

Praktikum 12

Teede otsimine graafi sügavuti läbides

- **Harjutusülesanded**

GTE_2-0. Läbida graaf sügavuti, seades välju „eellane“

GTE_2-1. Printida üks tee graafi kahe tipu vahel

GTE_2-2. Printida kõik teed graafi kahe tipu vahel

GTE_2-3. Salvestada kõik teed graafi kahe tipu vahel

Harjutusülesanded

GTE_2-0. Läbida graaf sügavuti, seades välja „eellane“

Kirjutada ja testida rekursiivne Java-meetod *läbida_sügavuti*.

```
Amadeus - workspace2017s\GTE_2-0.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

import ee.ut.kiho.aa.graaf.*;»
TEST
void läbida_sügavuti(Graaf g, Tipp t)»
    spetsifikatsioon
    »
    » Antud: graaf g ja selle tipp t
    » Tulemus: g läbitud lähtudes tiput t;
    » . g tippudele on lisatud välja "eellane" ja "vaadeldud";
    » . x.väli("eellane") =
    » . <selles tipp number, kust läbimise käigus tippu x tuldi>
    spetsifikatsioon
    t.seadaVäli("vaadeldud", "x"); tippule t panna lipp "vaadeldud"
    for (Tipp v : g.naabrid(t)) // minna edasi naabritest»
        if (v.väli("vaadeldud") == null) t naaber v veel väisamata
            v.seadaVäli("eellane", "" + (g.indeks(t)+1)); v eellaseks tippu t nr
            läbida_sügavuti(g, v);»
```

```
Amadeus - workspace2017s\GTE_2-0.algja...
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

TEST

Sisestada graaf g:
String fNimi = "gTest11.txt";»
//fNimi = "graafTeed.txt";»
new GraafiJoonistaja(fNimi);»
Graaf g = Graaf.sisestadaTabelist(fNimi);»

int aNr = 3; lähtetipu number
Tipp a = g.tipp(aNr - 1); lähtetipp
println("Läbitakse graaf failist " + fNimi +
        " lähtudes tipust nr " + aNr);

läbida_sügavuti(g, a);»

kuvada tulemus:
fNimi = "ajutine.txt";»
g.väljastadaTabelina(fNimi);»
new GraafiJoonistaja(fNimi);»

TEST
```

GTE_2-1. Printida üks tee graafi kahe tipu vahel

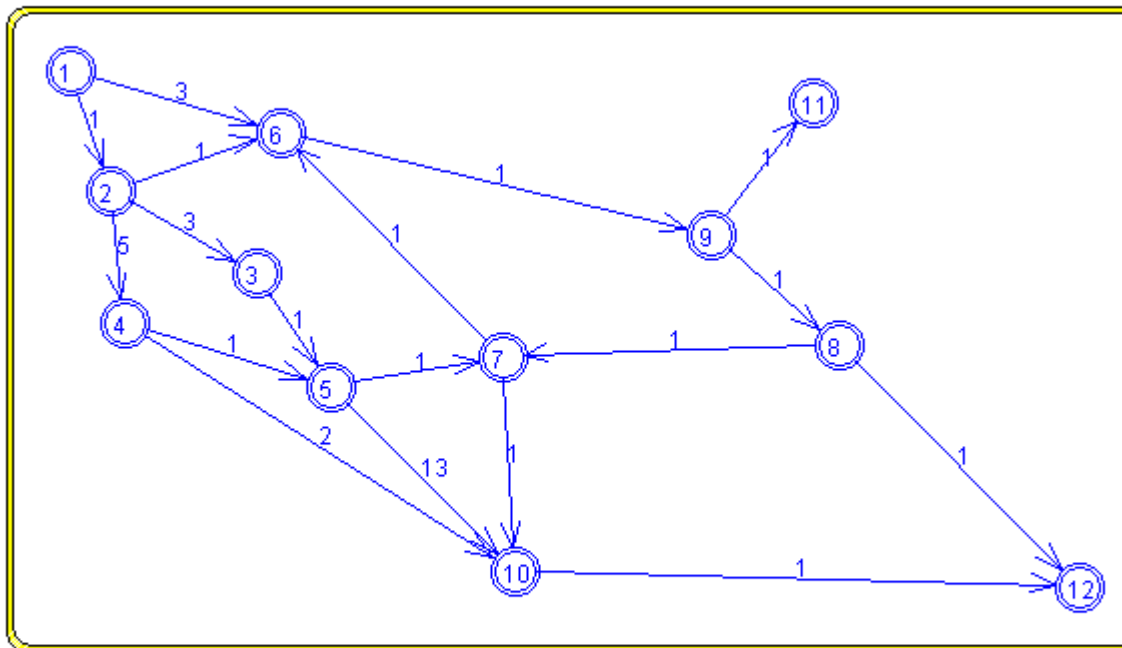
Kirjutada ja testida rekursiivne Java-meetod *otsida_tee*.

Testi tulemuse näide (*graafTeed.txt* korral):

Tee tipust 1 tippu 12:

Tee: [1, 2, 3, 5, 10, 12]. Kogupikkus: 19

C:_ISIKLIK\ownCloud\AlgJava\graafTeed.txt



MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.

TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga. Eemaldamine: parem

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

GRAAF ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

FAIL: C:_ISIKLIK\ownCloud\AlgJava\gxxx.txt Värskendamine: pidev.

Amadeus - workspace2017s\GTE_2-1.algjava

Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

projekt: AlgJavaWorks2017oCDB

```
import ee.ut.kiho.aa.graaf.*;>
```

```
import java.util.Vector;>
```

TEST

```
void otsida_tee(Graaf g, Tipp t, Tipp b)>>
```

spetsifikatsioon

>>

>> Antud: graaf g ja selle tipud t ning b

>> Tulemus: g läbitud lähtudes tiput t;

>> . g tippudele on lisatud välju "eellane" ja "vaadeldud";

>> . kui leidub tee tipust t tippu b, siis

>> . see tee on määratud tipuväljadega "eellane"

>> . (x.väli("eellane") = <selle tipu number, kust tee tuleb tippu x>);

>> . printitakse leitud tee (tagurpidi <b, b.eellane, ..., eellaseta tipp>)

>> . ja selle kogupikkus

spetsifikatsioon

t.seadaVäli("vaadeldud", "x"); tippule t panna lipp "vaadeldud"

>>

```
if (t == b) // kui oleme sihil>>
```

```
printida (tippunumbrid): tagurpidi t, t.eellane, ...
```

>>

```
else>>
```

```
for (Tipp v : g.naabrid(t)) // minna edasi naabritest
```

```
Amadeus - workspace2017s\GTE_2-1.algjava
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWork

if (t == b) // kui oleme sihil»
»
» printida (tippunumbrid): tagurpidi t, t.eellane, ...
double sum = 0.0; leitava tee kogupikkus
Vector<Integer> tee = new Vector<Integer>(); teel olevate tippude numbrid
Tipp tt = b; viimane teel, tt -- jooksev, tagasi mööda eellasi
* while(tt.väli("eellane") != null)»
»
» tee.add(0, g.indeks(tt)+1); teele lisada tipu tt number
Tipp t0 = tt; // kaare lõpptipp;»
tt = g.tipp(Integer.parseInt(tt.väli("eellane")) - 1); tt := tt.eellane
Kaar k = g.kaar(tt, t0);»
sum += Double.parseDouble(k.nimi()); kogupikkusele lisada kaare tt-->t0 pikkus

if (sum > 0) kui tee ei olnud tühi (tipul b oli eellane)
»
» tee.add(0, g.indeks(tt)+1); teele lisada veel tipu tt number (esimene teel)

println("Tee: " + tee + ". Kogupikkus: " + sum);»
```

```
Amadeus - workspace2017s\GTE_2-1.algjava
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

for (Tipp v : g.naabrid(t)) // minna edasi naabritest»
»
» if (v.väli("vaadeldud") == null) t naaber v veel väisamata
»
» v.seadaVäli("eellane", "" + (g.indeks(t)+1)); v eellaseks tipu t nr
» otsida_tee(g, v, b);»
```

```
Amadeus - workspace2017s\GTE_2-1.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

TEST
»
» sisestada graaf g
println("\nGraaf failist " + fNimi + ".");»
» määrata tipud a ja b
»
» määrata algus- ja sihttipu numbrid (aNr ja bNr)
Tipp a = g.tipp(aNr - 1); lähtetipp
Tipp b = g.tipp(bNr - 1); sihttipp
» määrata tipud a ja b
println("\nTee tipust " + aNr + " tippu " + bNr + ":");
otsida_tee(g, a, b);

TEST
```

GTE_2-1a. Printida üks tee graafi kahe tipu vahel pärast läbimist

Kirjutada ja testida vastav meetod *otsida_tee*.

```
Amadeus - workspace2017s\GTE_2-1a.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

void otsida_tee(Graaf g, Tipp a, Tipp b)»
»
» Antud: graaf g ja selle tipud a ning b
» Tulemus: leitud üks tee a..b, printitud koos kogupikkusega
  läbida_sügavuti(g, a); GTE_2-0|
» printida (tippnumbrid): tagurpidi t, t.eellane, ...
  double sum = 0.0; leitava tee kogupikkus
  Vector<Integer> tee = new Vector<Integer>(); teel olevate tippude nu
  Tipp tt = b; viimane teel, tt -- jooksev, tagasi mööda eellasi

  * while(tt.väli("eellane") != null)»
    »
    » tee.add(0, g.indeks(tt)+1); teele lisada tipu tt number
    » Tipp t0 = tt; // kaare lõpptipp;»
    » tt = g.tipp(Integer.parseInt(tt.väli("eellane")) - 1); tt := tt.eellane
    » Kaar k = g.kaar(tt, t0);»
    » sum += Double.parseDouble(k.nimi()); kogupikkusele lisada kaare

  »
  » if (sum > 0) kui tee ei olnud tühi (tipul b oli eellane)
  »
  » tee.add(0, g.indeks(tt)+1); teele lisada veel tipu tt number (esimen

  println("Tee tipust " + (g.indeks(a)+1) + " tippu " + (g.indeks(b)+1) + "
  println("\nTee: " + tee + ". Kogupikkus: " + sum);»

void läbida_sügavuti(Graaf g, Tipp t) // GTE_2-0
```

GTE_2-1b. Printida üks tee graafi kahe tipu vahel pärast läbimist, kasutades klassi *TeeInfo*.

Kirjutada ja testida vastav meetod *otsida_tee*.

```
Amadeus - workspace2017s\GTE_2-1b.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
»
» graaf g ning tipud a ja b
TeeInfo tInf = otsida_tee(g, a, b);»
println("Tee tipust " + aNr + " tippu " + bNr + " :");»
println(tInf.tipud_teel() + " Kogupikkus: " + tInf.pikkus());
TEST
»
» TeeInfo otsida_tee(Graaf g, Tipp a, Tipp b)»
»
» Antud: graaf g ja selle tipud a ning b
» Tulemus: leitud tee a..b, tagatstakse selle tee info
läbida_sügavuti(g, a); GTE_2-0
TeeInfo tInf = new TeeInfo(g, b);»
return tInf;»
» void läbida_sügavuti(Graaf g, Tipp t) // GTE_2-0
```

```
class TeeInfo{
    // isendimuutujad:
    Double pikkus; // tee kogupikkus
    Vector<Integer> tipud_teel; // selle teel olevate tipunumbrite list

    TeeInfo(Double p, Vector<Integer> tee){// konstruktor 1
        this.pikkus = p;
        this.tipud_teel = tee;
    }//konstruktor 1

    TeeInfo(Graaf g, Tipp b) { // konstruktor 2
        // Antud: graaf g, mille tippudele on seatud välju "eellane"
        // ja tipp b, millest leida tee (tagurpidi mööda eellasi) tipuni,
        // millel ei ole välja "eellane"
        // Tulemus: määratud isendiväljad (pikkus ja tipud_teel)
        ...
    }//konstruktor 2

    public String toString(){
        return "" + Math.round(pikkus*100)/100.0 + " " + tipud_teel;
    }

    // piilumeetodid("getters"):
    Double pikkus() {return pikkus;}
    Vector<Integer> tipud_teel(){return tipud_teel;}
}//TeeInfo
```

GTE_2-2. Printida kõik teed graafi kahe tipu vahel

Kirjutada ja testida rekursiivne Java-meetod *otsida_teed*.

Testi tulemus *graafTeed.txt* (vt GTE_2-1) korral:

```
Graaf failist graafTeed.txt.  
Leitud teed tipust 2 tippu 7:
```

```
Tee: [2, 3, 5, 7]. Kogupikkus: 5.0
```

```
Tee: [2, 4, 5, 7]. Kogupikkus: 7.0
```

```
Tee: [2, 6, 9, 8, 7]. Kogupikkus: 4.0
```

```
Amadeus - workspace2017s\GTE_2-2.algjava*
projekt: AlgJavaWorks2017oCDB

TEST
Sisestada graaf g:
String fNimi = "graafTeed.txt";»
//fNimi = "graafTeedSuur.txt";»
new GraafiJoonistaja(fNimi);»
Graaf g = Graaf.sisestadaTabelist(fNimi);»
println("\nGraaf failist " + fNimi + ".");»
määrata lähte- ja sihttipu numbrid aNr ja bNr
Tipp a = g.tipp(aNr - 1); lähtetipp
Tipp b = g.tipp(bNr - 1); sihttipp
println("Leitud teed tipust " + aNr + " tippu " + bNr + ":");»
otsida_teed(g, a, b);»
println("\n-----.");»

TEST
void otsida_teed(Graaf g, Tipp t, Tipp b)»
    spetsifikatsioon
    »
    » Antud: graaf g ja selle tipud t ning b
    » Tulemus: graaf g läbitud sügavuti;
    » . g tippudele on lisatud välju "eellane" ja "vaadeldud";
    » . prinditud kõik teed (tagurpidi <b, b.eellane, ..., a>)
    » . koos nende kogupikkustega
    spetsifikatsioon
```


GTE_2-3. Salvestada kõik teed graafi kahe tipu vahel

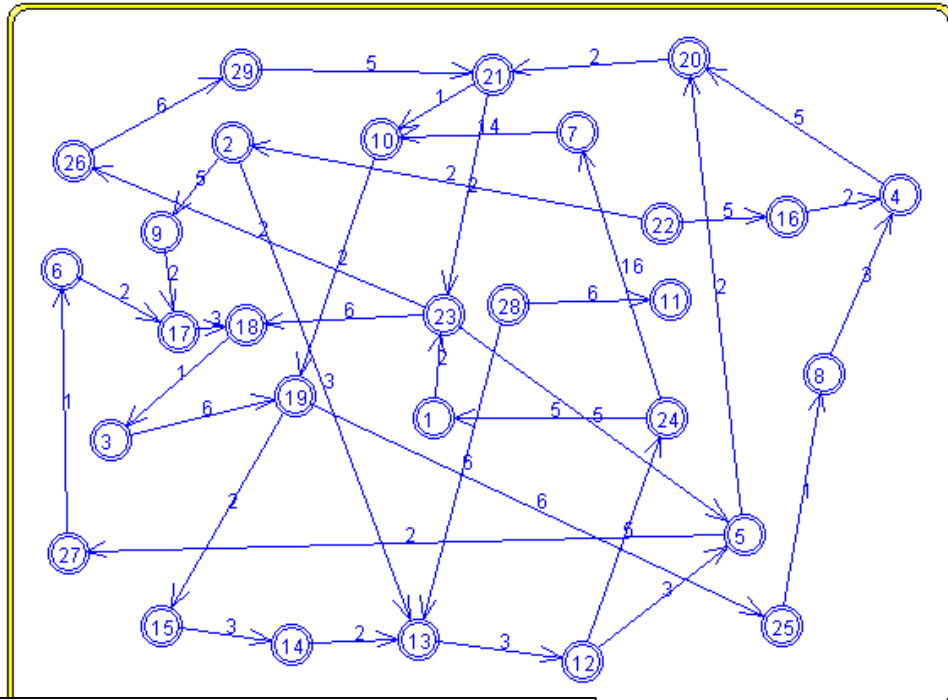
Kirjutada ja testida vastav meetod *otsida_teed*.

```
void otsida_teed(Graaf g, Tipp t, Tipp b, Vector<TeeInfo> teed)»  
»  
» spetsifikatsioon  
»  
» Antud: antud graaf g, selle tipud t ja b  
» . ning koht 'teed' leitavate teede salvestamiseks  
» Tulemus: g tippudele on lisatud välju "eellane" ja "vaadeldud";  
» . salvestatakse kõik teed (tagurpidi <b, b.eellane, ..., eellaseta  
» . koos kogupikkusega  
»  
» spetsifikatsioon  
t.seadaVäli("vaadeldud", "x"); panna lipp  
»  
» if (t == b) // oleme sihil (järjekordselt)»  
»  
» TeeInfo tInf = new TeeInfo(g, b);»  
» teed.add(tInf);»  
»  
» else»  
» for (Tipp v : g.naabrid(t)) // jätkata naabritest  
»  
»
```

```
import java.util.Vector;»  
import java.util.Collections; min / max  
» Kasutatakse: TeeInfo.class  
»»  
» TEST  
»  
» Sisestada graaf g  
» määrata lähte- ja sihttipp (tipud a ja b)  
» Vector<TeeInfo> teed = new Vector<TeeInfo>(); tulemuse koht  
» otsida_teed(g, a, b, teed);»  
» if (teed.size() == 0)  
» printida leitud teed:  
» println("\n Leitud teed: ");»  
»  
» * for(TeeInfo tInf : teed)»  
»  
» println(tInf.tipud_teel() + " Kogupikkus: " + tInf.pikkus());»  
»  
» leitud teedest lühima ning pikima kogupikkus  
» TeeInfo tInfmin =  
» Collections.min( teed, (tInf1, tInf2) -> (int)(tInf1.pikkus() - tInf2.pikkus()));»  
» TeeInfo tInfmax =  
» Collections.max( teed, (tInf1, tInf2) -> (int)(tInf1.pikkus() - tInf2.pikkus()));»  
» println("\n Vähim pikkus: " + tInfmin.pikkus());»  
» println("\n Suurim pikkus: " + tInfmax.pikkus());»  
» println("\n-----.");»  
» TEST
```

Suurem näide.

Kõik teed graafis *graafTeedSuur.txt* tipust nr 1 tippu nr 12.



TABEL

1)	2-->23
2)	3-->13 5-->9
3)	6-->19
4)	5-->20
5)	2-->27 2-->20
6)	2-->17
7)	14-->10
8)	3-->4
9)	2-->17
10)	2-->19
11)	
12)	3-->5 5-->24
13)	3-->12
14)	2-->13
15)	3-->14
16)	2-->4
17)	3-->18
18)	1-->3
19)	2-->15 6-->25
20)	2-->21
21)	2-->23 1-->10
22)	5-->16 2-->2
23)	2-->26 5-->5 6-->18
24)	16-->7 5-->1
25)	1-->8
26)	6-->29
27)	1-->6
28)	6-->13 6-->11
29)	5-->21

0) graafi ala.
 : vasakuga. Eemaldamine: parem klõps valitud tipul (2 x parem klõps).
 : valida algustipp, seejärel lõpptipp.
 Users\kiho\AlgPython\workspace2014\graafTeedSuur.txt Värskendamine: pidev.

Lähtetipu nr: 1, sihttipu nr: 12

Leitud teed:

- [1, 23, 26, 29, 21, 10, 19, 15, 14, 13, 12] Kogupikkus: 28.0
- [1, 23, 5, 27, 6, 17, 18, 3, 19, 15, 14, 13, 12] Kogupikkus: 32.0
- [1, 23, 5, 20, 21, 10, 19, 15, 14, 13, 12] Kogupikkus: 24.0
- [1, 23, 18, 3, 19, 15, 14, 13, 12] Kogupikkus: 25.0

Vähim pikkus: 24.0
 Suurim pikkus: 32.0

-----.