Algorithmen und Komplexität

Dr. Bruno Becker

Übungsblatt 1

1. Formulieren Sie einen rekursiven Algorithmus für n! (n Fakultät). Wie oft wir die Funktion für n >0 aufgerufen?

2. Gegeben sei ein Array a von n positiven ganzen Zahlen a[0],…,a[n-1] und eine Funktion g(x), die für eine positive ganze Zahl x den Wert 1 liefert, falls x gerade ist und 0 sonst. Analysieren Sie bitte die folgende Java Methode:

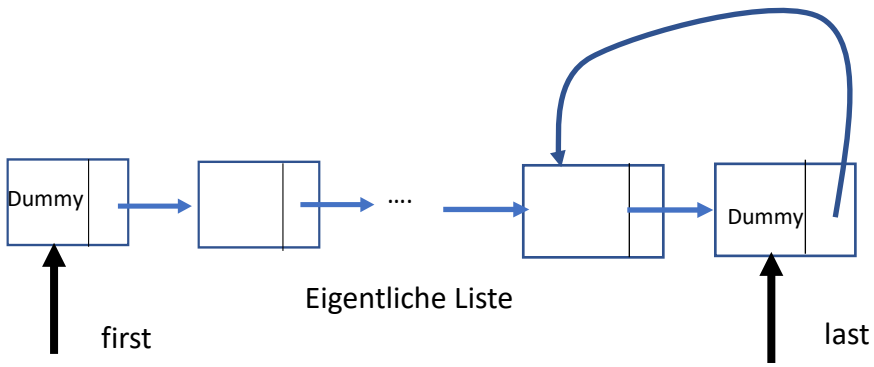
```
public static int gtest (int li, int re)
{
        if li > re return 0;
        else
        {
                int m = (li + re ) / 2;  // Ganzzahlige Division
                int gt = gtest (li,m-1) + g(a[m]) + gtest (m+1,r);
                return gt;
        }
}
```

   a) Welches Resultat liefert die Methode beim Aufruf gtest (0, n-1)?

   b) Wieviele Additionen werden beim Aufruf von gtest (li,re) benötigt? Geben Sie hierfür eine rekursive Formel in Abhängigkeit von re und li an.

   c) Formulieren Sie ein alternatives, iteratives Verfahren zur Berechnung von gtest.

3. Gegeben sei eine nicht leere verkettete lineare Liste *L* ganzer Zahlen mit ungerader Elementanzahl. Die Liste beginnt und endet mit einem bedeutungslosen „Dummy"-Element (d.h. ohne relevanten Wert). Zwischen den beiden Dummy-Elementen befinden sich die eigentlichen Listenelemente. Der Zeiger *first* zeigt auf das erste Dummy-Element, der Zeiger *last* auf das letzte Dummy-Element.



a) Schreiben Sie eine Methode *mitte* mit Inputparametern *first* und *last,* die das mittlere Element in der Liste entfernt.

b) Schreiben Sie eine Methode *teilen* mit Inputparametern *first, firsteven, firstodd,* die die Liste L mit Anfangszeiger *first* aufteilt in zwei (anfangs leere, also durch zwei Dummy-Elemente gegebene) Listen mit Anfangszeiger *firsteven* bzw. *firstodd.* In die eine Liste sollen die Elemente aus L mit geradzahligen Elementen gehängt werden, in die andere Liste die Elemente aus L mit ungeradzahligen Elementen. Aufrufe von **new** sind **nicht** erlaubt.

c) Schreiben Sie eine Methode *umdrehen* mit Inputparametern *first* und *last,* die die Reihenfolge der Elemente, d.h. die Zeiger in der Liste, umdreht. Keine zweite Liste erstellen, d.h. Aufrufe von **new** sind **nicht** erlaubt.

4. Gegeben sei eine doppelt verkettete nichtleere Liste. Schreiben Sie eine Methode, die das Maximum in der Liste (d.h. das Element mit dem größten Schlüsselwert) sucht, aus der Liste entfernt und den maximalen Schlüsselwert zurückgibt.