

Algorithmen und Datenstrukturen I

WS 2010/11, **3. Aufgabenblatt**, Abgabe 08.12.2010

Aufgabe 11

12 Punkte

Das folgende Programm `tau` nimmt Vertauschungen der Elemente in einem Array `L` der Länge `n` vor. Der Stack `S` sei bei Aufruf leer.

```
tau (int L[], int n) {
    int i,j;

    for (i=j=0; i<n; i++) {
        while ( (not EMPTY(S)) && (TOP(S)<L[i]) ) {
            L[j]=TOP(S); POP(S); j=j+1;
        }
        PUSH(S,L[i]);
    }

    while (j<n) {
        L[j]=TOP(S); POP(S); j++;
    }
}
```

- (a) Führen Sie einen Aufruf von `tau` mit `L[]=(1,5,4,3,2,6)` und $n = 6$ durch. Geben Sie jeweils nach Beendigung der inneren `while`-Schleife den Inhalt von `S` an. Geben Sie die Liste `L[]` nach Beendigung des Programms an. (4 Punkte)
- (b) Geben Sie eine möglichst kurze Liste an, die durch einmalige Bearbeitung mit `tau` nicht aufsteigend sortiert wird. (2 Punkte)
- (c) Begründen Sie, weshalb die Laufzeit von `tau` in $\Theta(n)$ liegt. Wie können Sie bereits anhand der Laufzeit erkennen, daß `tau` kein Sortieralgorithmus ist? (3 Punkte)
- (d) Nun werde nach Durchlauf von `tau` geprüft, ob `L[]` sortiert ist. Falls nicht, werde `tau` erneut aufgerufen (mit der zuvor zurückerhaltenen Liste). Zeigen Sie, dass diese Iteration terminiert: für jede anfängliche Liste `L[]` gibt es eine endliche Zahl von Aufrufen von `tau`, nach der die Liste sortiert ist. (3 Punkte)

Aufgabe 12

9 Punkte

Die Reisenden einer Teefahrt ins Wunderland haben allesamt an der Grenze ihren Reisepass abgegeben, um ein 7-minütiges Visum für das Wunderland zu erhalten. Die mit einem Pik-Buben abgestempelten Reisepässe liegen nun auf einem Stapel S bereit, um von ihren Besitzern wieder in Empfang genommen zu werden. Das weiße Kaninchen starrt angestrengt auf die Uhr und ist sicher, dass alles viel zu lang dauert und sie zu spät zur Teeparty kommen. Die Grinsekatzte schlägt vor, die Reisegäste sollen sich in einer Schlange Q anstellen. Der jeweils vorderste Reisende schaut nach, ob der oberste Pass des Stapels seiner ist. Falls ja, nimmt er den Pass vom Stapel, trinkt den Verkleinerungstrunk und verlässt die Schlange. Ansonsten bleibt der Stapel unverändert, und der Reisegast stellt sich hinten wieder an. Dies wird iteriert, bis entweder der Stapel oder die Schlange leer sind. Das Programm *chesirecat* sieht so aus

```
function chesirecat ( stack S, queue Q ) {  
  
    while(!EMPTY(Q) and !EMPTY(S)) {  
  
        if(FRONT(Q) == TOP(S)) {  
            POP(S);  
            drinkshrinkdrink(FRONT(Q));  
        } else {  
            ENQUEUE(Q, FRONT(Q));  
        }  
        DEQUEUE(Q);  
  
    }  
  
    END;  
  
}
```

- Welche notwendige und hinreichende Bedingung müssen S and Q beim Aufruf von *chesirecat* erfüllen, damit das Programm terminiert, also der Befehl `END` irgendwann erreicht wird? (2 Punkte)
- Betrachten Sie ab jetzt nur noch Instanzen, bei denen das Programm terminiert. Geben Sie eine exakte Schranke $\Theta(\dots)$ für die Laufzeit (*worst case*) von *chesirecat* als Funktion der Problemgröße n an. Hierbei sei $n = |S| + |Q|$ die Summe der Zahlen der Elemente in S und Q bei Aufruf. Begründen Sie Ihre Antwort. (3 Punkte)
- Nun sei bei Aufruf $Q = [Theo, Max, Paula, Annemarie, Otto]$. S enthalte dieselben Elemente wie Q . Geben Sie eine Startreihenfolge der Elemente in S an, so dass *chesirecat* (i) möglichst wenige und (ii) möglichst viele Schleifendurchläufe braucht. Geben Sie in beiden Fällen die Anzahl der Durchläufe an. (4 Punkte)

Aufgabe 13

8 Punkte

Verwenden Sie *Selection-Sort* und *Bubble-Sort* mit Abbruchkontrolle, um die folgende Liste zu sortieren: (4, 45, 23, 1, 24, 99, 4, 81). Geben Sie jeweils nach Beendigung der inneren Schleife die aktuelle Liste und die Anzahl der bis dahin stattgefundenen Schlüsselvergleiche an.

Aufgabe 14

6 Punkte

Die fürstliche Post des Fürstentums Binaco verwendet fünfstellige Postleitzahlen im Binärsystem zur Unterscheidung der 32 Zustellbezirke.

- (a) In der zentralen Poststelle werden täglich die n versandten Briefe per Fachverteilen (Distribution Sort) mit zwei Fächern nach Postleitzahlen sortiert. Die Zeitfaktoren für das Streuen und Sammeln sind dabei $C_1 = C_2 = 1$. Geben Sie den gesamten Zeitaufwand $T_{\text{Fach}}(n)$ an unter der Annahme, dass in jeder Binärstelle Duplikate auftreten.
- (b) Die Cousine eines stellvertretenden Staatssekretärs im Effizienzministerium schlägt die Umstellung auf ein modernes Sortierverfahren mit Zeitaufwand $T(n) = \lceil n \log_2 n \rceil$ vor. Berechnen Sie (Lösungsweg angeben), für welche Zahl an Briefen dieses Verfahren schneller als das Fachverteilen ist.