Lietuvos mokinių informatikos olimpiada

Mokyklos etapas • 2009 m. lapkričio 20 d.

Uždavinių sprendimai

Knyga (teorinis uždavinys, bendras uždavinys). Uždavinys savo sprendimo idėja panašus abiems amžiaus grupėms. Tik vyresniems skirtoje uždavinio versijoje knyga turi ne 20, o 100 puslapių. Toliau pateiktas uždavinio sprendimas, kai knyga turi 20 puslapių. Tačiau visiškai ta pati idėja tinka ir tam atvejui, kai knyga turi 100 puslapių.

Protingai versdamas Benas gali visada atversti paskutinį puslapį. Pradžioje knyga atversta per patį vidurį, todėl yra tiek pat vietų atversti knygai link pradžios, kiek ir link galo. Benas gali užtikrinti, kad visose vietose nuo knygos pradžios ligi vidurio knygą atvers jis, jei savo ėjimu vers puslapius su kuo didesniu numeriu (aišku, laikydamasis taisyklių). Tokiu būdu, Benas turės nemažiau vietų, kur atversti knygą, negu Adas. Kadangi pradeda Adas, jis tikrai pirmas nebeturės, kur atversti.

Pavyzdžiui, kai knyga turi 20 puslapių,situacija galėtų susiklostyti taip:

Berniukas	Atverčia	Jau atversti
—	—	**10**, **11**
Adas	12–13 psl.	10, 11 **12**, **13**
Benas	8–9 psl.	**8**, **9**, 10, 11, 12, 13
Adas	14–15 psl.	8, 9, 10, 11, 12, 13, **14**, **15**
Benas	6–7 psl.	**6**, **7**, 8, 9, 10, 11, 12, 13, 14, 15
Adas	16–17 psl.	6, 7, 8, 9, 10, 11, 12, 13, 14, 15, **16**, **17**
Benas	4–5 psl.	**4**, **5**, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
Adas	18–19 psl.	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, **18**, **19**
Benas	2–3 psl.	**2**, **3**, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Adas	20–ą psl.	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, **20**
Benas	1–ą psl.	**1**, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
Adas	nebegali atversti nei vieno puslapio	

Medus (uždavinys vyresniesiems). Kopdamas aukštyn ir žemyn iki avilio, esančio aukštyje h_i, meškiukas numes $2h_i$ kilogramų, o valgydamas to avilio medų — priaugs m_i kilogramų. Taigi jo svorio pokytis bus: $m_i - 2h_i$. Tačiau, kaip parašyta sąlygoje, reikia suskaičiuoti, kiek *daugiausiai* meškiukas gali priaugti. Daugiausiai bus priaugama, jei meškiukas lips į visus medžius, kurie jam „prideda svorio" ir nelips į likusius medžius. Žemiau pateikiamas programos pseudokodas:

```
read S, n
for i ← 1 to n
        do read h, m
            if m > 2h
                then S ← S + m − 2h
print S
```

Algoritmo požiūriu šis uždavinys lengvas ir galėtų tikti ir jaunesniems. Tačiau, duomenų, ypač sekos, skaitymas dažnokai būna sudėtingesnė užduotis mokiniams. Ypač jei bandoma skaityti po simbolį iki eilutės bei failo pabaigos. Tad uždavinys paskirtas vyresniųjų grupei. Jaunesniems būtų naudinga paanalizuoti sprendimą ir įsitikinti, kad duomenų skaitymas olimpiadiniuose uždaviniuose yra labai paprasta operacija.

Šokoladas (uždavinys jaunesniesiems). Žemiau pateiksime gražų, matematiškai pagrįstą ir *siektiną* sprendimą. Tačiau tai uždavinys jaunesniems. Tad puikiausiai tinkamas ir paprastesnis sprendimas, kai suskaičiuojami visi n dalikliai, ne didesni už dalmenį.

Šokolado plytelės ilgis ir plotis yra sveikieji skaičiai tarp 1 ir n. Be to, jei viena plytelės kraštinė yra lygi x, tai kita bus lygi n/x. Galime bandyti visas galimas vienos kraštinės ilgio reikšmes x ir tikrinti, ar galima matmenų $x \times n/x$ plytelė — taip bus, jei x dalina n be liekanos.

Pavyzdžiui, jei $n = 6$, tai rasime keturis n daliklius: 1, 2, 3 ir 6, ir atitinkamų matmenų plyteles: 1×6, 2×3, 3×2 ir 6×1. Tačiau plytelė 1×6 iš tiesų sutampa su plytele 6×1, o plytelė 2×3 — su 3×2. Kad tokių plytelių neįskaičiuotume po du kartus, tikrinsime x reikšmes nuo 1 tol, kol $x \leq n/x$, arba, perrašius kitaip, $x^2 \leq n$. Kitaip tariant, perrinksime *trumpesniosios kraštinės* ilgio reikšmes. Šitaip gausime ir gerokai efektyvesnę programą, kadangi bus tikrinami ne n potencialių kraštinių ilgių, bet tik $\sqrt{n}$.

Taigi šis uždavinys iš esmės yra daliklių skaičiavimo uždavinys. Jei $d(n)$ pažymėsime skaičiaus n daliklių skaičių, tai uždavinio atsakymas bus $\frac{d(n)}{2}$, arba $\frac{d(n)+1}{2}$, jei $n = k^2$ (n yra sveiko skaičiaus kvadratas). Žemiau pateikiamas programos pseudokodas:

```
read n
atsakymas ← 0
x ← 1
while x² ≤ n
      do if n mod x = 0
            then atsakymas ← atsakymas +1
         x ← x + 1
print atsakymas
```

Stulpai (bendras uždavinys). Eilutės simbolius reikia nagrinėti po vieną, visą laiką atsimenant, kurioje pozicijoje buvo aptiktas paskutinis + simbolis. Pirmasis eilutės simbolis visuomet yra +. Aptikus antrąjį + simbolį, reikia apskaičiuoti atstumą tarp jo, ir pirmojo. Aptikus bet kurį kitą + simbolį, reikia patikrinti, ar atstumas tarp jo ir paskutiniojo + simbolio yra toks pat. Žemiau pateikiamas programos pseudokodas.

```
1   read S
2   paskutinis ← 1
3   atstumas ← NEZINOMAS
4   for i ← 2 to length(S)
5         do if S[i] = +
6                then if atstumas = NEZINOMAS
7                       then atstumas ← i − paskutinis −1
8                       else  if atstumas ≠ i − paskutinis −1
9                                then atstumas ← SKIRTINGAS
10                    paskutinis ← i
11  print atstumas
```

Pseudokode naudojama konstanta SKIRTINGAS lygi -1, o konstanta NEZINOMAS gali būti bet kokia reikšmė, kuri negali būti uždavinio atsakymu, pavyzdžiui -2 arba 1000.

Jei įvykdoma 9 eilutė, tai paiešką būtų galima nutraukti. Tačiau jei ji ir tęsiama (kaip pateiktame pseudokode), tai kintamojo *atsakymas* reikšmė nebepasikeis ir liks lygi -1.

Pateikėme tik vieną iš daugelio galimų sprendimų. Kadangi duomenys yra eilutės pavidalo, galima sugalvoti daug įvairių sprendimų, kuriuose naudojamos darbui su eilutėmis skirtos funkcijos bei procedūros. Pavyzdžiui, galima pašalinti pirmąjį eilutės simbolį `+`, toliau pakaitomis ieškoti pirmojo eilutėje sutinkamo pliuso bei nukirpti eilutės pradžią iki atrasto pliuso. Pliusas visuomet turėtų būti toje pačioje pozicijoje. Jei nors vienas pliusas yra kitoje pozicijoje — atstumai nevienodi.

Darbas su eilutėmis mokiniams dažnai lengviau suvokiamas nei darbas su masyvais, tad šis uždavinys yra paskirtas abiems amžiaus grupėms.