Claus/Lewandowski Sommersemester 2000 Aufgabenblatt 2

Übungen zur Vorlesung „Entwurf und Analyse eff. Algorithmen“

---

1. Der Dijkstra-Algorithmus berechnet in einem Graphen $G = (V, E)$ kürzeste Wege von einem Startknoten s zu allen anderen Knoten. In vielen Anwendungen ist nicht nur der kürzeste Weg interessant. Entwerfen sie einen Algorithmus, der in einem Graphen $G = (V, E)$ den zweitkürzesten Weg zwischen einem Startknoten s und einem Zielknoten z berechnet. Geben Sie die Zeitkomplexität ihres Algorithmus an.

Der berechnete Weg sollte nach Möglichkeit keine Schleifen enthalten. Kann ihr Algorithmus dies garantieren?

2. Eine Knotenüberdeckung eines ungerichteten Graphen $G = (V, E)$ ist eine Teilmenge $K \subseteq V$ so, dass jede Kante $e \in E$ mindestens ein Ende in K hat, d.h. $e \cap K \neq \emptyset$. Eine Knotenüberdeckung ist optimal (oder minimal), falls es keine Überdeckung mit weniger Knoten gibt.

Gegeben seien die folgenden Greedy-Strategien zur Berechnung einer optimalen Knotenüberdeckung. (Der Grad eines Knoten ist die Anzahl der mit ihm verbundenen Kanten).

a) Suche Knoten x mit maximalem Grad, entferne den Knoten x in G und nehme ihn in K auf. Entferne die mit x verbundenen Kanten $\{x, y\} \in E$. Iteriere dieses Vorgehen, bis keine Kante mehr vorhanden ist.

b) Wähle eine Kante $\{x, y\} \in E$. Nehme x und y in K auf und entferne alle mit x und y verbundenen Kanten $\{x, z\}, \{y, z'\} \in E$. Iteriere dieses Vorgehen, bis keine Kante mehr vorhanden ist.

Die Güte einer Überdeckung ist die Zahl $\frac{|K|-o}{|K|}$, wobei o die Kardinalität der optimalen Überdeckung ist. Zeigen sie, dass die zweite Strategie mindestens die Güte $1/2$ besitzt. Besitzt die erste Strategie eine bessere Güte?

3. An einem Ufer befinden sich n Boote $B_1, ..., B_n$, die an das andere Ufer gebracht werden sollen. Zum Übersetzen benötigt das Boot B_i t_i Minuten. Wie lange brauchen 2 Personen, um alle n Boote an das andere Ufer zu bringen?

Entwerfen Sie einen Algorithmus, der dieses Problem für n Boote mit den Übersetzdauern t_i, $1 \leq i \leq n$ löst. Geben Sie die Zeitkomplexität des Algorithmus an. Geben Sie die Lösung für $n = 4$ und $t_1 = 1$, $t_2 = 2$, $t_3 = 4$ und $t_4 = 5$ an.

4. Folgende beiden Programme berechnen $f(x) = \sqrt{|x|}$:

```
program squareroot1;
  var a, b, eps, h, z: real;
begin
  read (z); read (eps);
  if z<0 then z:=-z fi;
  if z<1 then a:=0; b:=0.5
  else a:=1;
    while 4*a*a<=z do a:=a+a od;
    b:=a/2
  fi;
  h:=z-a*a;
  while (h>eps) and (b>0) do
    if h>=(2*a+b)*b then a:=a+b; h:=z-a*a fi;
    b:=b/2
  od;
  write('Wurzel von ',z,' = ',a)
end.
```

und

```
program squareroot2;
  var a, b, eps, h, z: real;
begin
  read (z); read (eps);
  if z<0 then z:=-z fi;
  a:=z;
  h:=a*a-z;
  while (|h|>eps) do
    a:=(a+z/a)/2; h:=z-a*a
  od;
  write('Wurzel von ',z,' = ',a)
end.
```

Untersuchen Sie empirisch, mit welchem Aufwand sich bei der Wurzelberechnung eine Genauigkeit auf k Dezimalstellen erreichen lässt. Versuchen Sie, ihre Ergebnisse zu beweisen.

Abgabe: Montag, 29. Mai, nach der Vorlesung (Box im Foyer 1. Stock)