

Praktikum 4

Kombinatorika. Rekursioon (II)

- Hulga kõikide alamhulkade hulk

Harjutusülesanded

- KMB_5. Alamhulkade hulk rekursiivselt
- KMB_6. Alamhulkade hulk mitterekursiivselt
- KMB_7. Osasummad
- KMB_8. Limiteeritud „kaaluga“ alamhulkade hulk
- KMB_9. Sõjaväerongi koosseisud (rekursioon)
- KMB_10. Sõjaväerongi koosseisude generaator
- KMB_11. Sõjaväerongi koosseisud lisatingimustega
- KMB_12.1. Arvude hulga suurim nullsummaga alamhulk
- KMB_12.2. Arvude hulga suurimad nullsummaga alamhulgad
- KMB_13. Algoritm kastide pakkimisprobleemi lahendamiseks

Iseseisev töö nr 2

Hulga kõikide alamhulkade hulk

Tähistus: $P(A)$ – hulga A kõigi alamhulkade hulk.

Variant I. Mitterekursiivne

$P(A)$ elemendid saame, valides alamhulgad hulgast A maskide abil,
 $mask = 0, 1, \dots, 2^n - 1$, kus $n = |A|$

Variant II. Rekursiivne või mitterekursiivne

$P(B \cup \{x\}) = P(B) \cup X$,
kus X on hulk, mille iga element on saadud
elemendi x lisamisel ühele hulga
 $P(B)$ elemendile.

Näide:

$$A = \{2, -1, 3\}$$

$$B = \{2, -1\}$$

$$A = B \cup \{3\}$$

$$P(B) = \{\{\}, \{2\}, \{-1\}, \{2, -1\}\}$$

$$\begin{aligned} P(A) &= P(B \cup \{3\}) = P(B) \cup X = \\ &= \{\{\}, \{2\}, \{-1\}, \{2, -1\}\} \cup \{\{3\}, \{2, 3\}, \{-1, 3\}, \{2, -1, 3\}\} = \\ &= \{\{\}, \{2\}, \{-1\}, \{2, -1\}, \{3\}, \{2, 3\}, \{-1, 3\}, \{2, -1, 3\}\} \end{aligned}$$

Variant III. Mitterekursiivne

Hulga A kõigi alamhulkade hulk hulga A
kombinatsioonide ühendina:

$$\begin{aligned} P(A) &= \{komb(A, 0)\} \cup \\ &\cup \{komb(A, 1)\} \cup \\ &\dots \\ &\cup \{komb(A, |A|)\} \end{aligned}$$

Näide:

$$\begin{aligned} P(\{2, -1, 3\}) &= \{\{\}\} \cup \\ &\cup \{\{2\}, \{-1\}, \{3\}\} \cup \\ &\cup \{\{2, -1\}, \{2, 3\}, \{-1, 3\}\} \cup \\ &\cup \{\{2, -1, 3\}\} = \\ &= \{\{\}, \{2\}, \{-1\}, \{3\}, \{2, -1\}, \{2, 3\}, \{-1, 3\}, \{2, -1, 3\}\} \end{aligned}$$

Harjutusülesanded

KMB_5. Alamhulkade hulk rekursiivselt

Kirjutada ja testida rekursiivne generaator järgmise ülesande lahendamiseks.

Antud: naturaalarvude hulk a (listina)

Tulemus: järjekordsel pöördumisel ($next()$) antakse välja a järjekordne alamhulk (listina).

Näiteks:

Klass: *Gen_Ah.java*

```
Antud hulk: [1, 3, 4, 2]
Selle alamhulkade hulk:
[]
[2]
[4]
[3]
[1]
[4, 2]
[3, 2]
[3, 4]
[1, 2]
[1, 4]
[1, 3]
[3, 4, 2]
[1, 4, 2]
[1, 3, 2]
[1, 3, 4]
[1, 3, 4, 2]
```

KMB_6. Alamhulkade hulk mitterekursiivselt

Kirjutada ja testida mitterekursiivne generaator järgmise ülesande lahendamiseks.

Antud: hulk a (massiivina)

Tulemus: järjekordsel pöördumisel ($next()$) antakse välja a järjekordne alamhulk (massiivina).

KMB_7. Osasummad

[Ülesannete kogu, ül nr 4.13]

Tulemuse näide:

```
Antud: [2, -3, 3, 3]
Selle osasummadeks on:
[-3, -1, 0, 2, 3, 5, 6, 8]
```

```
import java.util.LinkedHashSet;
```

Java (I)

Korduvate eemaldamine listist s:

```
LinkedHashSet<Integer> h = new LinkedHashSet<Integer>(s);
s.clear();
s.addAll(h);
```

```
import java.util.TreeSet;
```

Java (II)

Korduvaid elemente ei ole. Elemendid hoitakse loomulikus järjestuses.

Soovitav kasutada osasummade hoidmiseks.

Hulga tõkestatud alamhulkade hulk

Ülesannetes, kus tuleb leida vaid teatavat kriteeriumit rahuldavad alamhulgad, saab tihtipeale vältida kõikvõimalike alamhulkade läbivaatamist lahendamise käigus.

Näiteks on mõnel juhul teada, et vaatlusele „ei tasu“ enam võtta alamhulki ah' , mis on saadud alamhulgast ah sellele veel ühe elemendi lisamise teel.

KMB_8. Limiteeritud „kaaluga“ alamhulkade hulk

Kirjutada ja testida rekursiivne generaator järgmise ülesande lahendamiseks.

Antud: naturaalarvude hulk a (listina) ja alamhulga summa limiit („kaal“) L .

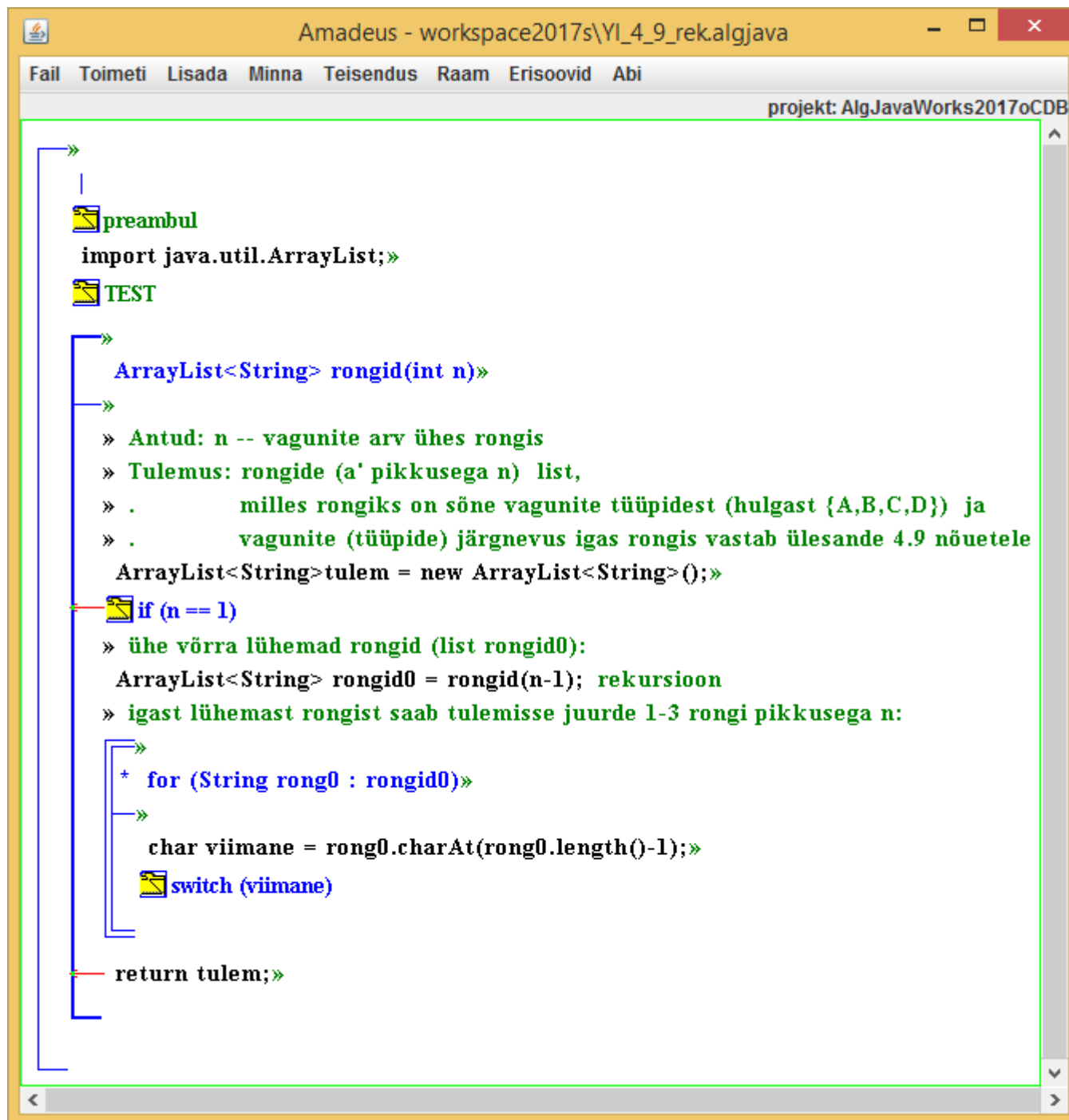
Tulemus: järjekordsel pöördumisel ($next()$)

antakse välja a järjekordne alamhulk (listina), mille summa $\leq L$.

Klass: *Gen_Ah.java*

KMB_9. Sõjaväerongi koosseisud (rekursioon)

[Ülesannete kogu, ül nr 4.9].



```
Amadeus - workspace2017s\YI_4_9_rek.algjava
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB

preambul
import java.util.ArrayList;»

TEST

ArrayList<String> rongid(int n)»
» Antud: n -- vagunite arv ühes rongis
» Tulemus: rongide (a' pikkusega n) list,
» . milles rongiks on sõne vagunite tüüpidest (hulgast {A,B,C,D}) ja
» . vagunite (tüüpide) järgnevus igas rongis vastab ülesande 4.9 nõuetele
ArrayList<String>tulem = new ArrayList<String>();»
if (n == 1)
» ühe võrra lühemad rongid (list rongid0):
ArrayList<String> rongid0 = rongid(n-1); rekursioon
» igast lühemast rongist saab tulemisse juurde 1-3 rongi pikkusega n:
+ for (String rong0 : rongid0)»
» char viimane = rong0.charAt(rong0.length()-1);»
switch (viimane)

return tulem;»
```

KMB_10. Sõjaväerongi koosseisude generaator

[Ülesannete kogu, ül nr 4.9].

Klass: *Gen_Rong.java*

KMB_11. Sõjaväerongi koosseisud lisatingimustega

Ülesannete kogu, ül nr 4.9. Täiendatud järgmiste tingimustega:

- . vagunite arv (n) hulgast {16 .. 30};
- . rongis peab olema vähemalt 4 vagunit igast tüübist;
- . rongi esimeses ja teises pooles peab olema ühepalju B-tüüpi vaguneid;
- . rongi esimeses ja teises pooles peab olema ühepalju C-tüüpi vaguneid;
- . julgestusmeeskondi saab paigutada ainult D-tüüpi vagunisse, seepärast
 - . rongi esimeses veerandis ($n/4+1$ vagunit) peab olema vähemalt 2 D-tüüpi vagunit,
 - . rongi keskpaigas peab olema vähemalt 1 D-tüüpi vagun,
 - . rongi viimases veerandis peab olema vähemalt 1 D-tüüpi vagun

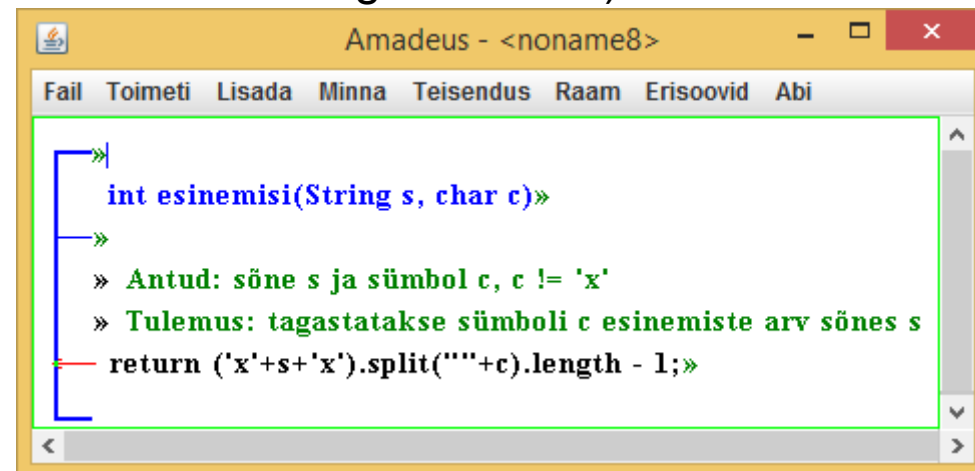
Leida tuleb üks selliseid tingimusi rahuldav rongikoosseis,

Näiteks $n=19$ korral sobib vastuseks rongikoosseis: BDBDACDACDABABACACD

Kasutatav klass: generaator *Gen_Rong.java*

(annab välja ül nr 4.9 nõuetele vastavaid rongikoosseise)

Meetod sümboli sõnes esinemiste loendamiseks:



```
Amadeus - <noname8>
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
int esinemisi(String s, char c)»
» Antud: sõne s ja sümbol c, c != 'x'
» Tulemus: tagastatakse sümboli c esinemiste arv sõnes s
return ('x'+s+'x').split(""+c).length - 1;»
```


KMB_12.1. Arvude hulga suurim nullsummaga alamhulk

Kirjutada ja testida Java-meetod, mis leiab antud täisarvude hulga suurima elementide arvuga (ühe) alamhulga, mille elementide summa on 0.

Meetodis tuleb rakendada indeksite kombinatsioonide generaatorit *Gen_Komb*.

Tulemuste näiteid:

```
Antud: [-29, 15, 63, 7, 52, 79, 37, 50, -16, 66, 92, 70, -44, 51, -25, 93]
Elemente 16.
```

```
Leitud alamhulk: [-29, 7, 37, -16, 70, -44, -25]
Elemente 7. Summa 0.
```

```
Antud: [-72, 75, 99, -73, -80, -50, 49, -41, -87, -49, -52, 72, 60, -55, -66]
Elemente 15.
```

```
Leitud alamhulk: [75, 99, -73, -50, 49, -41, -87, -49, 72, 60, -55]
Elemente 11. Summa 0.
```

Meetodi skeem:

```
Amadeus - workspace2017s\AhKahanevalt.algjava*
Fail Toimeti Lisada Minna Teisendus Raam Erisoovid Abi
projekt: AlgJavaWorks2017oCDB
TEST (peameetod main)
void print_ah0(ArrayList<Integer> hulk) KMB_12.1
  » Antud: arvude hulk (listina)
  » Tulemus: prinditud (üks) suurim alamhulk, mille elementide summa on 0
  int m = hulk.size(); elementide arv
  L
  * for(int k = m; k >= 0; k--)»
  k: "mitme kaupa", tsüklis väheneb
  Gen_Komb gen = new Gen_Komb(m, k); kombinatsioone indeksitest 0, ..., m-1 k kaupa
  int[] indkomb; koht järjekordsele kombinatsioonile
  * while((indkomb = gen.next()) != null)»
  » indkomb on üks kombinatsioon indeksitest 0, ..., m-1 k kaupa
  int summa = 0;»
  ArrayList<Integer> ah = new ArrayList<Integer>(); tulemuse koht
  * for(int i : indkomb)»
  »
  summa += hulk.get(i); liita summale
  ah.add(hulk.get(i)); lisada tulemusse
  if (summa == 0)
  L
```

KMB_12.2. Arvude hulga suurimad nullsummaga alamhulgad

Kirjutada ja testida Java-meetod, mis leiab antud täisarvude hulga kõik suurima elementide arvuga alamhulgad, mille elementide summa on 0.

Meetodis tuleb rakendada indeksite kombinatsioonide generaatorit *Gen_Komb*.

Tulemuse näide:

```
Antud: [34, 61, -6, 13, -29, 41, 67, 55, -39, -91, 32, -57, 80, -77, 7]
Elemente 15.
```

```
Leitud alamhulk: [61, -6, 13, -29, 67, -91, 32, -57, 80, -77, 7]
Elemente 11. Summa 0.
```

```
Leitud alamhulk: [34, -6, 13, -29, 67, 55, -39, 32, -57, -77, 7]
Elemente 11. Summa 0.
```

```
Leitud alamhulk: [34, -6, 13, -29, 41, 67, -39, -91, 80, -77, 7]
Elemente 11. Summa 0.
```

```
Leitud alamhulk: [34, -6, 13, -29, 41, 67, 55, -39, -91, 32, -77]
Elemente 11. Summa 0.
```

```
Leitud alamhulk: [34, 61, -6, -29, 41, 67, -39, -91, 32, -77, 7]
Elemente 11. Summa 0.
```

KMB_13. Algoritm kastide pakkimisprobleemi lahendamiseks [Ülesannete kogu, ül nr 4.29a]:

- Antud: kastide rivi, milles kastid on sorteeritud kõrguste järgi mittekahanevalt
- Tulemus: pakkimise eeskiri kastipaaride hulgana, kus paar (k_1, k_2) tähendab, et esimene neist (k_1) tuleb panna teise kasti (k_2) .

Teha mahtvusrelatsioon mr : kastipaaride hulk, kuhu on lisatud kõik sellised kastipaarid, mille korral paari esimene kast mahub paari teise kasti.

* Mahtvusrelatsiooni mr iga alamhulga $mrah$ korral
(võttes alamhulki suuruse, st elementide arvu vähenemise järjekorras)

Alamhulgas $mrah$ ei ole korduvaid kaste?
($mrah$)

Moodustada neljast kastist koosnev näiterivi, milles kastid on sorteeritud kõrguste järgi mittekahanevalt. Sooritada ülaltoodud algoritmis ettenähtud tegevused leidmaks pakkimisprobleemi lahendus.

Iseseisev töö nr 2

Esitamise tähtaeg rühmal inf1 (J. Liivi 2-202, E 12 -14): **7. oktoober 2017, kell 12.15**

Esitamise tähtaeg rühmal inf7 (J. Liivi 2-202, T 14 -16): **8. oktoober 2017, kell 14.15**