

Atmintukas

1 sec / 10 sec

1 GB

Atmintukas turi B bitų *flash* atminties. Jame reikia laikyti M bitų kintamojo reikšmę. Flash atmintis turi techninį apribojimą – bitus galima laisvai keisti iš 0 į 1, bet pakeisti bitą iš 1 į 0 galima tik ištrinant visą atmintuko turinį. Flash atmintis trinama dėvisi, todėl prieš trindami norime įrašyti kiek įmanoma daugiau reikšmių.

Raskite efektyvų būdą saugoti kintamąjį flash atmintyje taip, kad visada būtų galima perskaityti dabartinę kintamojo reikšmę. Konkrečiau, jūsų programa turi realizuoti dvi operacijas:

- Įrašyti reikšmę: šiai operacijai pateikiamas dabartinis atmintuko turinys ir nauja reikšmė kurią reikia įrašyti, o jūsų programa turi išvesti atnaujintą atmintuko turinį.
- Perskaityti reikšmę: šiai operacijai pateikiama atmintuko būsena po keleto įrašymo operacijų, o jūsų programa turi išvesti paskutinę į atmintuką įrašytą reikšmę.

Skaitymo ir rašymo operacijos negali tarpusavyje apsikeisti informacija jokių kitu būdu išskyrus atmintuką.

Sąveika. Šis uždavinys yra interaktyvus. Kai jūsų programa paleidžiama, pirmoje įvesties eilutėje pateikiamas sveikasis skaičius T , kur $T = 0$ reiškia, kad jūsų programa turės įrašyti reikšmes į atmintuką, o $T = 1$ reiškia, kad jūsų programa turės skaityti reikšmes iš atmintuko. Antroje eilutėje yra pateikti sveikieji skaičiai B ir M . Tolimesnėse eilutėse aprašomos operacijos.

Pirmoje operaciją aprašančioje eilutėje pateikiamas skaičius C , kur $C = 1$ reiškia, kad toliau bus pateikta rašymo arba skaitymo užklausa (o po jos bus pateikta tolimesnė operacija), o $C = 0$ reiškia, kad jūsų programa turi baigti darbą.

- Kai $T = 0$ ir $C = 1$, antroje eilutėje bus pateiktos dvi tarpu atskirtos simbolių eilutės: B ilgio bitų seka (dabartinis atmintuko turinys), ir M ilgio bitų seka (nauja reikšmė kurią jums reikia įrašyti). Jei jūsų programa gali įrašyti naują reikšmę į atmintuką tik keisdama kai kuriuos bitus iš 0 į 1, tai ji turi išvesti skaičių 1, ir kitoje eilutėje išvesti B bitų ilgio seką – naują atmintuko būseną. Jei jūsų programa įrašyti naujos reikšmės negali, ji turi išvesti skaičių 0.
- Kai $T = 1$ ir $C = 1$, antroje eilutėje bus pateikta B ilgio bitų seka (dabartinis atmintuko turinys). Jūsų programa turi išvesti M ilgio bitų seką – paskutinę reikšmę kuri buvo įrašyta į atmintuką.

Pavyzdys. Pradiniai duomenys Rezultatai

0	
6 2	
1	
111111 00	0
1	
000000 11	1
	110000
0	

Šiame pavyzdyje jūsų programa turi įrašinėti dviejų bitų reikšmes į šešių bitų talpos atmintuką. Pirmą užklausa prašo išsaugoti reikšmę 00, ko jūsų programa atlikti negali. Antra užklausa prašo išsaugoti reikšmę 11, ką jūsų programa gali padaryti. Atkreipkite dėmesį, kad atmintuko turinys pateiktas antrai užklausiai nesutampa su turiniu po pirmos užklauskos.

Pavyzdys. Pradiniai duomenys Rezultatai

1	
6 2	
1	
110000	
	11
1	
110100	
	01
0	

Šiame pavyzdyje jūsų programa turi skaityti dviejų bitų reikšmes iš šešių bitų talpos atmintuko. Pirma užklausa prašo perskaityti reikšmę kai atmintuko turinys yra 110000, ir jūsų programa perskaito reikšmę 11. Antra užklausa prašo perskaityti reikšmę iš atmintuko kurio turinys yra 110100, ir jūsų programa perskaito reikšmę 01.

Pastaba. Išvedus kiekvienos užklaustos atsakymą jūsų programa turi ištuštinti standartinės išvesties buferį (taip pat atkreipkite dėmesį, kad atsakymas turi baigtis naujos eilutės simboliu):

Language	Command
C	<code>fprintf(stdout, "1\n%s\n", s); fflush(stdout);</code>
C++	<code>cout << 1 << "\n" << s << endl;</code>
Java	<code>System.out.println("1"); System.out.println(s); System.out.flush();</code>
Python	<code>sys.stdout.write("1\n{}\n".format(s)) sys.stdout.flush()</code>

Testavimas sistemoje. Kiekvienam testui vienu metu bus paleistos 4 jūsų programos kopijos: 2 rašymui ir 2 skaitymui. Vykdyimo laiko ir atminties ribojimai skaičiuojami visoms 4 programos kopijoms kartu. **Bet kokie bandymai kitais kanalais nei numatyta sąlygoje perduoti duomenis tarp programos kopijų bus laikomi nesąžiningais.**

Keli B bitų talpos atmintukai bus užpildomi nuliais. Tada kažkokia tvarka su jais bus atliekamos rašymo ir skaitymo operacijos.

Rašymo operacijos metu kažkuri iš rašymui skirtų programos kopijų gaus vieno iš atmintukų turinį ir reikšmę kurią reikia įrašyti. Laikykite, kad rašomos reikšmės buvo parinktos nepriklausomai viena nuo kitos su vienoda tikimybe iš intervalo $0 \dots 2^M - 1$. Jei jūsų programa gali įrašyti pateiktą reikšmę, atmintuko turinys bus pakeista į tokį, kokį gražino jūsų programa. Jei jūsų programa įrašyti reikšmės negali, tas atmintukas nebebus naudojamas jokioms tolimesnėms rašymo operacijoms.

Skaitymo operacijos metu, kažkuri iš skaitymui skirtų programos kopijų gaus vieno iš atmintukų turinį, kuriame jūsų rašymo programa buvo išsaugojusi reikšmę. Vertinimo sistema patikrins ar jūsų programos perskaityta reikšmė yra tokia pati kaip ir paskutinė į atmintuką įrašyta reikšmė. Kiekvienos rašymo operacijos rezultatas skaitymui bus panaudotas lygiai vieną kartą.

Vertinimas. Kiekvienos dalinės užduoties įvertinimas apskaičiuojamas taip: jei jūsų programa į vieną atmintuką sugebės įrašyti vidutiniškai V reikšmių, tai už dalinę užduotį bus skiriama $100 \cdot V/P\%$ taškų, kur P reikšmės yra pateiktos žemiau. Jei jūsų programa gražins neteisingą skaitymo operacijos rezultatą, už dalinę užduotį bus skiriama 0 taškų. Kitų klaidų atveju teste įrašytų reikšmių skaičius bus laikomas 0.

Dalinėms užduotims galioja šie ribojimai:

1. (5 taškai) $B = 16$, $M = 8$, $P = 4.062445024495069624056$.
2. (5 taškai) $B = 32$, $M = 8$, $P = 12.264904841300964834177$.
3. (5 taškai) $B = 32$, $M = 16$, $P = 4.129591513707784802006$.
4. (5 taškai) $B = 64$, $M = 8$, $P = 30.039277894268828900030$.
5. (5 taškai) $B = 64$, $M = 16$, $P = 12.953148094217360432715$.
6. (5 taškai) $B = 64$, $M = 32$, $P = 4.073559788233661501537$.
7. (5 taškai) $B = 128$, $M = 8$, $P = 69.777892228928747548775$.
8. (5 taškai) $B = 128$, $M = 16$, $P = 34.731791275143635240976$.
9. (5 taškai) $B = 128$, $M = 32$, $P = 13.950788987705638908663$.
10. (5 taškai) $B = 128$, $M = 64$, $P = 4.039918210604800133907$.
11. (5 taškai) $B = 256$, $M = 8$, $P = 174.468047086071038511453$.
12. (5 taškai) $B = 256$, $M = 16$, $P = 82.222614151404177334554$.
13. (5 taškai) $B = 256$, $M = 32$, $P = 37.629382269769206488916$.
14. (5 taškai) $B = 256$, $M = 64$, $P = 14.263462282054140577686$.
15. (5 taškai) $B = 256$, $M = 128$, $P = 4.015569093893943430859$.
16. (5 taškai) $B = 512$, $M = 16$, $P = 204.746242127410346170221$.
17. (5 taškai) $B = 512$, $M = 32$, $P = 91.778595148073111539847$.
18. (5 taškai) $B = 512$, $M = 64$, $P = 39.230279242145938712621$.
19. (5 taškai) $B = 512$, $M = 128$, $P = 15.000000002167672268601$.
20. (5 taškai) $B = 512$, $M = 256$, $P = 4.005423277111055468876$.

Be to, visiems testams galioja $N \cdot B \leq 120\,000$, kur N yra didžiausias įrašymo operacijų kurias jūsų programa turės atlikti, skaičius.

Olimpiados metu dalinių užduočių sprendimai bus vertinami su nedideliu testų skaičiumi. Po olimpiados paskutinis jūsų įkeltas sprendimas ir daugiausiai taškų olimpiados metu surinkęs sprendimas bus pertestuoti su didesniu testų skaičiumi. Geresnis iš dviejų pertestavus gautų įvertinimų bus galutinis jūsų šio uždavinio rezultatas. **Olimpiados metu CMS rodomas taškų skaičius nėra jūsų galutinis taškų skaičius už šią užduotį.** Taip daroma siekiant patikslinti jūsų sprendimo įvertinimą. Vertinimas su didesniu testų skaičiumi olimpiados metu būtų per lėtas.