

25 Punkte Aufgabe 1 (Programmanalyse)

Gegeben sei das folgende Programmfragment:

```
function f2(n : integer) : integer;
var i, r : integer;
begin
  if n = 1 then
    return 1
  else
    begin
      r := 0;
      for i := 1 to n do
        begin
          r := r + f2(n div 2)
        end;
      return r
    end
end;
```

```
function f1(n : integer) : integer;
var c, r : integer;
begin
  c := n*n;
  r := 0;
  while c > 0 do
    begin
      r := r + f2(n);
      c := c-1;
    end;
  return r
end;
```

- 16 Punkte (a) Welchen Wert berechnet $f1(n)$? Begründen Sie Ihr Ergebnis formal. Gehen Sie davon aus, daß n eine Zweierpotenz ist.
- 9 Punkte (b) Welche Laufzeit hat $f1$ in Abhängigkeit von n ? Geben Sie das Ergebnis in O-Notation an und begründen Sie es. Ein formaler Beweis ist nicht notwendig.

Aufgabe 2 (2-3 Bäume)

10 Punkte

- (a) Fügen Sie die unten stehende Schlüsselfolge in einen 2-3 Baum ein. Skizzieren Sie den Baum nach jeder Strukturänderung und erläutern Sie welche Operationen ausgeführt werden.

5 Punkte

3, 4, 7, 5, 8, 1, 2, 6

- (b) Löschen Sie nacheinander die Elemente 8 und 5 und skizzieren Sie jeweils die dabei auftretenden Strukturänderungen.

5 Punkte

Aufgabe 3 (Hashing)

22 Punkte

- (a) Nennen Sie Vorteile und Nachteile von Hashverfahren. Erläutern Sie den Begriff Kollision und erklären Sie kurz die im Kurstext genannten Strategien zu deren Auflösung.

6 Punkte

- (b) Die im Kurstext vorgestellte Variante des quadratischen Sondierens garantiert, daß alle Behälter getroffen werden, falls die Anzahl m eine Primzahl der Form $4 \cdot j + 3$ ist. Wie müssen die Parameter m und b für geschlossenes Hashing bestimmt werden, damit der Speicherplatz optimal ausgenutzt wird?

7 Punkte

Beispiel: Die ersten Primzahlen der Form $4 \cdot j + 3$ sind 7, 11, 19, 23, 31, 43, ... Will man nun z.B. 33 Schlüssel speichern, so hat man bei der Wahl von $m = 43$ einen Verlust von 10 Elementen, während man bei der Wahl von $m = 11$ und $b = 3$ den Speicher optimal ausnutzt.

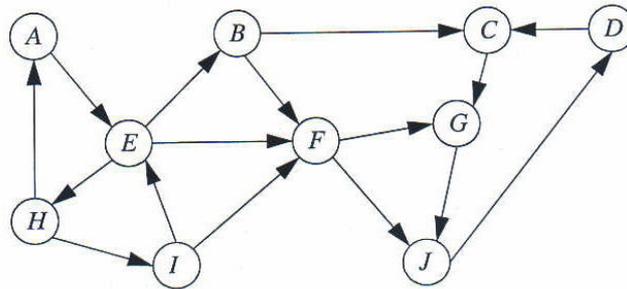
Beschreiben Sie allgemein wie zu gegebenen n Schlüsseln optimale Parameter m und b bestimmt werden können.

- (c) Fügen Sie die Zeichenketten $ABC, ABB, BAC, BAB, CAB, BBA, ACB, CBA, ABA, AAC, CAA, AAB, BAA, ABA$, für die die Summe der Buchstabenpositionen im Alphabet als Schlüssel verwendet werden soll, mittels der Kollisionsbehandlung aus Teil (b) in eine Hashtabelle mit $m = 7$ und $b = 2$ ein.

9 Punkte

10 Punkte Aufgabe 4 (Starke Komponenten)

Berechnen Sie mit Hilfe des im Kurs angegebenen Algorithmus die starken Komponenten des folgenden Graphen. Geben Sie das Ergebnis jedes Zwischenschritts des Algorithmus an. Stehen in einem Schritt mehrere Knoten zur Auswahl, gehen Sie in alphabetischer Reihenfolge vor.

**20 Punkte Aufgabe 5 (Geometrische Algorithmen)**

Im Kurstext wurde das Finden von Paaren sich schneidender Rechtecke in einer Rechteckmenge aufgeteilt in die Lösung des Punkteinschluß- und Segmentschnittproblems. Hier soll nun ein Planesweep-Algorithmus skizziert werden, der dieses Problem direkt löst. Stellen Sie dabei die Rechtecke in getrennter Darstellung durch ihre linken und rechten Enden dar. Spezifizieren Sie die Anforderungen an die Sweep-Status-Struktur, ohne eine konkrete Datenstruktur direkt anzugeben. Zudem brauchen Sie ein grundsätzliches Problem bei Rechteckschnittalgorithmen, nämlich die Vermeidung der doppelten Ausgabe von Schnitten, nicht zu lösen.

Beachten Sie: Ein Schnitt soll dann ausgegeben werden, wenn zwei Rechtecke gemeinsame Punkte haben.

13 Punkte Aufgabe 6 (Externes Sortieren)

Sortieren Sie die unten angegebene Folge mittels externen Sortierens. Hierbei soll *natürliches Mischen* verwendet werden, d.h. es sollen zu Beginn Läufe variabler Länge erzeugt werden. Der Hauptspeicher kann dabei 4 Elemente aufnehmen.

100, 10, 11, 20, 7, 23, 90, 87, 86, 43, 99, 32, 40, 41, 70, 31, 30, 17, 18