

1. Enniku mõiste

Olgu antud $n, n > 0$ ja mittenegatiivsete täisarvude järjest $m = [m_0, m_1, \dots, m_{n-1}]$.

Siis **ennikuks** nimetatakse n -elemendilist järjestit

$$e = [e_0, e_1, \dots, e_{n-1}], \text{ kus } 0 \leq e_j \leq m_j \text{ (} j = 0, 1, \dots, n-1 \text{)}.$$

Kui $n = 3$, siis on ennikuks arvukolmik, kui $n = 4$, siis arvunelik jne.

Näide. Kõik ennikud (arvukolmikud), kui $n = 3$ ja $m = [3, 1, 2]$:

1. [0, 0, 0]
2. [0, 0, 1]
3. [0, 0, 2]
4. [0, 1, 0]
5. [0, 1, 1]
6. [0, 1, 2]
7. [1, 0, 0]
8. [1, 0, 1]
9. [1, 0, 2]
10. [1, 1, 0]
11. [1, 1, 1]
12. [1, 1, 2]
13. [2, 0, 0]
14. [2, 0, 1]
15. [2, 0, 2]
16. [2, 1, 0]
17. [2, 1, 1]
18. [2, 1, 2]
19. [3, 0, 0]
20. [3, 0, 1]
21. [3, 0, 2]
22. [3, 1, 0]
23. [3, 1, 1]
24. [3, 1, 2]

Ennikut vaadeldakse koosnevat järkudest (nt kolmiku korral on ennikus kolm järku, üldiselt – n järku). Järgus indeksiga j olevat arvu nimetame j -nda järgu väärtuseks. Näites toodud igas ennikus on kolm järku, ennikus nr 20 on järkude väärtusteks vastavalt 3, 0 ja 1.

Järjendi m (vt jaotise alguses) elementi m_j nimetame j -nda järgu **laeks**; j -nda järgu minimaalne väärtus on 0, maksimaalne aga m_j .

Näiteks erijuhu $m = [1, 1, 1, 1]$ korral on kõikide ennikute hulgaks 4-kohaliste bitijärjendite hulk.

Mitmete programmeerimisülesanne lahendamise saab taandada kõigi (või osa) teatud järku ja teatud lagedega ennikute hulga läbivaatamisele. Märgime, et ennikute hulk võib osutada üsna suureks, Nimelt on lagede $m = [m_0, m_1, \dots, m_{n-1}]$ korral (n -järguliste) ennikute koguarvuks korrutis $\prod_{j=0}^{n-1} (m_j + 1)$.

2. Rakendusi

I Ülesanded, kus on tarvis läbi vaadata terve ennikute hulk

Kui ennikute koguarv ei ole ülemäära suur, siis sobib kasutada ennikute generaatorit, näiteks `Gen_Ennikud.java` veebilehelt <http://kodu.ut.ee/~kiho/ads/fall17/Mudelprogramme/Generaatorid/>. Muuhulgas võimaldab see hõlpsasti käsitleda uuritava objektihulga variatsioone, permutatsioone ja kombinatsioone.

Ülesanne 1. Leida antud n -elemendilisest objektide järjendist kõik korduvate elementidega kolmekaupade variatsioonid.

Iga variatsioon on määratud mingi (objektide indeksitest koosneva) indeksikolmikuga. Indeksikolmikuteks on siin aga kolme järguga ennikud, lagede järjendi $m = [n - 1, n - 1, n - 1]$ korral.

Märgime, et n -elemendilisest objektide järjendist J korduvate elementideta variatsioonid (k kaupa) saame, kui vaadelda ainult ennikuid (pikkusega k), milles korduvad elemendid (indeksid) on "keelatud". Kui aga piirduda üksnes kasvavate järguväärtustega ennikutega, siis nendeks on objektide kombinatsioonide indeksikomplektid. Järjendi J permutatsioonid on selle korduvate elementideta variatsioonid n kaupa.

II Ülesanded, kus saab piirduda vaid ennikute (suhteliselt väike-se) alamhulgaga

Tegemist on probleemidega, mille puhul kõikvõimalike ennikute arv on väga suur.

Ülesanne 2. Leida, kuidas saab vähima arvu euromüntidega täpselt tasuda antud summa.

Täpsemalt, algandmeteks on tasumisele kuuluv summa s (EUR) ja müntide vääringute järjend (näiteks) $v = [2, 5, 10, 20, 50, 100, 200]$. Leida tuleb täisarvulised kordajad võrrandis

$$a \cdot 2 + b \cdot 5 + c \cdot 10 + d \cdot 20 + e \cdot 50 + f \cdot 100 + g \cdot 200 = s \cdot 100,$$

kusjuures kordajate summa $a + b + c + d + e + f + g$ peab olema võimalikult väike.

Seega, vastust tuleb otsida 7-järguliste ennikute seast, kusjuures lagede järjendiks on $[s/2, s/5, s/10, s/20, s/50, s/100, s/200]$ – elemendiks müntide arv juhul, kui (saaks) tasuda ainult vastavas (ühes ja samas) vääringus müntidega.

Käesoleval juhul on kõikvõimalike ennikute arv väga suur, nimelt

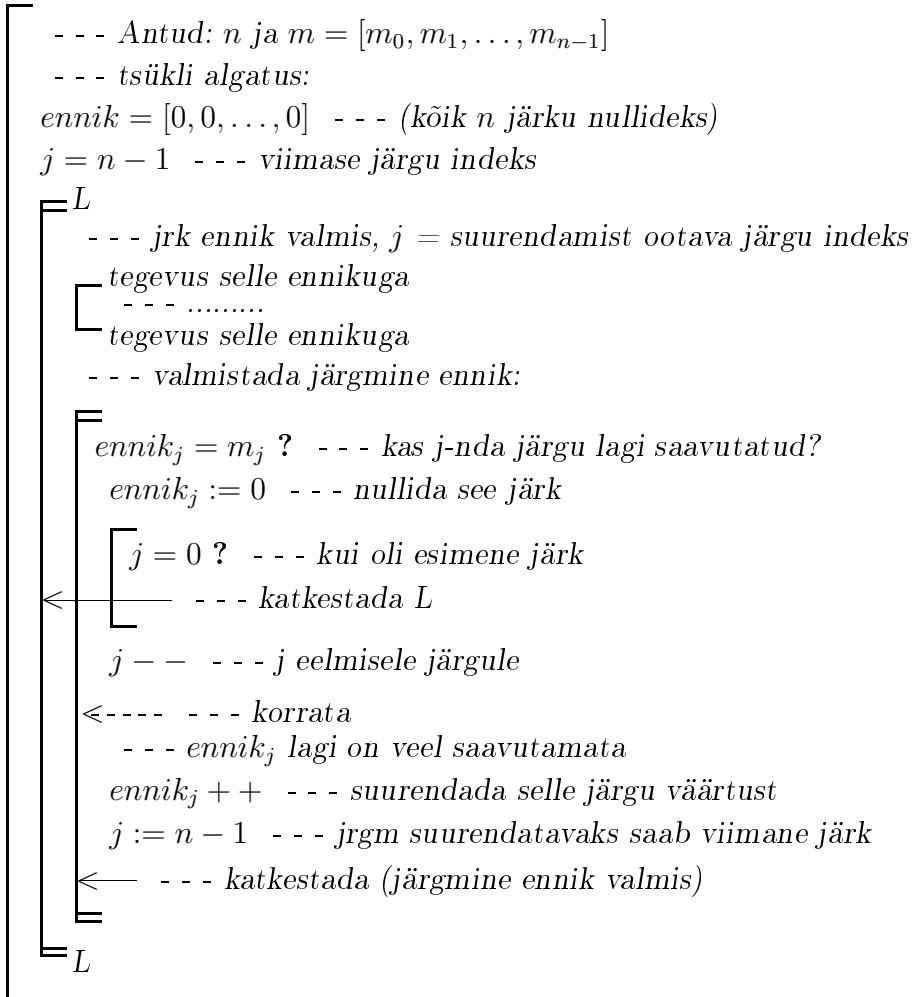
$$(s/2 + 1) \cdot (s/5 + 1) \cdot (s/10 + 1) \cdot (s/20 + 1) \cdot (s/50 + 1) \cdot (s/100 + 1) \cdot (s/200 + 1),$$

näiteks $s = 6.53 \text{ EUR}$ korral ligemale 36 600 000 000. Ometigi saab selle ülesande praktilise lahendamise korraldada suuremat osa ennikutest läbi vaatamata.

Paneme ka tähele, et antud vääringute järjendi v korral saab suvalise summa täpselt tasuda, kui $1 \in v \vee (2 \in v \wedge 5 \in v)$.

3. Ennikute läbivaatamise üldine tsükkel

I Skeem



II Java programmis

```
//////// ennikute läbivaatamine (antud on lagede massiiv m[]):
// tsükli algatus:
int n = m.length;
int[] ennik = new int[n]; // [0, 0, ..., 0]
int j = n-1; // viimase järgu indeks

L: for(;;){
    // jrk ennik valmis, j = suurendamist ootava järgu indeks

    ////////// tegevus selle ennikuga:
    ...
    ...
    ...
    ////////// tegevus selle ennikuga.

    for(;;){
        if (ennik[j] == m[j]){
            ennik[j] = 0; // nullida see järk
            if (j == 0) break L; // kui oli esimene järk
            j--; // j eelmisele järgule
            continue;
        }
        // ennik[j] lagi on veel saavutamata
        ennik[j]++; // suurendada seda järku
        j = n-1; // edasine suurendamine algab viimasest järgust
        break; // järjekordne ennik valmis
    }//for
} //L:for
///////// ennikute läbivaatamine.
```

III Ülesande 2 lahendamise Java-programm

Vt <http://kodu.ut.ee/~kiho/ads/fall17/>

Mudelprogramme/Ennikud/EnnikudY12.java.