

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

4. PSK Paisksalvestus. Loendamine

Mõisteid

Paisktabel (*Hashtable, HashMap*)

Paisktabeli kasutamine loendamisülesannetes

Paiskfunksioon, kollisioonid (põrked)

Praktikumitööd

PSK_1. Järjendi elementide esinemiste loendamine

PSK_2. Järjendi elementide võrdkomplektide esinemiste loendamine

PSK_3. Järjendi elementide vähemalt-esinemiste loendamine

PSK_4. Paiskfunktsiooni väärtuste jaotus

PSK_1. Järjendi elementide esinemiste loendamine [Ülesannete kogu, ül nr 17.4]

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: järjend a (listina).

Tulemus: info järjendi a elementide korduste kohta listina,
mille elemendiks on paar kujul (\langle element \rangle , \langle selle elemendi korduste arv \rangle).

Näiteks järjendi $a = [9, 0, -2, 9, 8, 6, 0, 8, 9, -1]$ korral on tulemuseks list
[[0, 2), (6, 1), (8, 2), (9, 3), (-2, 1), (-1, 1)].

```
import java.util.HashSet;
```

Java

```
HashSet<Integer> h = new HashSet<Integer>(Arrays.asList(a));
```

```
-- hulk  $h$  sisaldab kõik listi  $a$  elemendid (tüüpi Integer) ühekordselt
```

```
-- arvu  $x$  esinemiste arv (tüüpi long) listis  $a$ :
```

```
Arrays.stream(a).filter(elem -> elem == x).count()
```

```
////////////////////////////////////// efektiivsem:
```

```
import java.util.Hashtable;
```

```
Hashtable<Integer, Integer> tabel = new Hashtable<Integer, Integer>();
```

```
for(int x : a){
```

```
    Integer n = tabel.get(x);
```

```
    tabel.put(x, (n == null)? 1 : n+1);
```

```
}
```

```
-- listis  $a$  esinenud arvude hulk (tüüpi Set): tabel.keySet()
```

```
-- arvu  $x$  esinemiste arv (tüüpi Integer) listis  $a$ : tabel.get(x)
```

preambul

```
import java.util.ArrayList; »  
import java.util.Hashtable; »  
import java.util.Arrays; printimiseks testimisel
```

test

//test (vers 2, aeglasem)

```
»  
» ArrayList<int[]> esinemiste_arvud(int[] a)»  
»  
» Antud: täisarvude järjend (massiivina) a  
» Tulemus: luuakse ja tagastatakse järjend (list), kuhu on salvestatud  
» . paarid [<elemendi väärtus> , <selle esinemiste arv>].  
» . Näiteks, kui a = [9, 0, -2, 9, 8, 6, 0, 8, 9, -1], siis  
» . tagastatakse [[0, 2], [6, 1], [8, 2], [9, 3], [-2, 1], [-1, 1]]  
» tabel kiireks loendamiseks:  
» Hashtable<Integer, Integer> tabel = new Hashtable<Integer, Integer>();  
»  
» + for(int x : a)»  
» Integer n = tabel.get(x);»  
» tabel.put(x, (n == null)? 1 : n+1);»  
»  
» tulemuspaaride hulk:  
» ArrayList<int[]> tulem = new ArrayList<int[]>();»  
» + for(int võti : tabel.keySet()) iga tabelis oleva võtme (arvu) korral  
» tulem.add(new int[]{võti, tabel.get(võti)}); lisada paar [arv, kordusi]  
»  
» return tulem;»
```

PSK_2. Järgendi elementide võrdkomplektide esinemiste loendamine

[Ülesannete kogu, ül nr 17.6]

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: järjest a (listina).

Tulemus: list elementidega (n, vn) , kus

vn on järjestis a leiduvate (pikimate) võrdsetest

elementidest koosnevate ennikute arv, $n > 1$, $vn > 0$

Näiteks järjesti $[9, 0, 5, -2, 9, 8, 6, 0, 8, 5, 9, -1, 5, 4, -1, 5, 9]$ korral

on tulemuseks list $[(2, 3), (4, 2)]$, st

võrdsete paare on 3 (nendeks on $(0,0)$, $(8,8)$ ja $(-1,-1)$)

(võrdsete kolmikuid ei ole)

võrdsete nelikuid on 2 (nendeks on $(9,9,9,9)$ ja $(5,5,5,5)$)

Soovitav on rakendada meetodit (nt *esinemisi*) ülesandest PSK_1.

Näiteks

$a = [9, 0, 5, -2, 9, 8, 6, 0, 8, 5, 9, -1, 5, 4, -1, 5, 9]$,

korral

$esinemisi(a) = [(0, 2), (4, 1), (5, 4), (6, 1), (8, 2), (9, 4), (-1, 2), (-2, 1)]$.

Selles järjestis olevate paaride teiste liikmete järjest $b = [2, 1, 4, 1, 2, 4, 2, 1]$,

b ilma ühtedeta $c = [2, 4, 2, 4, 2]$

Vastuse annab

$esinemisi(b) = [(2, 3), (4, 2)]$.

PSK_3. Järjendi elementide vähemalt-esinemiste loendamine

[Ülesannete kogu, ül nr 17.5]

Kirjutada ja testida meetod järgmise ülesande lahendamiseks

Antud: järjend a (listina) ja positiivne arv k .

Tulemus: järjend (list), kuhu on lisatud iga vähemalt k korda esineva elemendi korral paar (\langle elemendi väärtus \rangle , \langle selle esinemisi \rangle).

Näiteks, kui $a = [9, 0, -2, 9, 8, 6, 0, 8, 9, -1]$, siis

$k = 2$ korral on tulemuseks $[(0, 2), (8, 2), (9, 3)]$

$k = 1$ korral on tulemuseks $[(0, 2), (6, 1), (8, 2), (9, 3), (-2, 1), (-1, 1)]$.

Testi tulemuse näide:

```
Antud: [9, 0, -2, 9, 8, 6, 0, 8, 9, -1]
```

```
Elemendid, mis esinevad vähemalt 4 korda: []
```

```
Elemendid, mis esinevad vähemalt 3 korda: [9]
```

```
Elemendid, mis esinevad vähemalt 2 korda: [0, 8, 9]
```

```
Elemendid, mis esinevad vähemalt 1 kord: [0, 6, 8, 9, -2, -1]
```

PSK_4. Paiskfunktsiooni väärtuste jaotus [Ülesannete kogu, ül nr 8.12]

Valida ja realiseerida üks paiskfunktsioon $h(võti, M)$, kus **võtmeks on isiku nimi** ning arv M on paisktabeli ridade (slottide) arv.

Koostada ja rakendada seda paiskfunktsiooni sisaldav programm, mis arvutab mingi andmefaili korral paiskfunktsiooni väärtused kõigi kirjete korral ning väljastab statistika **analoogiliselt** järgmise näitega:

```
Fail: prot_TRR2014_133km.txt
Kirjete arv: 1970
M = 2001
      2 kirjet ühte slotti 303 korda
      3 kirjet ühte slotti 48 korda
      4 kirjet ühte slotti 5 korda
      5 kirjet ühte slotti 1 kord
Unikaalseid h väärtusi: 1195 (60.66%)
```

Java

```
import java.util.HashSet;
```

ANSI-kodeeringus sõnekujulisele võtmele *võti* vastav täisarvuline räsiväärtus r arvutatakse

```
int r = võti.hashCode();
```

Andmefailide valik vt järgmine slaid.

	Andmefail (Kogust <i>Andmefaile.zip</i>)	Võtmevälja asukoht (indeks) kirjes
1	prot_KOVvalimised2013.txt	0
2	prot_TRR2014_70km.txt	3
3	prot_TRR2014_133km.txt	3
4	prot_tm_2013_31ab_lp.txt	1
5	prot_tm_2013_63ab_lp-1.txt	3
6	prot_PaideTyri32_13km.txt	3
7	prot_Saaremaa40_pj.txt	3
8	prot_TartuSygisj_10km.txt	3
9	prot_Tartu_rattaralli2015	3
10	prot_Tartu_rattamaraton2016	3
11	prot_rm_2013_40ab_lp.txt	1
12	prot_Tartu_jm_2017.txt	3

Andmefailid on ANSI-kodeeringus tekstifailid. Igas reas on üks kirje, väljaarvatud read, mis algavad "//" , need loetakse väljakommenteerituks (st ei ole kirjed). Kirjes on väljad eraldatud tabulatsioonisümboliga.

Amadeus - workspace2017s\PSK_4.algjava*

Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

projekt: AlgJavaWorks2017oCDB

```
>>
>>
>> preambul
import java.util.Scanner;»
import java.io.FileReader; »
import java.io.IOException; »
>>>
import java.util.ArrayList;»
import java.util.Hashtable; loendamisel
>>>
>> määrata faili nimi (fNimi) ja võtmevälja indeks selle kirjetes (vInd)
>> määrata M -- paisktabeli ridade (slottide) arv
ArrayList<Integer> koodid = new ArrayList<Integer>(); paisk-koodide koht
int arv = 0; kirjete arv
>> faili fNimi kirjevõtmete paisk-koodid (paiskfunktsiooni väärtused) --> listi 'koodid',
>> ühtlasi leitakse ka võtmete (vaadeldud kirjete) arv
>> try
>> erinevate paisk-koodide esinemiste loendamine:
ArrayList<int[]> esin_arvud = esinemiste_loendus(koodid);»
>> statistika ja vastuse väljastamine
ArrayList<int[]> esinemiste_loendus(ArrayList<Integer> a)
int h(String võti, int M)
```

Amadeus - workspace2017s\PSK_4... - □ ×

Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi

projekt: AlgJavaWorks2017oCDB

```
!! try»
» avada fNimi fail fr:
  FileReader fr = new FileReader(fNimi);»
  Scanner in = new Scanner(fr);»
  arv = 0;»
  teha paisk-koodid
  * while (in.hasNextLine())»
    »
    String rida = in.nextLine();»
    if (rida.startsWith("//"))
      String[] ss = rida.split("\\t");»
      String võti = ss[vInd];»
      int h = h(võti, M);»
      koodid.add((Integer)h);»
      arv++;»
  teha paisk-koodid
  fr.close();»
catch (IOException e)
```