

Návodné úlohy k domácemu kolu OI: kategória A

Toto sú návodné úlohy k domácemu kolu 33. ročníka Olympiády v informatike. Ide teda o sadu ľahších úloh, ktoré tematicky súvisia so súťažnými úlohami. Riešenie týchto úloh môže byť dobrou prípravou na riešenie súťažných úloh. Za riešenia týchto návodných úloh nie sú žiadne body do súťaže.

A-I-1 Trojice

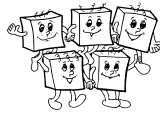
1. Marika má štyri kamienky. Keď sa dnes s nimi hrala, postupne každú dvojicu kamienkov odvážila na kuchynskej váhe. Dostala nasledovné váhy: 60, 80, 100, 120, 140 a 160 gramov. Nepamätá si však už, ktorá váha zodpovedá ktorej dvojici. Koľko vážia Marikine kamienky? Nájdite všetky možné riešenia!
2. Mirko má päť kamienkov. Spravil s nimi to isté ako Marika, dostal teda zoznam 10 rôznych váh dvojíc. Mirkovo dvojča Janko tento zoznam nepozná, ale tvrdí, že keď mu ho Mirko ukáže, tak z neho bude určite vedieť jednoznačne určiť váhy všetkých piatich kamienkov. Ako bude Janko postupovať? Nájdite algoritmus, ktorým túto úlohu aj vy zvládnete vždy vyriešiť.
3. Dané je pole obsahujúce n celých čísel. Napíšte program, ktorý vygeneruje a vypíše všetky možné súčty dvojíc, usporiadané od najmenšieho po najväčší. Program by mal mať časovú zložitosť $O(n^2 \log n)$, ale pozor: pamäťovú zložitosť len $O(n)$.

A-I-2 Telenovela

1. Napíšte program, ktorý načíta vstup v rovnakom formáte ako v súťažnej úlohe a overí, či sa dá postupne v správnom poradí vidieť úplne všetky diely telenovely.
2. Napíšte program, ktorý načíta vstup v rovnakom formáte ako v súťažnej úlohe a overí, či sa dá telenovelu pozrieť tak, že nikdy nevynecháme dva po sebe idúce diely. (Např. pre telenovelu, ktorá má $e = 9$ epizód, môžeme postupne pozrieť diely 1, 2, 4, 6, 7 a 9.)
3. Daný je reťazec S tvorený n veľkými písmenami anglickej abecedy. Napíšte program, ktorý spočíta, koľkými spôsobmi sa dá zmazať presne $n - 5$ písmen tak, aby ostávajúcich 5 tvorilo slovo MACKA. Inými slovami, spočítajte, koľkými rôznymi spôsobmi sa dá slovo MACKA nájsť v reťazci S ako (nie nutne súvislá) podpostupnosť. Príklad: pre $S = \text{MAACKQAZ}$ je odpoveď 2. Dobrá rada: Jedno možné efektívne riešenie okrem iného zodpovie otázku, koľkými spôsobmi sa z prvých 47 písmen reťazca S dá vybrať slovo MAC.

A-I-3 Reverzy prefixov

1. Koľko najmenej operácií *flip* stačí na to, aby sme zaručene dostali najmenší prvok na začiatok poľa?
2. Koľko najmenej operácií *flip* stačí na to, aby sme zaručene dostali najväčší prvok na koniec poľa?
3. Koľko najmenej operácií *flip* stačí na to, aby sme vymenili obsah políčok $A[7]$ a $A[8]$ (pričom zvyšok poľa musí na konci vyzeráť rovnako ako na začiatku)?
4. Nech $t(n)$ je súčet tretích mocnín cifier čísla n . Např. $t(4701) = 4^3 + 7^3 + 0^3 + 1^3 = 408$. Basha si myslí, že keď zoberie ľubovoľné kladné celé číslo n a začne postupne písať čísla $n, f(n), f(f(n)), \dots$, tak sa táto postupnosť časom zacyklí – teda zopakuje sa v nej nejaká hodnota. Miška si naopak myslí, že existuje nejaké n , pre ktoré sa v tejto postupnosti žiadne číslo nikdy nezopakuje.



5. Zvoľme si pevne nejaké kladné celé číslo $m > 1$.

Uvažujme teraz postupnosť zvyškov, ktorú dávajú Fibonacciho čísla po delení m : nech $\forall i : p_i = F_i \bmod m$.

Napr. pre $m = 10$ postupnosť p začína nasledovne: 0, 1, 1, 2, 3, 5, 8, 3, 1, 4, 5, 9, 4, 3, 7, ...

Je pre každé m postupnosť p periodická, či existuje nejaké m , pre ktoré táto postupnosť nemá žiadnu periódu – len sa v nej donekonečna „chaoticky“ menia cifry?

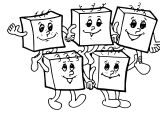
Periodické postupnosti vo všeobecnosti môžu mať predperiódu, ktorá sa ešte neopakuje. Napr. postupnosť 4, 7, 0, 0, 1, 0, 1, 0, 1, ... začína predperiódou 4, 7, 0 a po nej sa už do nekonečna opakuje perióda 0, 1.

Ak je postupnosť p periodická, čo viete povedať o dĺžke jej predperiódy? A prečo je to tak?

(Dobrá rada: Ak netušíte, čo robiť s touto úlohou, skúste si najskôr experimentálne zistiť, ako sa tieto postupnosti správajú. A potom sa zamyslite nad tým, ako je definovaná Fibonacciho postupnosť.)

A-I-4 Stavebnica funkcií

1. Ukážte, že Miška vie vyrobiť úplne ľubovoľnú konštantnú funkciu ľubovoľnej arity – napríklad funkciu so 17 vstupmi, ktorá všetkých 17 vstupov odignoruje a na výstup vždy vráti hodnotu 47.
2. Môžete predpokladať, že už máte zostrojené všetky konštantné funkcie (viď predchádzajúca úloha). Zostrojte unárnu funkciu sgn (signum), pre ktorú platí, že $sgn(0) = 0$ a $\forall x > 0 : sgn(x) = 1$.
3. Z funkcie mul (násobenie, viď súťažnú podúlohu C) zostrojte unárnu funkciu sqr , ktorá pre každé x vráti jeho štvorec, teda hodnotu x^2 .
4. Zostrojte unárnu funkciu f , ktorá pre každé x vráti hodnotu $47x$.
Nájdite dva rôzne spôsoby konštrukcie: jeden bez použitia funkcie mul a druhý s jej použitím.
Pri konštrukcii si dajte dobrý pozor na arity jednotlivých funkcií, ktoré používate.



Návodné úlohy k domácejmu kolu OI: kategória B

Toto sú návodné úlohy k domácejmu kolu 33. ročníka Olympiády v informatike. Ide teda o sadu ľahších úloh, ktoré tematicky súvisia so súťažnými úlohami. Riešenie týchto úloh môže byť dobrou prípravou na riešenie súťažných úloh. Za riešenia týchto návodných úloh nie sú žiadne body do súťaže.

B-I-1 Babkina špajza

1. Napíšte program, ktorý súťažnú úlohu rieši pre $k = 0$, teda v situácii, kedy si Peťka môže zobrať ľubovoľné tri poháre džemu.

Optimálne riešenie beží v lineárnom čase, teda bez toho, aby celé zadané pole usporiadalo.

2. Prvú návodnú úlohu sa dokonca dá riešiť nie len v lineárnom čase, ale navyše v konštantnej pamäti. Existuje teda program, ktorý ju vyrieši a v ktorom používame len konštantne veľa celočíselných premenných (a žiadne polia ani nič podobné). Takýto program postupne po jednom číta zo vstupu objemy džemu v pohároch a každý pohár rovno po prečítaní vhodne spracuje.

Skúste napísať takýto program. (Pomôže vám to aj s prípravou na súťažnú úlohu 3B.)

3. Janko sa snaží Peťke poradiť, ako riešiť pôvodnú súťažnú úlohu: „Podľa mňa by si mala vždy začať tým, že zoberieš najväčší zo všetkých pohárov.“

Má Janko pravdu? Neexistuje taká postupnosť pohárov a také číslo k , že sa Peťke oplatí nechať najväčší pohár v špajzi?

Ak má Janko pravdu, viete to zdôvodniť?

Ak Jankova rada nefunguje, viete ju nejak upraviť aby už fungovala, alebo je takýto prístup principiálne zlý?

4. Na štart pretekov v skoku do diaľky sa postavilo n pretekárov. Postupne každý z nich raz skočil, pričom i -ty skokan pristál vo vzdialenosti d_i . Napíšte program, ktorý načíta postupnosť čísel d_i a vypočíta, koľkokrát sa počas súťaže zmenil vedúci pretekár. Váš program by mal mať lineárnu časovú zložitosť.

Príklad: Ak boli dĺžky skokov postupne 70, 40, 80, 95 a 47, tak sa vedúci pretekár zmenil trikrát. Najskôr sa do vedenia dostal prvý pretekár, potom ho predbehol tretí, no a nakoniec toho predbehol štvrtý.

5. Hovoríme, že skokan do diaľky je *stabilný*, ak jeho výkon už neprekonal nik so skokanov, ktorí skákali po ňom. Teda napríklad ak po svojom skoku bol skokan tretí, stabilný je vtedy, ak ostane tretí až do konca súťaže.

Napíšte program, ktorý načíta vstup v rovnakom formáte ako v predchádzajúcej úlohe a vypočíta a vypíše čísla všetkých stabilných skokanov.

Príklad: Ak boli dĺžky skokov postupne 90, 20, 70, 80 a 47, tak sú stabilní traja skokani: prvý, štvrtý a piaty.

B-I-2 Bicyklový výlet

1. Napíšte program, ktorý bude sledovať počet ľudí v miestnosti. Na vstupe bude tento program dostávať znaky +, - a ?. Znak + znamená, že do miestnosti niekto vošiel, znak - že z nej niekto vyšiel a znak ? znamená, že váš program má vypísať, koľko ľudí je momentálne v miestnosti. (Na začiatku behu programu je miestnosť prázdna.)

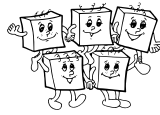
2. Máme pole veľkosti n . V poli sú uložené celé čísla, nie nutne navzájom rôzne. Napíšte program, ktorý nájde číslo, ktoré sa v poli nachádza najviackrát. Napríklad pre pole (10, 20, 10, 47, 47, 10) by váš program mal vypísať číslo 10.

Dobré riešenie tejto úlohy má časovú zložitosť $O(n \log n)$.

3. Kamila študuje na vysokej škole. Tento mesiac by chcela ísť na n prednášok. O každej prednáške vie čas z_i kedy začína a čas k_i kedy končí. (Pre jednoduchosť predpokladajme, že z_i aj k_i sú jednoducho celé čísla – čím väčšie číslo, tým neskorší čas.)

Napíšte program, ktorý zistí, či Kamila stíha všetky prednášky alebo či sa jej niektoré prekrývajú.

Dobré riešenie tejto úlohy má časovú zložitosť $O(n \log n)$.



B-I-3 Dvanásťminútovka

1. Aký je súvis medzi najmenším spoločným násobkom (nsn) a najväčším spoločným deliteľom (nsd) dvoch čísel?
2. Ak poznáme prvočíselný rozklad čísel x a y , ako z nich vieme určiť prvočíselný rozklad ich nsn a nsd?
3. Vyskúšajte si implementovať Euklidov algoritmus počítajúci nsd (viď linka v zadaniach domáceho kola).
Postupne ho spustíte pre všetky dvojice x, y spĺňajúce $1 \leq x, y \leq 1000$. Pre ktorú z nich spraví tento algoritmus najviac krokov?
Viete aj povedať, čím sú tieto čísla x, y zaujímavé? (Rada: pozrite sa aj na priebežné hodnoty, ktoré algoritmus počas behu vypočítal.)
4. Na okruhu behajú dvaja žiaci. Obaja bežia tým istým smerom. V čase 0 boli obaja naraz na štarte. Prvý žiak obehne okruh za čas t_1 , druhý za čas t_2 .
Napíšte program, ktorý načíta čísla t_1, t_2 a t_3 a vypočíta, koľkokrát sa (kdekoľvek na okruhu, nie nutne len na štarte) obaja žiaci na trati stretli počas časového intervalu od 0 po t_3 .

B-I-4 Reálne čísla

1. Ako vyzerá v dvojkovej sústave číslo $22/7$?
2. V dvojkovej sústave začína zápis odmocniny z dvoch nasledovne:
1.01101 01000 00100 11110 01100 11001 11111 10011 10111 10011 00100 10000 10001...
Ak túto hodnotu uložíme do premennej typu float (single precision), aká presne hodnota sa nám do premennej uloží?
Odhadnite tiež veľkosť chyby, ktorá pri tomto zaokrúhlení nastane.
3. V premennej typu double máme uložené nejaké reálne číslo rôzne od nuly. Ako sa zmení obsah premennej (znamienko, mantisa, exponent), keď toto číslo vynásobíme číslom (-8) ?
A ako by sa zmenili, ak by sme dotyčné číslo číslom (-8) vydělili?
4. Uvažujme nasledujúci C++ program:

```
float x = 1;
for (int i=1; i<=100; ++i) {
    x *= 3;
    printf("%f\n", x);
}
```

Program teda v premennej typu float postupne počíta mocniny troch. Niekoľko prvých hodnôt vypíše presne, ale časom to prestane byť pravda. Prečo? Ktorá je posledná presne vypočítaná hodnota?
5. Skôr, než si to vyskúšate v praxi, skúste čo najpresnejšie odhadnúť, čo sa stane, keď v predchádzajúcom programe zmeníme float na double (teda použijeme double precision).