

10. Übungsaufgaben zur LV Algorithmen & Datenstrukturen

Abgabetermin: Mi, 02.06.04

Aufgabe:

- Vervollständigung der Methode *kuerzesteWege()* aus der Klasse *eNetz*:

```
//...
/* schauen, ob Fehler durch neg. BgLängen entstanden sind und versuchen zu *
 * korrigieren; funktioniert nicht bei negativen Kreisen! */
double maxDt; int minBg;
for (k=1; k<=m; k++) {
    if (getY(k) - c(k) > eps) {
        setzWrz (AKn(k));
        if (StzRchtg (EKn(k)) < 0) return false;
        else {
            do {
                markCZ (getStzBg (EKn(k)));
                maxDt = getY(k) - c(k); minBg = k;
                for (j=1; j<=m; j++)
                    if (czKoeff(j) == -1 && maxDt > c(j) - getY(j))
                        { maxDt = c(j) - getY(j); minBg = j; }
                for (j=1; j<=n; j++)
                    if (getMarke(j) == 2) setT(j, getT(j) - maxDt);
                setStzBg (EKn(minBg), minBg);
            } while (minBg != k);
        }
        setzWrz (Start);
    }
}
return true;
}
```

- Klasse *Dialog_eNetz* (auszugsweise):

```
public class Dialog_eNetz extends JDialog {
    int spAnz=3, knAnz=8, bgAnz=13;
    String[][] Tab = new String[knAnz][spAnz];
    String[] TabKopf = new String[] {
        "Knotenname", "Weglänge", "erreicht über Knoten"};
    JTable jt = new JTable(Tab, TabKopf);
    JOptionPane jOptPan = new JOptionPane();
    //...

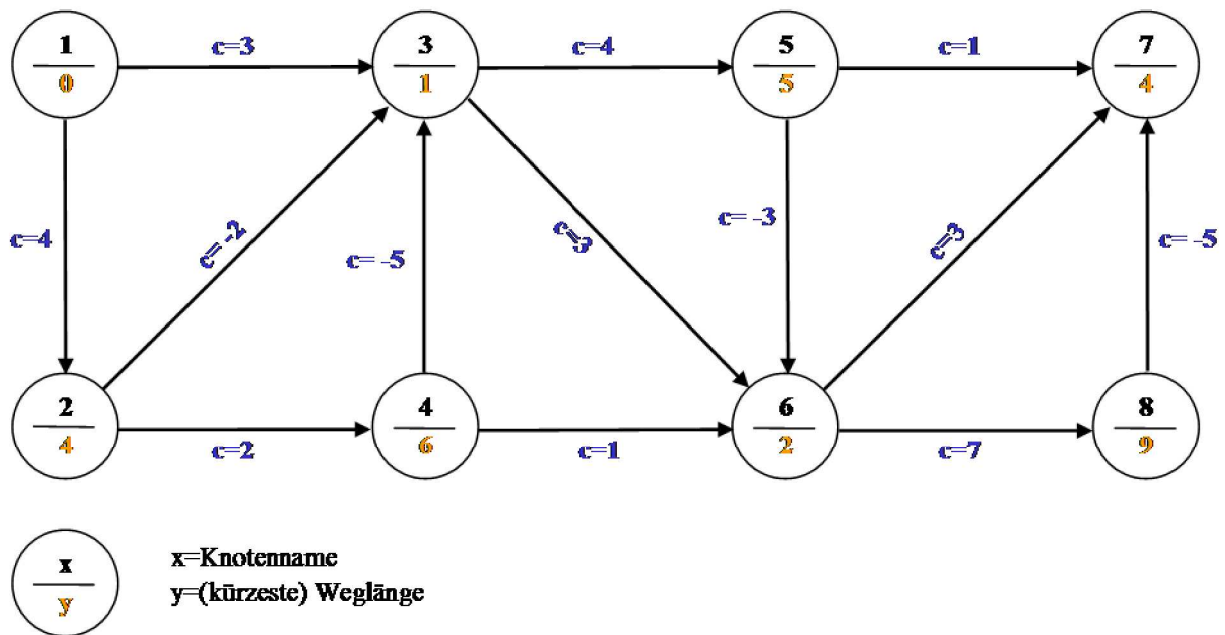
    void zeigen(int was) {
        int startKn=1;
        boolean ok;
        eNetz ntz = new eNetz(knAnz, bgAnz);
        //##### Testnetz aufbauen #####
        ntz.insertBg("1", "2", "4"); ntz.insertBg("1", "3", "3");
        ntz.insertBg("2", "3", "-2"); ntz.insertBg("2", "4", "2");
        ntz.insertBg("4", "3", "-5"); ntz.insertBg("3", "5", "4");
        ntz.insertBg("3", "6", "3"); ntz.insertBg("4", "6", "1");
        ntz.insertBg("5", "6", "-3"); ntz.insertBg("5", "7", "1");
        ntz.insertBg("6", "7", "3"); ntz.insertBg("6", "8", "7");
        ntz.insertBg("8", "7", "-5");
    }
}
```

```

//##### Weglänge (kürzeste/längste) ermitteln #####
switch(was) {
  case -1:
    if(ntz.kuerzesteWege(startKn)==false)
      jOptPan.showMessageDialog(this,
        "Es sind Kreise negativer Länge im Graphen vorhanden!");
    break;
  case 1:
    if(ntz.laengsteWege(startKn)==false)
      jOptPan.showMessageDialog(this,
        "Es sind Kreise positiver Länge im Graphen vorhanden!");
    break;
  default: return;
}
//##### Ausgabe in Tabelle #####
for(int i=1;i<=ntz.n;i++){
  jT.setValueAt(ntz.KnName(i),i-1,0);
  jT.setValueAt(String.valueOf(ntz.getT(i)),i-1,1);
  jT.setValueAt(ntz.KnName(ntz.AKn(ntz.getStzBg(i))),i-1,2);
}
this.show();
}
//....

```

verwendetes eNetz - Beispiel:



• Ausgabe:

Knotenname	Weglänge	erreicht über Knoten
1	0.0	
2	4.0	1
3	1.0	4
4	6.0	2
5	5.0	3
6	2.0	5
7	4.0	8
8	9.0	6

Beenden