Aufgaben zur Veranstaltung
Algorithmen und Datenstrukturen, SS 2019

H. Pflug, J. Dietel FH Aachen, Campus Jülich; IT Center, RWTH Aachen

Präsenzaufgaben 14 24./25.06.2019

Aufgabe 1:

Nach dem gemeinsamen Besuch des Kurses *„Kochen für Geeks - ein alternativer Weg zur inneren Selbstfindung"*, sind Beavis und Butt-Head völlig fasziniert von der scharfen asiatischen Küche und beschließen den mutigen Selbstversuch. *„Erstmal wäre Sport wichtig, damit wir auch richtig hungrig sind!"* regt Beavis an. Butt-Head denkt kurz nach und hat dann eine zündende Idee, wie man Angenehmes mit Nützlichem kombinieren kann:

> *„Wir brauchen fünf Zutaten und haben fünf Märkte in unserer Stadt. Lass uns die Zutaten in unterschiedlichen Märkten kaufen, damit wir direkt alle kennenlernen. Die fünf Märkte joggen wir in der kürzest möglichen Reihenfolge ab, so dass wir zügig beginnen können und wenn wir zum Markt 'Dunkle Sonne' kommen, machen wir einen kurzen Abstecher zu Paolas pinker Powerbude zum Fitness-Training, die ist schließlich im gleichen Gebäude. Da können wir uns richtig auspumpen, Kalorien verbrennen und so endgültig den nötigen Hunger bekommen - ist ja auch nur ein läppischer Umweg von 20 Metern!".*
>
> *„Butt-Head, Du bist genial - aber wie finden wir jetzt die richtige Reihenfolge für die Märkte, so dass die gesamte Laufstrecke minimal wird?"*.

Helfen Sie Beavis und Butt-Head! Bekannt sind die folgenden Entfernungen (Meter) zwischen Beavis und Butt-Heads Studentenwohnheim (BB), sowie den Märkten Roter Drache (RD), Goldener Stern (GS), Süßer Wind (SW), Dunkle Sonne (DS) und Rote Laterne (RL):

M	BB	RD	GS	SW	DS	RL
BB		90	80	220	60	120
RD	90		70	170	45	80
GS	80	70		200	40	45
SW	220	170	200		170	230
DS	60	45	40	170		85
RL	120	80	45	230	85	

(a) Erläutern Sie kurz, wie man dieses Problem mit einem Graphen modelliert (Sie müssen keinen Graph zeichnen!). Überlegen Sie sich insbesondere, wie Sie Paolas pinke Powerbude berücksichtigen wollen. Begründen Sie dann kurz, warum das Durchprobieren aller möglichen Lösungen für den allgemeinen Fall (n - 1 Märkte und ein Wohnheim, also n Stationen) zu aufwändig ist.

(b) Beschreiben Sie eine Lösung, die sich höchstens um den Faktor 2 von der optimalen Lösung unterscheiden darf. Verwenden Sie dazu ein aus der Vorlesung bekanntes Verfahren und erläutern Sie dieses.

Aufgabe 2:

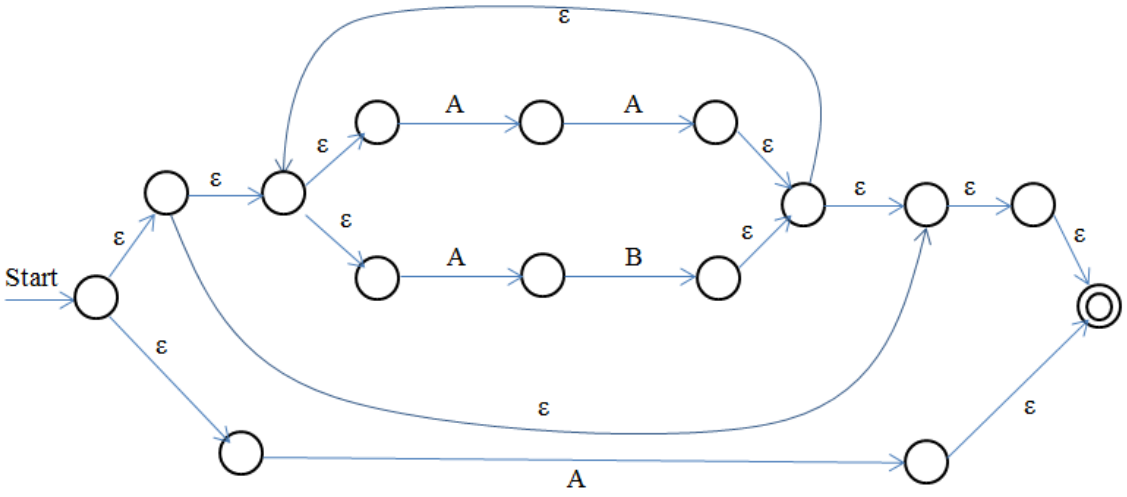
Gegeben sei die Rekursionsgleichung

$$T(n) = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ 7 \cdot T(n-1) - 12 \cdot T(n-2) & \text{sonst} \end{cases}$$

Benutzen Sie die Technik der Erzeugenden Funktion um eine geschlossene Form für *T(n)* zu finden.

Aufgabe 3:

a) Gegeben sei der folgende nichtdeterministische endliche Automat. Stellen Sie den zugehörigen regulären Ausdruck auf.



Aufgabe 4:

Sortieren Sie das folgende Feld mit Quick-Sort in aufsteigende Reihenfolge. Dabei sollte als Pivot-Element immer das *am weitesten links* stehende Element gewählt werden. Teilfelder von 2 Elementen brauchen Sie nicht mehr mit Quick-Sort sortieren - hier dürfen Sie die beiden Elemente gegebenenfalls einfach tauschen.

3	8	1	6	2	4	9	5

Aufgabe 5:

Der folgende Algorithmus hat die Aufgabe, ein Feld in aufsteigender Reihenfolge zu sortieren:

```
public static void sort(int[] num) {
     int j;

     int key;
     int i;

     for (j = 1; j < num.length; j++) {
           key = num[j];
           for(i = j - 1; num[i] > key; i--) {
                 num[i+1] = num[i];
           }
           num[i+1] = key;

           //Ausgabe

     }
}
```

a) Testen Sie den Algorithmus mit der folgenden Zahlenfolge. Geben Sie die Zahlenfolgen an, die das Feld jeweils beim Erreichen der Kommentarzeile „Ausgabe“ hat. Notieren Sie eventuelle Probleme.

3	8	7	5	4	1

b) Um welchen Algorithmus handelt es sich (näherungsweise)?

c) Wie müssen Sie den Algorithmus ändern, damit er korrekt arbeitet?

Aufgabe 6:

Schreiben Sie eine Klasse `Wortsuche`. Fügen Sie der Klasse die unten beschriebenen Methoden hinzu, die es ermöglichen, in dem String nach einem Teilstring mit dem in der Vorlesung (Vorlesungsfolien ab Folie 469 im ILIAS) behandelten Boyer-Moore-Verfahren zu suchen.

```
public class Wortsuche {
   private String text;    //der Gesamtstring

   //Setzt den Text, in dem gesucht wird.
   public Wortsuche(String text)

   //Gibt die last-Tabelle fuer den String pattern als HashMap
   //zurueck. Nicht vorhandene Zeichen haben keinen Eintrag.
   private static HashMap<Character,Integer> getLastTable(String pattern)

   //Ueberprueft, ob der String pattern mit dem kompletten String ab
   //der Position pos in der benötigten Länge uebereinstimmt.
   private boolean fits(String pattern, int pos)

  //Sucht den String pattern nach dem Boyer-Moore-Sunday Verfahren
  //und gibt die Index-Position des 1. Vorkommens zurueck. Kommt der
  //String nicht vor, wird -1 zurueckgegeben.
   public int findFirst(String pattern)
}
```

Testen Sie:

```
public static void main(String[] args) {

      Wortsuche meinWort = new Wortsuche("DieguteBabanane123Lecker");
      int ergebnis=meinWort.findFirst("banane123");
      if (ergebnis==-1){
            System.out.println("Pattern wurde nicht gefunden.");
      }else{
            System.out.println("Gefunden an Position "+ergebnis);
      }
}
```