

1. Ülesandeid iseseisvaks tööks

I Ülesanded kogust [1], ptk 5 ja 7:

5.1, 5.6-5.10, 5.12. 5.13, 5.14, 7.12. 7.13, 7.14.

II Lisaks

1.1. (s) Koostada rekursiivne meetod täisarvudest koosneva järjekorra sorteerimiseks kohapeal, kui

- on teada, et järjendis olevad arvud on paarikaupa erinevad;
- meetodis tohib kasutada ainult kuni kolme *int*-tüüpi abimuutujat.

1.2. (s) Koostada meetod, mis etteantud naturaalarvu n jaoks leiab vähima positiivse, ainult numbritest 9 ja 0 koosneva (kümnendsüsteemi) arvu x , mis jagub arvuga n .

1.3. Juku on kogunud pikema aja jooksul münte. Tal on nii 10-, 20-, 50-sendiseid münte (igauht täpselt 15 tükki) kui ka 1- ja 2-euroseid münte (kumbagi täpselt 10 tükki). Ta tahab osta nende eest spordikoti hinnaga n EUR (n on lõigus $20 \dots 50$, võib olla mittetäisarv). Koostada programm, mis magasin- ja/või järjekorraoperatsioone kasutades leiab kõikvõimalikud Juku käes olevate müntide kombinatsioonid, mis annavad kokku täpselt n EUR.

1.4. (s) Olgu lausearvutuse valem defineeritud järgmiselt:

- Tõeväärtused t (tõene) ja f (väär) on lausearvutuse valemid.
- Kui A on lausearvutuse valem, siis ka eitus $\neg A$ on lausearvutuse valem.
- Kui A ja B on lausearvutuse valemid, siis ka konjunktsioon ja disjunktsioon $A \wedge B$ ja $A \vee B$ on lausearvutuse valemid.
- Prioriteetid kahanemise järjekorras: eitus, konjunktsioon, disjunktsioon. Prioriteedi muutmiseks kasutatakse ümarsulge (nagu aritmeetilistes avaldistes).

Koostada meetod, mis magasin- ja/või järjekorraoperatsioone kasutades teeb lausearvutuse valemiga järgmise (sõneks)teisenduse:

- $\text{teisenda}(t) = "1"$,
- $\text{teisenda}(f) = "0"$,
- $\text{teisenda}(\neg A) = \text{"not("} + \text{teisenda}(A) + \text{"})"$,
- $\text{teisenda}(A \wedge B) = \text{"and("} + \text{teisenda}(A) + \text{","} + \text{teisenda}(B) + \text{"})"$,
- $\text{teisenda}(A \vee B) = \text{"or("} + \text{teisenda}(A) + \text{","} + \text{teisenda}(B) + \text{"})"$,

kus A ja B on suvalised lausearvutuse valemid (sõne kujul).

Näiteks:

$\text{teisenda}(t \wedge f) = \text{"and}(1,0)\text{"}$;

$\text{teisenda}((t \vee f) \wedge t) = \text{"and}(\text{or}(1,0),1)\text{"}$;

$\text{teisenda}(\neg(t \wedge f)) = \text{"not}(\text{and}(1, 0))\text{"}$.

1.5. (s) Labürint on esitatud bitimaatriksina, 0-element on vaba koht; lisaks on antud veel sisenemiskoht ja väljumiskoht kahe indeksipaarina (rea indeks, veeru indeks). Astuda (vabalt kohalt) uuele kohale saab maatriksi piires üles, alla, vasakule, paremale, tingimusel, et sihtkoht ei ole 1-element.

Koostada programm leidmaks (kui võimalik) üks tee antud labürindis sisenemiskohast väljumiskohta.

Näiteks labürindi

(7 x 9 maatriks):

| | 0: | 1: | 2: | 3: | 4: | 5: | 6: | 7: | 8: |
|----|----|----|----|----|----|----|----|----|----|
| 0: | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2: | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 3: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 4: | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5: | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 6: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

sisenemiskoha indeksid: 4 0

väljumiskoha indeksid: 2 7

korral võiks tulemuseks olla väljund kujul

Leitud tee märgistatud arvudega 2, 3, ..., 19.

(7 x 9 maatriks):

| | 0: | 1: | 2: | 3: | 4: | 5: | 6: | 7: | 8: |
|----|----|----|----|----|----|----|----|----|----|
| 0: | 1 | 0 | 1 | 1 | 10 | 11 | 0 | 0 | 1 |
| 1: | 5 | 6 | 7 | 8 | 9 | 12 | 0 | 1 | 0 |
| 2: | 4 | 1 | 0 | 1 | 0 | 13 | 1 | 19 | 0 |
| 3: | 3 | 0 | 0 | 0 | 1 | 14 | 17 | 18 | 1 |
| 4: | 2 | 1 | 0 | 0 | 1 | 15 | 16 | 0 | 1 |
| 5: | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 6: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

sisenemiskoha indeksid: 4 0

väljumiskoha indeksid: 2 7

Suunised

1. Ülesandeid iseseisvaks tööks

1.1. Lähedane: <https://www.geeksforgeeks.org/sorting-queue-without-extra-space/> **1.2.** Vt <https://www.geeksforgeeks.org/smallest-multiple-of-a-given-number-made-of-digits-0-and-9-only/> **1.4.** Ülesannet saab lahendada sarnaselt aritmeetiliste avaldiste teisendamisega infikskujult prefikskujule. **1.5.** Pida da magasinini, milles on kohad parajasti vaadeldaval teel. Ringikäimise vältimiseks märkida juba viibitud kohad arvuga 1.

Viited

- [1] A. Peder, J. Kiho, H. Nestra. *Algoritmid ja andmestruktuurid. Ülesannete kogu*. TÜ, 2017, 131 lk.