
(Ann-Sophie fertigt aus diesen Zahlen eine Statistik an. Kurt und Corinna sehen nur diese Statistik. Wir möchten wissen, ob der Schwierigkeitsgrad in etwa richtig ist.)

Websuche war spannend ☐ okay ☐ langweilig ☐
schwierig ☐ okay ☐ einfach ☐