

<http://kodu.ut.ee/~kiho/ads/Praktikum/>

## 7. GTE Teed graafis

# Mõisted

**Graaf** [Õpik, ptk 2.4]

[ ka <http://kodu.ut.ee/~kiho/ads/Praktikum/> > --. Graafid. Puud. Kahendpuud ]

**Graafi läbimine. Teed graafis** [Õpik, ptk 6.2, 6.3]

# Graafi näide

Kastid (vt Ülesannete kogu, ülesanne 4.29) ja nendevaheliste mahtuvusseoste graaf

Antud kastid:

A[alus=9.0x9.0; h=0.2; d=0.1]

B[alus=20.0x2.0; h=2.0; d=0.1]

C[alus=10.0x1.0; h=1.0; d=0.1]

D[alus=22.0x9.0; h=9.0; d=0.1]

E[alus=50.0x3.0; h=2.5; d=0.1]

Mahtuvusrelatsioon:

Kast C mahub kasti B

Kast C mahub kasti E

Kast C mahub kasti D

Kast B mahub kasti E

Kast B mahub kasti D

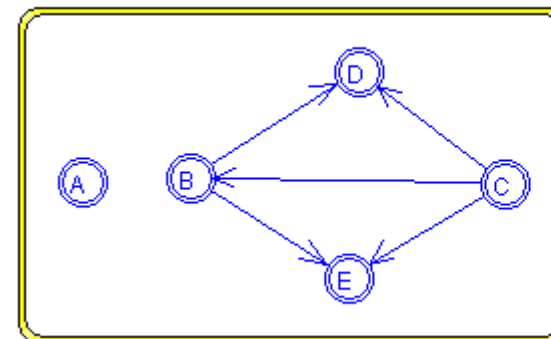
-----  
Kastide hulk:

A C B E D

Seoste hulk:

(C, B) (C, E) (C, D) (B, E) (B, D)

-----



TABEL

1) A

2) C -->3 -->4 -->5

3) B -->4 -->5

4) E

5) D

# ÜLESANDED

## GTE\_1. Etüüdid graafil

- GTE\_1-0. Saavutatavad tipud
- GTE\_1-1. Puustruktuuri kontroll
- GTE\_1-2. Graaf sümmeetriliseks
- GTE\_1-3. Tsüklilisuse ja sidususe kontroll

## GTE\_2. Teed graafis

- GTE\_2-0. Läbida graaf sügavuti, seades tipuvälju „eellane“
- GTE\_2-1. Leida üks (suva)tee graafi kahe tipu vahel
- GTE\_2-2. Leida kõik teed graafi kahe tipu vahel
- GTE\_2-3. Leida Dijkstra meetodil lühim tee graafi kahe tipu vahel

## GTE\_1-0. Saavutatavad tipud

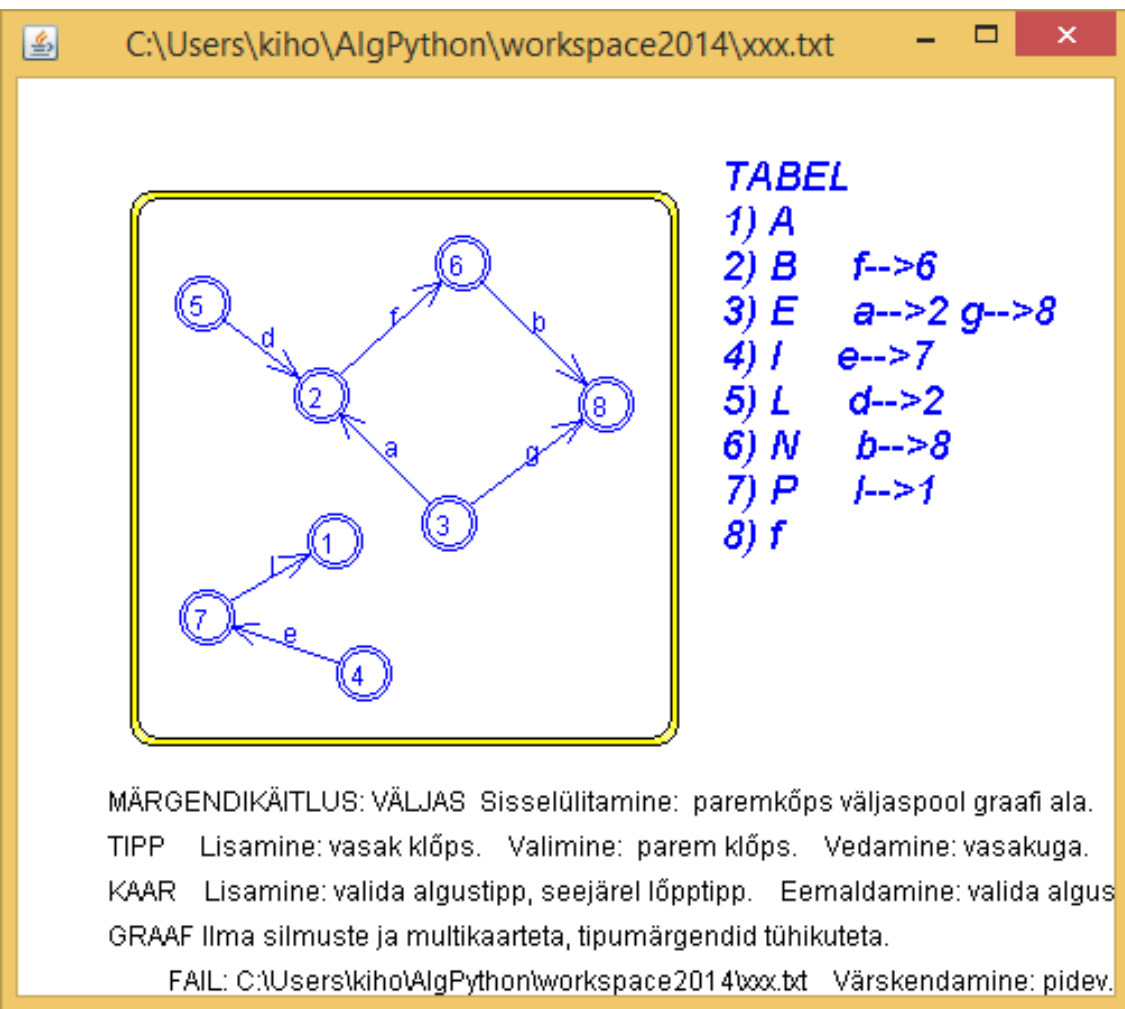
Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.

Antud: graaf ja selle tipp  $t$ .

Tulemus: tippude arv, mis on saavutatavad sellest tipust lähtuvalt (*incl* tipp  $t$ ).

[http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/--GP2P/N%C3%A4ited/Java/Graaf\\_N%C3%A4ide1.java](http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/--GP2P/N%C3%A4ited/Java/Graaf_N%C3%A4ide1.java)

### Testi tulemuse näide



The screenshot shows a window titled "C:\Users\kiho\AlgPython\workspace2014\xxx.txt". Inside, there is a directed graph with 8 nodes (1-8) and edges labeled with letters. Node 1 is at the bottom left, node 2 is in the middle left, node 3 is in the middle right, node 4 is at the bottom right, node 5 is at the top left, node 6 is at the top right, node 7 is at the bottom left, and node 8 is at the middle right. Edges are: 1 to 2 (d), 2 to 6 (f), 6 to 8 (b), 8 to 2 (a), 2 to 3 (g), 3 to 1 (l), 4 to 7 (e), 7 to 1 (l).

**TABEL**

1) A	
2) B	f-->6
3) E	a-->2 g-->8
4) I	e-->7
5) L	d-->2
6) N	b-->8
7) P	l-->1
8) f	

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi ala.  
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasakuga.  
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida algus  
GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.  
FAIL: C:\Users\kiho\AlgPython\workspace2014\xxx.txt Värskendamine: pidev.

Saavutatavaid tipust nr 1: 1  
Saavutatavaid tipust nr 2: 3  
Saavutatavaid tipust nr 3: 4  
Saavutatavaid tipust nr 4: 3  
Saavutatavaid tipust nr 5: 4  
Saavutatavaid tipust nr 6: 2  
Saavutatavaid tipust nr 7: 2  
Saavutatavaid tipust nr 8: 1

## GTE\_1-1. Puustruktuuri kontroll

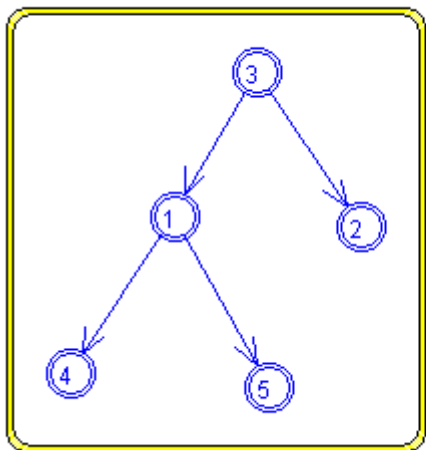
Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: graaf  $g$ .

Tulemus: kontrollitud, kas graaf  $g$  on puu (juurega graafi  $g$  mingis tipus).

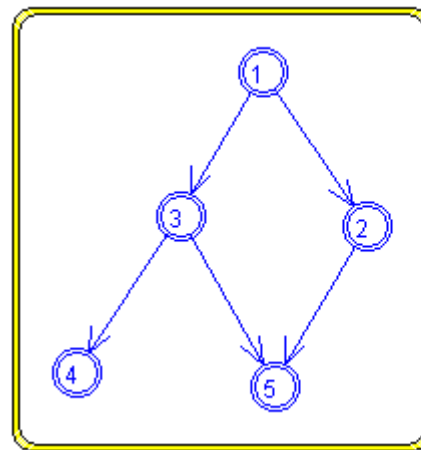
Vihje: üks vajalikest faktidest: puu korral kehtib teatud vahekord tippude ja kaarte arvude vahel.

Testi tulemusi



TABEL

- 1) -->4 -->5
- 2)
- 3) -->1 -->2
- 4)
- 5)



TABEL

- 1) -->2 -->3
- 2) -->5
- 3) -->4 -->5
- 4)
- 5)

Graaf failist graafTestGTE\_1\_1-3.txt.

On puustruktuur juurega tipus nr 3.

Graaf failist xxx.txt.

Ei ole puustruktuur.

## GTE\_1-2. Graaf sümmeetriliseks

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

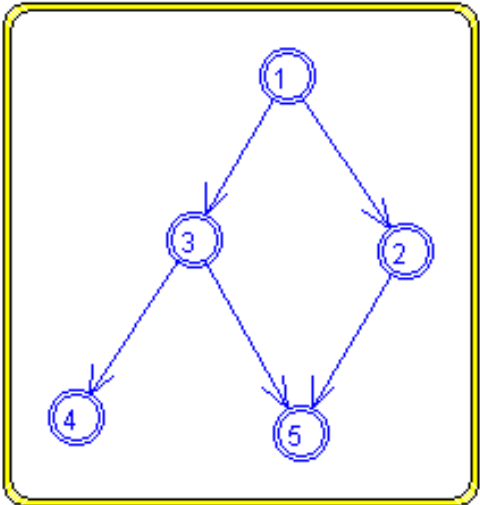
Antud: graaf  $g$ .

Tulemus: graafi  $g$  kaarte hulka lisatud puuduvad vastandkaared.

### Testi tulemuse näide

xxx.txt ==/sümmeetriliseks/==> yyy.txt

C:\Users\kiho\AlgPython\workspace2014\xxx.txt



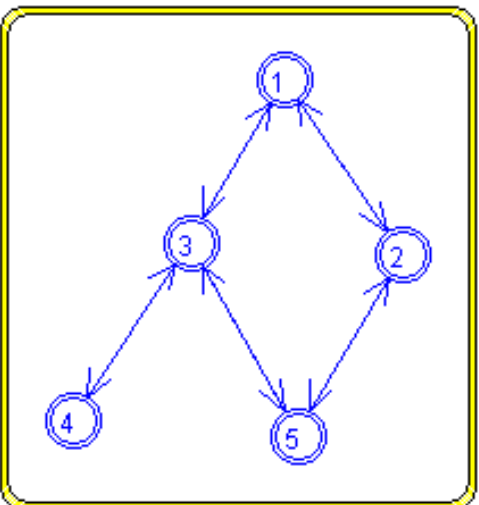
**TABEL**

1)	-->2 -->3
2)	-->5
3)	-->4 -->5
4)	
5)	

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi  
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasak klõps.  
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida lõpptipp.  
GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

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

C:\Users\kiho\AlgPython\workspace2014\yyy.txt



**TABEL**

1)	-->2 -->3
2)	-->5 -->1
3)	-->4 -->5 -->1
4)	-->3
5)	-->2 -->3

MÄRGENDIKÄITLUS: VÄLJAS Sisselülitamine: paremkõps väljaspool graafi  
TIPP Lisamine: vasak klõps. Valimine: parem klõps. Vedamine: vasak klõps.  
KAAR Lisamine: valida algustipp, seejärel lõpptipp. Eemaldamine: valida lõpptipp.  
GRAAF Ilma silmuste ja multikaarteta, tipumärgendid tühikuteta.

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

### GTE\_1-3. Tsüklilisuse ja sidususe kontroll

Kirjutada ja testida meetodid järgmiste ülesannete lahendamiseks.

#### GTE\_1-3a.

Antud: sümmeetriline graaf  $g$ .

Tulemus: kontrollitud, kas graafis  $g$  leidub tsükleid.

Vihje: üks võimalusi – eemaldada tipp, mille (väljund)aste on 0 või 1.

#### GTE\_1-3b.

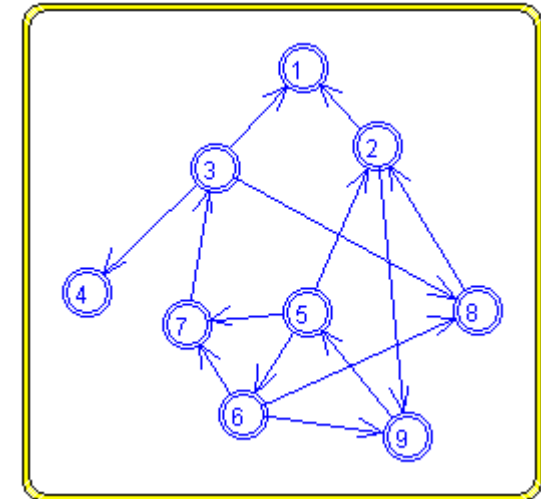
Antud: sümmeetriline graaf  $g$ .

Tulemus: kontrollitud, kas graaf  $g$  on sidus.

#### GTE\_1-3c. (Keerulisem)

Antud: antisümmeetriline, nummerdatud tippudega graaf  $g$ .

Tulemus: kui graafis  $g$  ei ole tsükleid, tagastatakse 0, vastasel korral tagastatakse mingil tsüklil asuva mingi tipu number.

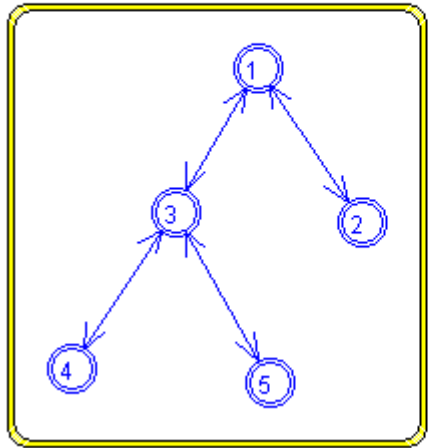


Tippu nr 8  
läbib vähemalt üks tsükkel.



# GTE\_1-3 a ja b testi tulemusi

Jyri\ownCloud\AlgJava\yyy.txt

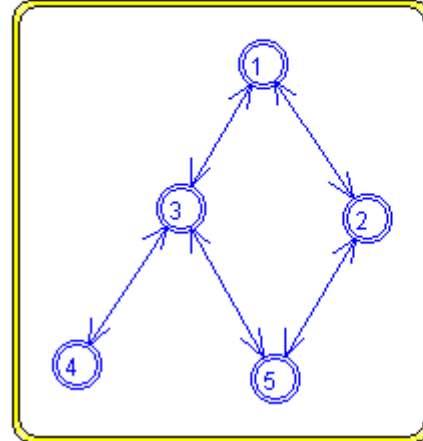


**TABEL**

- 1) -->2 -->3
- 2) -->1
- 3) -->4 -->5 -->1
- 4) -->3
- 5) -->3

On tsükleid: false  
On sidus: true

Jyri\ownCloud\AlgJava\yyy.txt

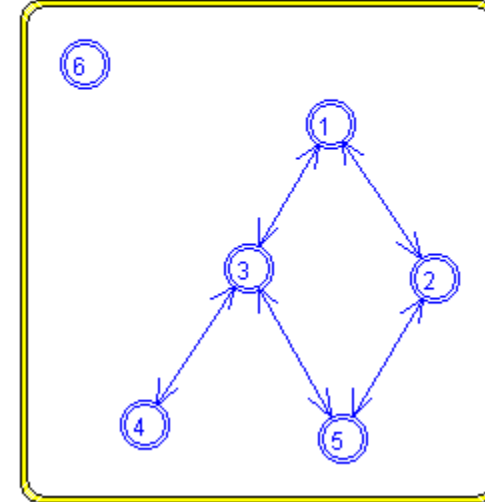


**TABEL**

- 1) -->2 -->3
- 2) -->5 -->1
- 3) -->4 -->5 -->1
- 4) -->3
- 5) -->2 -->3

On tsükleid: true  
On sidus: true

Jyri\ownCloud\AlgJava\zzz.txt



**TABEL**

- 1) -->2 -->3
- 2) -->5 -->1
- 3) -->4 -->5 -->1
- 4) -->3
- 5) -->2 -->3
- 6)

On tsükleid: true  
On sidus: false

## GTE\_2-0. Läbida graaf sügavuti, seades välju „eellane“

Kirjutada ja testida rekursiivne Java-meetod *läbida\_sügavuti*.

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

```
TEST
```

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

```
spetsifikatsioon
```

- »
- » Antud: graaf g ja selle tipp t
- » Tulemus: g läbitud, lähtudes tiput t;
- » . g tippudele on lisatud välju "eellane" ja "väisatud";
- » . x.väli("eellane") =
- » . <selles tippu number, kust läbimise käigus tippu x tuldi>

```
spetsifikatsioon
```

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

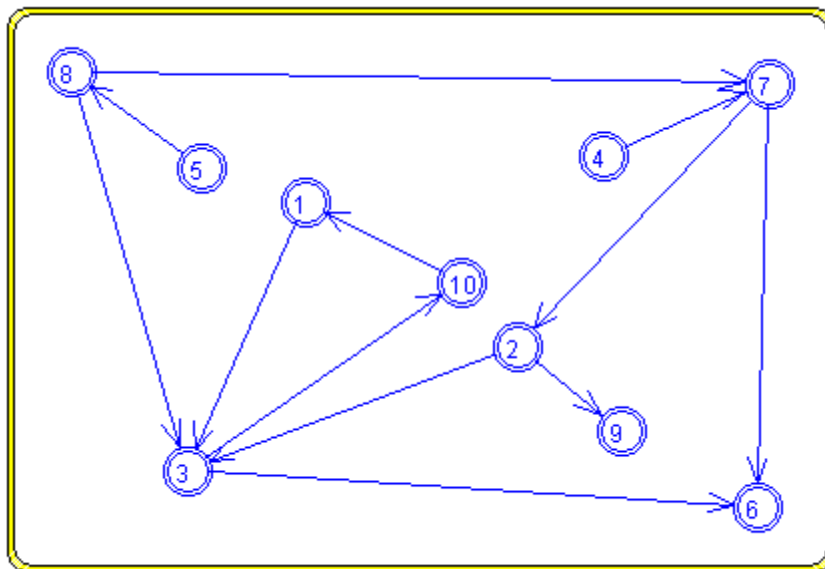
```
for (Tipp v : g.naabrid(t)) minna edasi naabritest
```

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

```
v.seadaVäli("eellane", "" + (g.indeks(t)+1)); v eellaseks tippu t nr  
läbida_sügavuti(g, v);»
```

### Testi tulemuse näide

Läbitakse graaf failist gTest11.txt, lähtudes tipust nr 3



### TABEL

- 1) väisatud=x; eellane=10; A -->3
- 2) B -->9 -->3
- 3) väisatud=x; C -->10 -->6
- 4) D -->7
- 5) E -->8
- 6) väisatud=x; eellane=3; F
- 7) G -->2 -->6
- 8) H -->7 -->3
- 9) I
- 10) väisatud=x; eellane=3; J -->1

## TEST

**Sisestada graaf g:**

```
String fNimi = "gTest11.txt";»  
//fNimi = "graafTeed.txt";»  
Graaf.kuvada(fNimi);»  
Graaf g = Graaf.failist(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.faili(fNimi);»
```

```
Graaf.kuvada(fNimi);»
```

```
»»
```

 mõne tee (alates lähtetipust) printimine

## TEST