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

3. MJ Magasin ja järjekord

Mõisteid

Magasin (*Stack*)

Järjekord (*Queue*)

Üldjärjekord (üldistatud järjekord, *deque***)**

Randomiseeritud (mitte-deterministlik, stohhastiline) algoritm

Praktikumitööd

MJ_1. Lahendada ülesanne 1.5* (labürindi läbimise üks tee). Vt *LabürindidM_Mall.java*, *LabürintM.java*.

MJ_2. Koostada randomiseeritud meetod antud labürindis lühima tee leidmiseks sisenemiskohast väljumiskohta. Labürint – vt ülesanne 1.5* . Idee: meetod leiab järjekordse tee, kusjuures koht kuhu jooksvast kohast *k* edasi minna valitakse juhuslikult koha *k* vabade naaberkohtade seast. Selle meetodi paljukordse rakendamise tulemuste seas leidub (väga tõenäoliselt) ka nõutud lühim tee.
(Kui on ette teada, mitu erisugust tulemust saab üldse olla, siis randomiseeritud meetodi paljukordne rakendamine võimaldab leida neist sobiva 100%-lise tõenäosusega.)

* http://kodu.ut.ee/~kiho/ads/Praktikum/Arvutipraktikum/3-MJ/%C3%9Clesanded/Peafail.pdf

Ühe tee leidmise meetodi võimalik skeem

```
»
_
» Kasutab klassi LabürintM
 import java.util.Stack;»
 import java.util.Collections; »
 import java.util.ArrayList; »
 import java.util.Arrays; »
 »
  static Stack<int[]> leidaTee(LabürintM lab)»
 »
  » Antud: labürint lab
  » Tulemus: otsitakse tee (sisenemiskoht-->väljumiskoht) antud labürindis;
  » .          tagastatakse magasin, milles kohad leitud teel;
  » .          tagastatav magasin on tühi, kui teed ei leitud
   int[] oleme    = lab.sisenemiskoht();   jooksev koht
   int[] lõpukoht = lab.väljumiskoht();  väljumiskoht
   »
  » teel sisendkoht-->oleme läbitud kohtade magasin:
   Stack<int[]> mag = new Stack<int[]>();»
  for(;;)
  » kohad leitud teel on magasinis mag (või on see tühi)
   return mag;»
```

```
»
* for(;;)»
 »
  »
   if (Arrays.equals(oleme, lõpukoht))  jõutud väljumiskohta
  »
   mag.push(lõpukoht);»
   break;»

  lab.set(oleme, 1);   panna koht 'oleme' kinni ringikäimise vältimiseks
 » leida koha 'oleme' esimene vaba naaberkoht (naabreid läbi vaadates):
  int[] vabaNaaberkoht = null;»
  //...»
  »
   if (vabaNaaberkoht != null)»
  »
   mag.push(oleme);   siit hiljem veel proovida edasi minna
   oleme = vabaNaaberkoht;   läheme edasi vabale naaberkohale
   continue;»

   koha 'oleme' vaba naabrit ei leitud
  »
   if (mag.isEmpty())  midagi enam tagasi võtta ei ole
  »
   break;»

  oleme = mag.pop();   tagasivõtt
```