Algorithmen und Programmierung 3, WS 2003/2004 — 4. Übungsblatt

Abgabe bis Donnerstag, 20. November 2003

23. (15 Punkte) Eine hash-Funktion zum Speichern ungeordneter Paare.
Finden Sie n ganze Zahlen $t_1, \ldots, t_n$, sodass alle Summen $t_i + t_j$ für $1 \leq i < j \leq n$ verschieden sind und in einem möglichst kleinen Intervall $\{a, a+1, \ldots, b\}$ enthalten sind, für $n = 5, 6, 7, 8, 9, 10$.

24. (3 Punkte) Speichern ungeordneter Paare in einem Array.
    (a) (0 Punkte) Wie kann man die Lösung der vorigen Aufgabe dazu verwenden, für eine feste Liste von n Objekten die Paare, die man aus je zwei dieser Objekte bilden kann, möglichst kompakt zu verwalten und zu speichern (wie in einer Hash-Tabelle), sodass man in konstanter Zeit auf jedes Paar $\{u, v\}$ zugreifen kann, wenn man u und v kennt?
    (b) (3 Punkte) Wie kann man dieses Problem auf andere Art mit einem Array der (optimalen) Länge $n(n+1)/2$ lösen?

25. (0 Punkte) Bestimmen des doppelten Elementes.
In einem Array $a_0, a_1, \ldots, a_n$ sind ganzzahlige Werte zwischen 1 und n gespeichert, und zwar kommt jede Zahl mindestens einmal vor. Daraus folgt, dass es genau eine Zahl geben muss, die doppelt vorkommt. Schreiben Sie ein Programm, das diese Zahl bestimmt. Das Programm soll lineare Laufzeit haben, auf das Array a nur *lesend* zugreifen, und nur konstanten zusätzlichen Speicher benötigen.[1]

26. (0 Punkte) Lösen Sie Aufgabe 23 für *geordnete* Paare: Finden Sie $2n$ ganze Zahlen $s_1, \ldots, s_n$ und $t_1, \ldots, t_n$ sodass alle Summen $s_i + t_j$ für $1 \leq i, j \leq n$ verschieden sind und in einem möglichst kleinen Intervall $\{a, a+1, \ldots, b\}$ enthalten sind.

27. Zusatzaufgabe (3 Zusatzpunkte). Finden Sie eine Formel für t_i in Aufgabe 23, die das Problem für allgemeines n löst (nicht unbedingt optimal in dem Sinn, dass das enthaltende Intervall kleinstmöglich ist). Analysieren Sie die Länge des enthaltenden Intervalls bei Ihrer Methode. (Für die beste abgegebene Formel gibt es bis zu 10 weitere Zusatzpunkte.) (Die gleiche Frage kann man natürlich auch für die geordneten Paare aus Aufgabe 26 stellen.)

28. (5 Punkte) Entfernen Sie die Endrekursion aus der Methode `zugroß` im Programm zur Verwaltung einer Halde.[2]

```
void zugroß(int i)
// a[i] ist möglicherweise größer als seine Nachfolger.
{   int kleinsterNachfolger;
    if (2*i+1 <= n) {
        if (a[2*i]<a[2*i+1]) kleinsterNachfolger = 2*i;
        else                 kleinsterNachfolger = 2*i+1;
    }
    else if (2*i <= n) kleinsterNachfolger = 2*i;
    else return;
    if (a[i] > a[kleinsterNachfolger]) {
        vertausche(i, kleinsterNachfolger);
        zugroß(kleinsterNachfolger);
    }
}
```

[1] Diese Aufgabe dient nur zur Unterhaltung und hat mit dem Stoff der Vorlesung nichts zu tun. Wenn Sie das Rätsel aus einer anderen Quelle bereits kennen, dann würde mich sehr interessieren, woher.

[2] Siehe Aufgabe 21 und das Heapsort-Programm aus der Vorlesung:
`http://www.inf.fu-berlin.de/~rote/Lere/2003-04-WS/Algorithmen+Programmierung3/heapsort.java`