

Praktikum 12

Teede otsimine graafi sügavuti läbides

- **Harjutusülesanded**

GTE_2-1. Eellaste seadmine

GTE_2-2. Kaks abimeetodit

GTE_2-3. Üks tee graafi kahe tipu vahel

GTE_2-4. Kõik teed graafi kahe tipu vahel

Harjutusülesanded

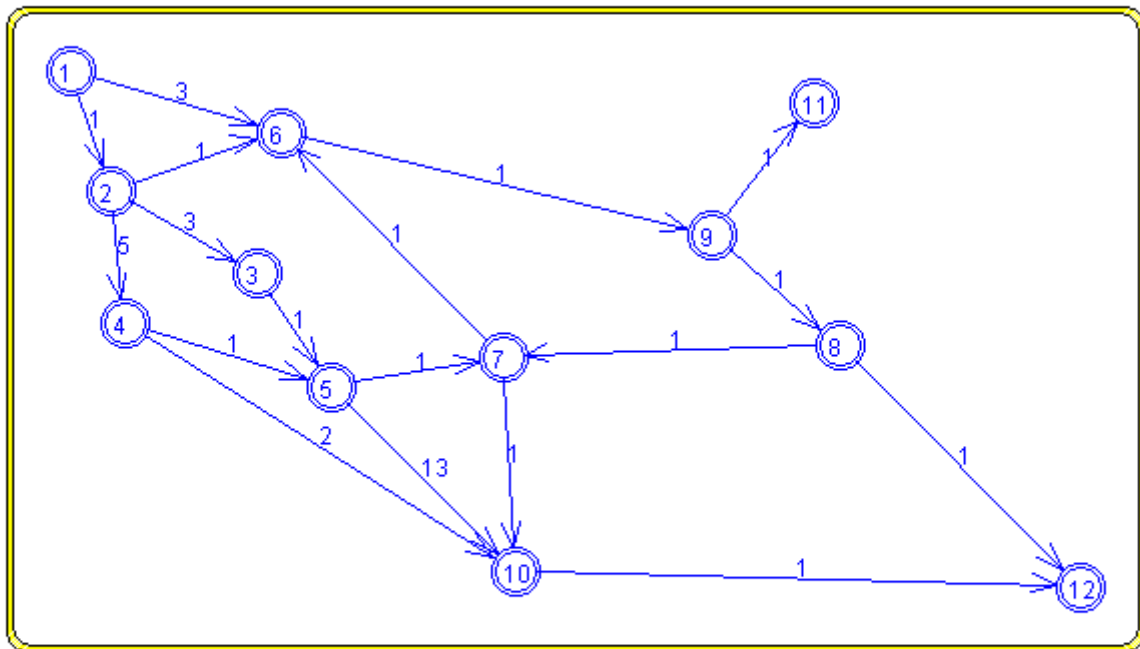
GTE_2-1. Eellaste seadmine

Kirjutada ja testida rekursiivne Java-meetod *eellased_teel*.

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

```
Tee tipust 1 tippu 12: [1, 2, 3, 5, 10, 12]
```

C:_ISIKLIK\ownCloud\AlgJava\graafTeed.txt



MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.

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

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 - workspace2016s\GTE_2_00.algjava

Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

projekt: AlgJavaWorks2016oC

```
void eellased_teel(Graaf g, Tipp t, Tipp b)
```

spetsifikatsioon

- » Antud: graaf g ja selle tipud t ning b
- » Tulemus: 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>)

spetsifikatsioon

```
t.seadaVäli("vaadeldud", "x");
```

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

```
return;
```

```
for (Tipp v : g.naabrid(t))
```

```
if (v.väli("vaadeldud") == null) t naaber v veel väisamata
```

```
v.seadaVäli("eellane", "" + (g.indeks(t)+1));  
eellased_teel(g, v, b);
```

GTE_2-2. Kaks abimeetodit

Kirjutada Java-meetodid *tee_pikkus* ja *tee* ning rakendada neid meetodi *eellased_teel* testiosas TEST.

Testi tulemusi *graafTeed.txt* (vt eelmine slaid) korral:

```
Tee tipust 1 tippu 12:  
[1, 2, 3, 5, 10, 12]  
tee pikkus: 19
```

```
Tee tipust 4 tippu 11:  
[4, 5, 7, 6, 9, 11]  
tee pikkus: 5
```



```
Amadeus - workspace2016s\GTE_2_01.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2016oCDB

import ee.ut.kiho.aa.graaf.*;
import java.util.Vector;

TEST

int tee_pikkus(Graaf g, Vector<Integer> tee)
» Antud: graaf g ja tee selles (teel olevate tippude numbrite list);
» . graafis g on kaare k pikkuseks k.nimi()
» Tulemus: antud tee kogupikkus (kaarepikkuste summa)

Vector<Integer> tee(Graaf g, Tipp b)
» Antud: graaf g, milles tipuväljad "eellane"
» . on määratud meetodiga 'eellased_teel'
» Tulemus: tagastatakse teel a .. b olevate tippude numbrite list,
» . kus a on eellaseta tipp

void eellased_teel(Graaf g, Tipp t, Tipp b)
```

GTE_2-3. Üks tee graafi kahe tipu vahel

Modifitseerida meetod *eellased_tee* meetodiks *otsida_teed*, viies abimeetodite rakendamise selle sisse (vt kõrvalolev skeem). Testida.

10.11.2016

```
Amadeus - <noname23>*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

int tee_pikkus(Graaf g, Vector<Integer> tee)
Vector<Integer> tee(Graaf g, Tipp b)

void otsida_teed(Graaf g, Tipp t, Tipp b)»
»
» spetsifikatsioon
» »
» » Antud: graaf g ja selle tipud t ning b
» » Tulemus: 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 tipu
» » . printitud tee a..b ja selle pikkus (a on eellaseta tipp)
» »
» » spetsifikatsioon
» » t.seadaVäli("vaadeldud", "x");»
» »
» » if (t == b) kui oleme sihil
» » »
» » » printida tee a .. b ja selle pikkus
» » » »
» » » Vector<Integer> tee = tee(g, b);»
» » » println("Tee: " + tee);»
» » » println(" tee pikkus: " + tee_pikkus(g, tee));»
» » » printida tee a .. b ja selle pikkus
» » » //return;»
» »
» » for (Tipp v : g.naabrid(t))
```

GTE_2-4. Kõik teed graafi kahe tipu vahel

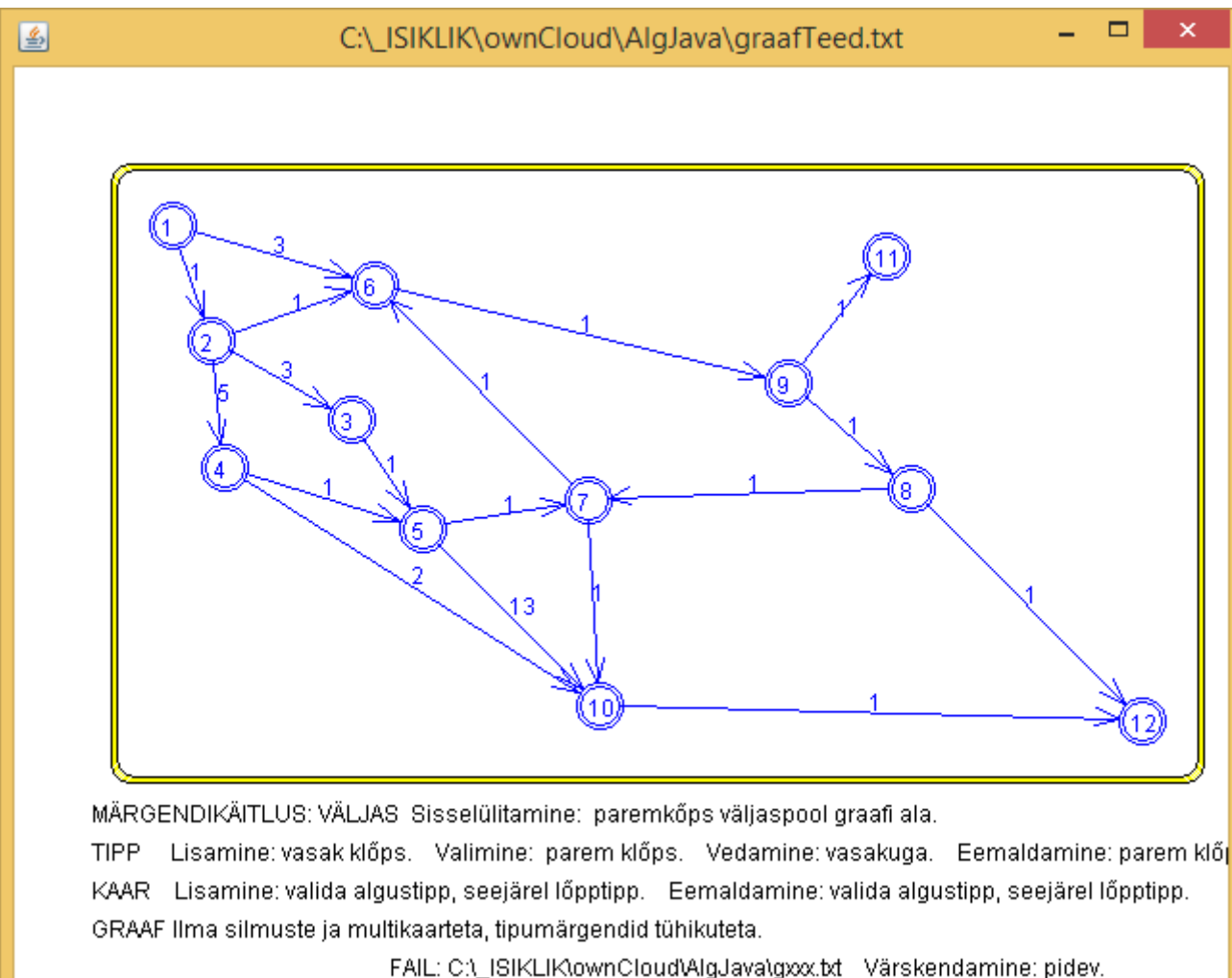
GTE_2-4-1. Printida kõik teed graafis

Modifitseerida meetod `otsida_teed` meetodiks, mis prindib kõik teed (ja nende pikkused), mis suubuvad sihttippu `b`.

Testida.

Testi tulemus `graafTeed.txt` korral:

```
Tee: [1, 2, 3, 5, 10, 12]
tee pikkus: 19
Tee: [1, 2, 3, 5, 7, 6, 9, 8, 12]
tee pikkus: 10
Tee: [1, 2, 3, 5, 7, 10, 12]
tee pikkus: 8
Tee: [1, 2, 4, 5, 10, 12]
tee pikkus: 21
Tee: [1, 2, 4, 5, 7, 6, 9, 8, 12]
tee pikkus: 12
Tee: [1, 2, 4, 5, 7, 10, 12]
tee pikkus: 10
Tee: [1, 2, 4, 10, 12]
tee pikkus: 9
Tee: [1, 2, 6, 9, 8, 7, 10, 12]
tee pikkus: 7
Tee: [1, 2, 6, 9, 8, 12]
tee pikkus: 5
Tee: [1, 6, 9, 8, 7, 10, 12]
tee pikkus: 8
Tee: [1, 6, 9, 8, 12]
tee pikkus: 6
```



GTE_2-4-2. Otsida kõik teed graafis

Modifitseerida meetod `otsida_teed` meetodiks, mis salvestab antud listi kõik teed, mis suubuvad sihttippu `b`.

Testida.

Testi tulemus `graafTeed.txt` (vt eelmine slaid) korral:

```
Tee: [1, 2, 3, 5, 7]
```

```
tee pikkus: 6
```

```
Tee: [1, 2, 4, 5, 7]
```

```
tee pikkus: 8
```

```
Tee: [1, 2, 6, 9, 8, 7]
```

```
tee pikkus: 5
```

```
Tee: [1, 6, 9, 8, 7]
```

```
tee pikkus: 6
```

Lühima tee pikkus on 5

```
Amadeus - <noname29>*
```

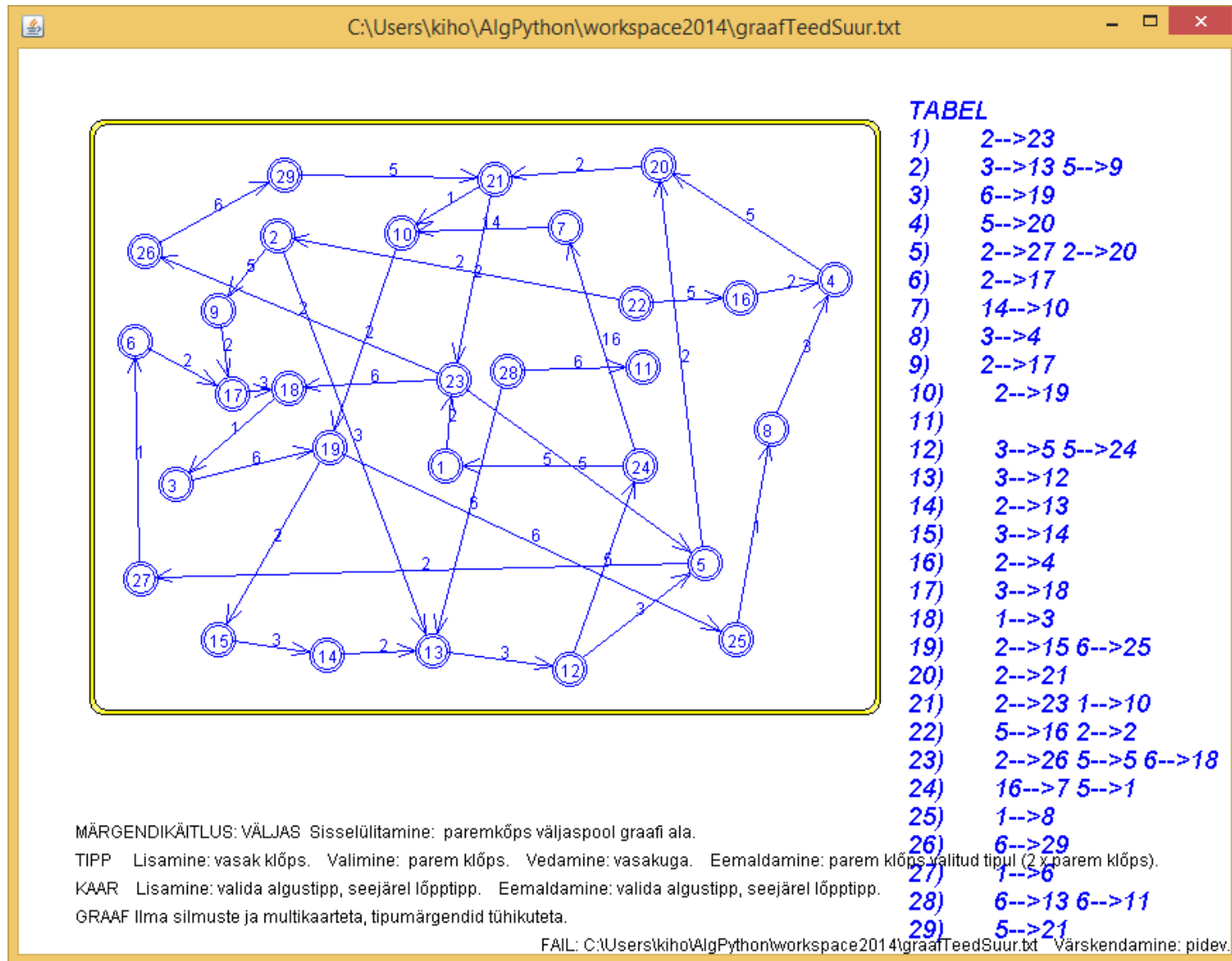
```
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
```

```
TEST
»
String fNimi = "graafTeed.txt";»
new GraafiJoonistaja(fNimi);»
Graaf g = Graaf.sisestadaTabelist(fNimi);»
int aNr = 1; lähtetipu number
int bNr = 7; sihttipu number
Tipp a = g.tipp(aNr - 1);»
Tipp b = g.tipp(bNr - 1);»
Vector<Vector<Integer>> teed = new Vector<Vector<Integer>> ();»
otsida_teed(g, a, b, teed);»
int minPikkus = Integer.MAX_VALUE;»
for(Vector<Integer> tee : teed)
println("----- \n Lühima tee pikkus on " + minPikkus);»
TEST
int tee_pikkus(Graaf g, Vector<Integer> tee)
Vector<Integer> tee(Graaf g, Tipp b)
void otsida_teed(Graaf g, Tipp t, Tipp b, Vector<Vector<Integer>> teed)
»
» spetsifikatsioon
» » Antud: graaf g, selle tipud t ja b ning list 'teed'
» » Tulemus: g tippudele on lisatud välju "eellane" ja "vaadeldud";
» » . listi teed on salvestatud kõik teed a..b, kus a on eellaseta tipp)
» spetsifikatsioon
» »
```

Suurem näide.

Kõik teed graafis *graafTeedSuur.txt*

tipust nr 1 tippu nr 12.



Lähtetipu nr: 1, sihttipu nr: 12
pikkus tee

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