

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

8. GTO Graafi toes

Mõisted

Klasside käitlemine. Galler-Fischeri meetod [Õpik, ptk 3.3]

Graafi minimaalne toes [Õpik, ptk 6.4]

ÜLESANDED

GT0_1. Galler-Fischeri klassikäitlus

GT0_2. Kruskali meetod

ISE_8. Iseseisva töö nr 8 näiteskeeme

GTO_1.

Lahendada ülesanne 1.3

<http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/8-GTO/%C3%9Clesanded/Peafail.pdf>

Klassile *GallerFischer.java* lisada ka testimisotstarbeline peameetod (*main*) järgmise skeemi kohaselt:

n := 9;

teha *n* isoleeritud, tühja tipuga graaf *g* (`new Graaf(n, 0)`);

seada Galler-Fischeri struktuur *gf* (`GallerFischer gf = new GallerFischer(g);`)

valida 2 juhutippu graafis *g*;

ühendada (kui need on eri klassides) neid tippe sisaldavad klassid

ja väljastada info jooksva klassijaotuse kohta, nt

1) printida graafi *g* tabel (`g.toString()`) või kuvada graaf *g*

2) printida klasside kõrgused (leiab tippudest, mille "viit" = null)

3) printida klasside arv

kui klasside arv = 1, siis

← väljuda tsüklist

GTO_2.

Lahendada ülesanne 1.4

<http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/8-GTO/%C3%9Clesanded/Peafail.pdf>

ISE_8

Dialogi ja tulemuse näide

Eesti asulate/teede plaanid:

1. g_linnad_teed_mini.txt
2. g_linnad_teed_midi.txt
3. g_linnad_teed.txt
4. gTest14.txt
5. gTest14a.txt

Palun plaani number: 1

Plaanil on järgmised asulad:

- 1.Narva
- 2.Põltsamaa
- 3.Pärnu
- 4.Rakvere
- 5.Tallinn
- 6.Tartu
- 7.Valga
- 8.Viljandi
- 9.Võru

Palun lähteasula number: 8

Lähteasula: Viljandi.

Asulate läbimise eeskiri mööda minimaaltoest (toese "maksumus" 692):

- 8 Viljandi
- 2 Põltsamaa
- 6 Tartu
- 9 Võru
- 7 Valga
- ==>9 Võru
- ==>6 Tartu
- 4 Rakvere
- 5 Tallinn
- ==>4 Rakvere
- 1 Narva
- ==>4 Rakvere
- ==>6 Tartu
- ==>2 Põltsamaa
- ==>8 Viljandi
- 3 Pärnu
- ==>8 Viljandi

Kokku läbitud: 1384 km

ISE_8

Programmi skeem

```
»
»
preambul
»
» Algoritmid ja andmestruktuurid. 2018/2019 sügissemester
» Arvutipraktikum. Iseseisev töö nr 8
» RÄNNAK GRAAFI TOESEL
» Autor: Ülli Õpilane
preambul
import ee.ut.kiho.aa.graaf.*;»
import java.util.Vector; listid
import java.util.Collections; sort()
LAHENDAMINE
int arvuna(String s)
Vector<Kaar> Kruskal_toes(Graaf g)
void Kruskal_toesegraaf(Graaf g, Vector<Kaar> A)
»
void rännak(Graaf g, Tipp t)»
»
» Antud: mingi graafi toes sümmeetrilise graafina g
» . ja selle tipp t (lähtetipp);
» Tulemus: prinditud graafi tippude järjekord g läbimisel lähtudes tipust t;
» . iga läbitava tipu korral prinditakse selle number ja märgend;
» . kui tippu on jõutud uuesti (st tagasi liikumisel)
» . siis lisatakse veel prefiks "==">"
println((g.indeks(t)+1) + " " + t.nimi());»
t.seadaVäli("x", "x"); vajalik, et ei läheks naabrile, kust tulnud
for (Tipp v : g.naabrid(t))
```

ISE_8

Meetodite skeeme

```
» Vector<Kaar> Kruskal_toes(Graaf g)»
[Õpik, lk 111, joonis 6.16]
  spetsifikatsioon
  »
  » Antud: teede graaf g; g on sümmeetriline (orienteerimata);
  » .      serva (teed) maksumusega (pikkusega) p kujutab
  » .      vastandkaarte paar, kus
  » .      mõlemal kaarel on välja "nimi" väärtuseks p
  » Tulemus: tagastatakse A --
  » .      g minimaalsesse toesse kuuluvate kaarte list,
  » .      milles iga kaar kujutab parajasti ühte toese serva;
  spetsifikatsioon
  Vector<Kaar> A = new Vector<Kaar>(); tulemus-list
  »»
  GallerFischer GF = new GallerFischer(g);»
  » iga tipp on omaette klassis, klassipuude kõrgused (tipuväli "h") =1:
  Vector<Kaar> servakaared = new Vector<Kaar>();»
  » servakaared: kõik kaared graafis;
  » .      iga serva {a, b} jaoks tuleb kaks kaart, a->b ja b->a
  » i = 0 .. g.m()-1
  » sorteerida list 'servakaared' kaaremaksumuste järgi (mittekahanevalt):
  »
  » Kruskali põhitsükkel:
  »
  » for ( Kaar kaar : servakaared)»
  »   (täiendatakse A)
  »   if (GF.ühendada_klassid(kaar.algus(), kaar.löpp()))
  return A;»
```

```
» void Kruskal_toesegraaf(Graaf g, Vector<Kaar> A)»
  post-Kruskal
  » Antud: graaf g ja selle toesekaarte list A (Kruskal_toes tulemus)
  » Tulemus: graafi g jäetud ainult toeseservad
  » NB! Antud graaf g ei säili.
  »»
  » eemaldada kõik tipuväljad igas g tipus:
  g.eemaldadaTipuväljad();»
  » graafi g kaared:
  Vector<Kaar> orig_kaared = new Vector<Kaar>();»
  » i = 0 .. g.m()-1
  » graafis g jätta alles ainult toesse (A) kuuluvad kaared:
  » for (Kaar kaar : orig_kaared)
  »   g.sümmeetriliseks, lisades vastandkaared:
  int m0 = g.m();»
  » i = 0 .. m0-1
```

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

[Arvutipraktikum/8-GTO/%C3%9Clesanded/Java/GallerFischerMall.java](http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/8-GTO/%C3%9Clesanded/Java/GallerFischerMall.java)
[Arvutipraktikum/8-GTO/%C3%9Clesanded/Java/KruskalMall.java](http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/8-GTO/%C3%9Clesanded/Java/KruskalMall.java)