

Praktikum 13

Lühimad teed graafis

Ülesanded

- GTE_3-1. Graafi laiuti läbimine
- GTE_3-2. Dijkstra meetod

Iseseisev töö nr 7

Ülesanded

GTE_3-1. Graafi laiuti läbimine

Kirjutada Java-programm, milles on meetodina realiseeritud graafis otseteede leidmise algoritm [Õpik, lk 107, joonis 6.12] ja peameetodis toimub selle meetodi test-rakendamine.

```
Amadeus - workspace2017s\GTE_3-1.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

otsetee
void läbida_laiuti(Graaf g, Tipp a)»
»
» [Õpik, lk 106-107, joonis 6.12] Välja .vaadeldud ei kasutata:
» tipp on vaadeldud, kui talle on seatud väli .eellane
» Antud: graaf g = (V, E) ja lähtetipp a,
» . milled alustada g tippude läbimist;
» Tulemus: on salvestatud tipuväljadesse .eellane järgmiselt:
» . iga tipu v (!=a) korral, mis on saavutatav tipust a
» . v.eellane on v-le eelnev tipp (tipu number) otseteel a..v;
a.seadaVäli("eellane", "0"); st vaadeldud
Vector<Tipp> Q = new Vector<Tipp>();»
Q.add(a);»

while (Q.size() != 0)»
»
» Tipp v = Q.remove(0); võtta Q esimene
»
» for (Kaar kaar : g.väljuvad(v))»
»
» Tipp w = kaar.löpp();»
» kaar on v-->w
»
» if (w.väli("eellane") == null) kui veel saavutamata
»
» w.seadaVäli("eellane", "" + (g.indeks(v) + 1));»
» Q.add(w);»

a.eemaldadaVäli("eellane"); lähtetipul eellast ei ole
otsetee
```

```
Amadeus - workspace2017s\GTE_3-1.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

import ee.ut.kiho.aa.graaf.*;»
import java.util.Vector;»
»
» Kasutatakse: TeeInfo.class
»
» TEST
»
» vaadeldav graaf g
»
» String fNimi = "graafTeed.txt";»
»
» println("Graaf failist " + fNimi);»
» new GraafiJoonistaja(fNimi);»
» Graaf g = Graaf.sisestadaTabelist(fNimi);»
»
» vaadeldav graaf g
»
» lähtetipu ja sihttipu numbrid a_nr ja b_nr
» Tipp lähtetipp = g.tipp(a_nr-1);»
» Tipp sihttipp = g.tipp(b_nr-1);»
»
»
» läbida_laiuti(g, lähtetipp);»
»
»
» g.väljastadaTabelina("x"+fNimi);»
» new GraafiJoonistaja("x"+fNimi);»
»
» väljastada leitud tee ja selle pikkus
»
» TeeInfo tInf = new TeeInfo(g, sihttipp);»
»
»
» println("Leitud lühim tee (otsetee): ");»
» println(tInf.tipud_teel());»
» println(tInf.tipud_teel.size() + " tippu");
»
» väljastada leitud tee ja selle pikkus
»
» TEST
```

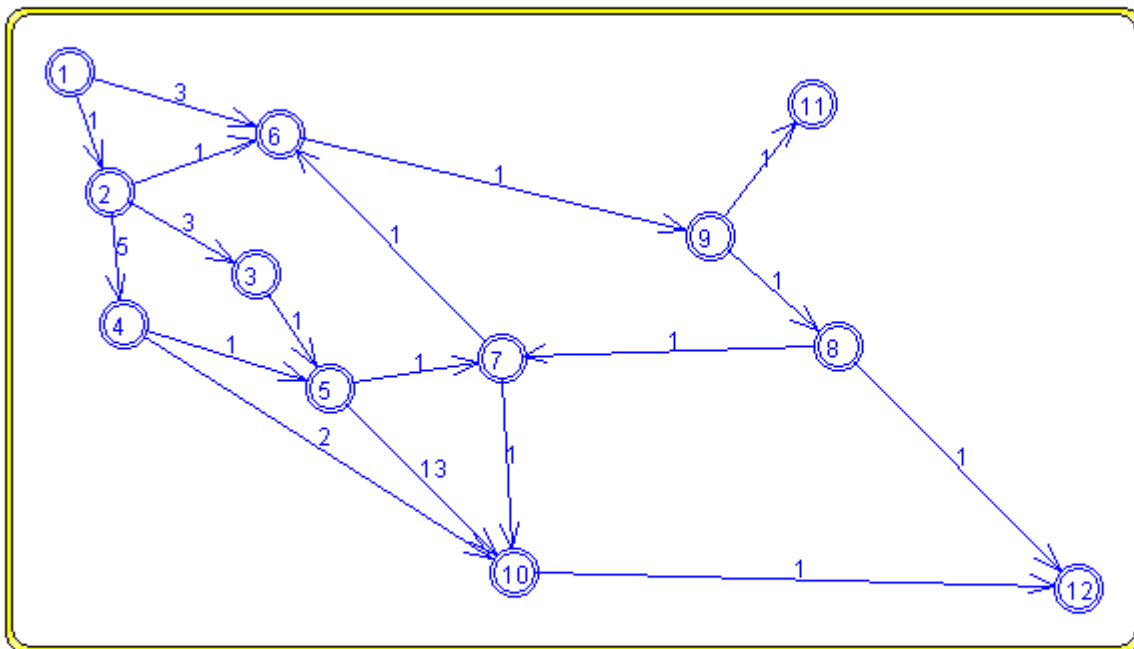
GTE_3-2. Dijkstra meetod

Kirjutada Java-programm, milles on meetodina realiseeritud graafis lühimate teede leidmise algoritm [Õpik, lk 100, joonis 6.6] ja peameetodis toimub selle meetodi test-rakendamine.

Testimisel on soovitatav kasutada näitegraafe *graafTeed.txt* ja *graafTeedSuur.txt*

Programmi teksti lühendamiseks on otstarbekohane kasutada ka meetodit

```
static double d(Tipp v){
    // Antud: tipp v
    // Tulemus: tagastatakse selle tipu väljal "d" olev arv (algoritmides v.d)
    return Double.parseDouble(v.väli("d"));
} //d
```



TABEL

- 1) 1-->2 3-->6
- 2) 3-->3 5-->4 1-->6
- 3) 1-->5
- 4) 1-->5 2-->10
- 5) 13-->10 1-->7
- 6) 1-->9
- 7) 1-->6 1-->10
- 8) 1-->7 1-->12
- 9) 1-->11 1-->8
- 10) 1-->12
- 11)
- 12)

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.

TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga. Eemaldamine: parem klõps valitud tipul (2 x parem klõps).

KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida algustipp, seejärel lõpptipp.

GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

FAIL: C:\Users\kiho\AlgPython\workspace2014\graafTeed.txt Värskendamine: pidev.

Graaf failist graafTeed.txt
 Lähtetipu nr: 1, sihttipu nr: 12
 Dijkstra leitud lühim tee on
 [1, 2, 6, 9, 8, 12]
 pikkusega 5.0

preambul

```
import ee.ut.kiho.aa.graaf.*;»
import java.util.Vector;»
//import java.util.Collections; list Q
import java.util.PriorityQueue; kuhi Q
```

```
»»
```

```
» Kasutatakse: TeeInfo.class
```

```
»»
```

TEST |

```
»
```

```
» vaeldav graaf g
```

```
» lähtetipu ja sihttipu numbrid a_nr ja b_nr
```

```
Tipp lähtetipp = g.tipp(a_nr-1);»
```

```
Tipp sihttip = g.tipp(b_nr-1);»
```

```
»»
```

```
läbida_Dijkstra(g, lähtetipp);»
```

```
»»
```

```
g.väljastadaTabelina("x"+fNimi);»
```

```
new GraafiJoonistaja("x"+fNimi);»
```

```
»»
```

```
väljastada leitud tee ja selle pikkus
```

```
»
```

```
TeeInfo tInf = new TeeInfo(g, sihttip);»
```

```
»»
```

```
println("Dijkstra leitud lühim tee: ");»
```

```
println(tInf.tipud_teel());»
```

```
println("pikkusega "+ tInf.pikkus());»
```

```
//println(tInf.tipud_teel.size() + " tippu");
```

```
väljastada leitud tee ja selle pikkus
```

```
TEST
```

double d(Tipp t)

void läbida_Dijkstra(Graaf g, Tipp a) [Õpik, lk 100]

- » Antud: orienteeritud graaf $G = (V, E)$ ja lähtetipp a ,
- » . millest arvestada teiste tippude kaugusi;
- » . igal kaarel (v, w) on pikkus $c(v, w) = \text{kaar.nimi} \geq 0$
- » . NB! Kui kaare (arvulist) pikkust ei ole, loetakse see = 1.0
- » Tulemus: on salvestatud tipuväljadesse $.d$ ja $.eellane$
- » . järgmiselt:
- » . iga tipu v korral on $v.d$ lühima tee $a \dots v$ pikkus ja
- » . $v.eellane$ on v -le eelnev tipp (tipu number) sellel teel;
- » . kui tipp v ei ole algustipust saavutatav, siis $v.d = +\text{lõpmatus}$
- » . ja väli $v.eellane$ puudub; ka $a.eellane$ puudub

double lõpm = Double.MAX_VALUE; +lõpmatus

kõigile tippudele $d \leftarrow +\text{lõpmatus}$

$a.seadaVäli("d", "0");$ lähtetipu kaugus iseendast

» abiks tippude eelistusjärjekord (front) Q , milles kirje võtmeks $d(\text{tipp})$:

$//\text{Vector}<\text{Tipp}> Q = \text{new Vector}<\text{Tipp}>();$ »

PriorityQueue<Tipp> $Q =$
 $\text{new PriorityQueue}<\text{Tipp}>(10, (t1, t2) \rightarrow (\text{int})(d(t1) - d(t2)));$ »

$//Q.add(a);$ lähtetipp fronti

$Q.offer(a);$ lähtetipp fronti

while ($Q.size() \neq 0$)

while ($Q.size() \neq 0$)»

» võtta frondist Q vähima d väärtusega tipp v :

$//\text{Tipp } v = \text{Collections.min}(Q, (t1, t2) \rightarrow (\text{int})(d(t1) - d(t2)));$ $Q.remove(v);$ »

Tipp $v = Q.poll();$ »

» frondis tipp v oli vähima d väärtusega

for (Tipp $w : g.naabrid(v)$)»

tipu v iga naabri w korral:

Kaar $kaar = g.kaar(v, w);$ $kaar$ on $v \rightarrow w$

if ($d(w) == \text{lõpm}$) kui veel saavutamata $\text{Math.abs}(d(w) - \text{inf}) < 0.0001$

$//Q.add(w);$ tipp w panna fronti

$Q.offer(w);$ tipp w panna fronti

double $kaar_v_w_pikkus;$ $kaar_v \rightarrow w$ pikkus

» $kaar_v_w_pikkus :=$ kas arv väljalt $kaar.nimi()$ või 1.0

» üks (võimalik) tee algustipust a tippu w on läbi tipu v , $a \rightarrow \dots \rightarrow v \rightarrow w$,

» . pikkusga $d(v) + kaar_v_w_pikkus$

if ($d(v) + kaar_v_w_pikkus < d(w)$)»

kui võimalik tee $a \rightarrow \dots \rightarrow w$ on lühem kui senine (pikkusega $d(w)$)

» Märkus. Tingimus saab kehtida ainult tipu w korral,

» . mis juba varem oli frondis Q , st mitte äsja lisatud w korral

» . ega ka frondist Q juba eemaldatu korral

» ümber seada tippu w väljad "eellane" ja "d"

Iseseisev töö nr 7

Esitamise tähtaeg rühmal inf1(J. Liivi 2-202, E 12 -14): **9. detsember 2017, kell 12.15**
Esitamise tähtaeg rühmal inf7(J. Liivi 2-202, T 14 -16): **10. detsember 2017, kell 14.15**