73. ročník Matematické olympiády – 2023/2024

Úlohy ústředního kola kategorie P – 2. soutěžní den

Na řešení úloh máte 4.5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb a přiděleného počítače (tzn. knihy, kalkulačky, mobily, apod.).

Řešením každého příkladu je zdrojový kód programu zapsaný v programovacím jazyce Pascal, C nebo C++. Řešení odevzdáváte pomocí soutěžního systému CMS, který ho automaticky otestuje na připravených sadách testovacích dat. Detaily hodnocení naleznete v letáku s popisem prostředí. Podrobnější informace o testovacích datech najdete na konci zadání každé úlohy.

Experimentální jazyky: V letošním ročníku MO-P je také možné odevzdávat praktické úlohy i v jazycích Python 3, C# 11 a Java 11. U nich však nezaručujeme, že bude možné získat plný počet bodů – je možné, že program nestihne doběhnout do časového limitu, byť používá algoritmus s optimální časovou složitostí.

Zadání naleznete na další straně

P-III-4 Bandasky

Absurdistán tvoří n oáz a mezi nimi $n-1$ obousměrných asfaltek. Oázy jsou očíslovány od 0 do $n-1$. Asfaltky jsou postaveny tak, aby se po nich dalo dostat z každé oázy do každé jiné, silniční síť v Absurdistánu tedy tvoří strom. Oázy propojené asfaltkou nazýváme sousední.

V oázách jsou nějak rozmístěny bandasky s benzínem. Velký vezír Jafar by chtěl dostat alespoň jednu z nich do oázy číslo j. Jafar má dokonalý přehled o tom, kolik bandasek se kde nachází. Vždy, když se v nějaké oáze x nacházejí alespoň dvě bandasky, Jafar si může vybrat sousední oázu y a vydat povel, aby jednu bandasku převezli z x do y. Když Jafar vydá povel, jeden z domorodců vezme svůj moped, jednu bandasku benzínu do něj nalije, druhou na něj naloží a převeze ji do zvolené sousední oázy. V každé oáze žije domorodec s mopedem. Jedna bandaska mu přesně stačí na jednu cestu tam a zpět. Nelze vézt více bandasek najednou.

Soutěžní úloha

Je dána mapa Absurdistánu a číslo oázy j. Najděte nejmenší počet bandasek b takový, že bez ohledu na to, jak jsou na začátku rozmístěné, Jafar vždy dokáže některou bandasku dostat do oázy číslo j (tím, že si bude chytře volit, kdy kterou bandasku kam převézt)

Přesná hodnota b může být velmi velká. Abyste ve svých programech nemuseli pracovat s velkými čísly, stačí nám, když vypočítáte zbytek čísla b po dělení číslem $10^9 + 7$.

Formát vstupu

Na prvním řádku vstupu je číslo n ($1 \leq n \leq 100\,000$) udávající počet oáz. Na druhém řádku je číslo j Jafarovy oázy. Zbytek vstupu tvoří $n-1$ řádků, na každém z nich jsou dvě mezerou oddělená čísla oáz, které jsou spojeny přímou asfaltkou. Je zaručeno, že silniční síť na vstupu tvoří strom.

Formát výstupu

Na výstup vypište jeden řádek a na něm jedno celé číslo: hodnotu ($b \bmod 1\,000\,000\,007$).

Bodování

Je 7 sad vstupů s různými omezeními, za každou, kterou váš program správně vyřeší, dostanete příslušné body:

- V sadě 1 (1 bod) platí $n \leq 6$, $j = 0$ a Absurdistán tvoří po řadě očíslovanou cestu, sousedí tedy oázy 0 a 1, 1 a 2, …, $n-2$ a $n-1$.
- V sadě 2 (1 bod) platí $n \leq 50$ a Absurdistan tvoří po řadě očíslovanou cestu.
- V sadě 3 (1 bod) platí $n \leq 100\,000$ a Absurdistan tvoří cestu, ne nutně po řadě očíslovanou, tj. z každé oázy vedou nanejvýš dvě asfaltky.
- V sadě 4 (1 bod) platí $n \leq 6$.
- V sadě 5 (2 body) platí $n \leq 30$.

- V sadě 6 (2 body) platí $n \leq 3\,000$.
- V sadě 7 (2 body) platí $n \leq 100\,000$.

Jako pomůcku uvádíme, že v sadách 1, 2, 4 a 5 je zaručeno, že se přesná hodnota b vejde do 64-bitové celočíselné proměnné.

Příklady

Vstup:	*Výstup:*
5 0 0 1 0 2 0 3 0 4	5

Pokud jsou jen čtyři bandasky, je možné, že jsou po jedné v oázách 1, 2, 3, 4 a nic nikam nelze převézt. Je-li jich pět, buď je nějaká rovnou v Jafarově oáze 0 (a je hotovo), nebo existuje jiná oáza, kde jsou alespoň dvě (a tedy jednu umíme převézt do oázy 0).

Vstup:	*Výstup:*
5 0 0 1 1 2 2 3 3 4	16

Tento vstup by mohl být v první sadě.

Vstup:	*Výstup:*
7 5 0 1 0 2 0 3 6 3 3 4 4 5	18

P-III-5 Pračka

Podařilo se nám vymyslet novou pračku, která pere mnohem lépe než jiné pračky díky tomu, že dynamicky mění rychlost otáček bubnu během praní. Řešením komplikované soustavy nelineárních diferenciálních rovnic se nám podařilo vypočítat optimální program: je třeba prát n minut (ty si očíslujeme od 0 po $n-1$) a během minuty i je třeba udělat a_i otáček bubnu.

Po postavení prototypu pračky se však zjevil nečekaný problém: ukázalo se, že je nebezpečné rychle zvyšovat otáčky, neboť občas se při tom celý buben utrhne, zdemoluje pračku a rozhází prádlo po širokém okolí. Problém je ale jen se zrychlováním: brzdění bubnu je řešeno jinak, a tak zpomalování tento problém nemá.

Přesněji řečeno, *přílišné zrychlení bubnu* nastává tehdy, když pro nějaké $i < j$ platí, že $(a_j - a_i) > k(j-i)$, kde k je předem známý parametr.

Soutěžní úloha

Do hardwaru pračky už neumíme zasahovat. Pokud tedy chceme předejít vlně reklamací, bude třeba upravit software: změnit posloupnost $a_0, \ldots, a_{n-1}$ na jinou posloupnost $b_0, \ldots, b_{n-1}$, při které nikdy nenastane přílišné zrychlení bubnu. Nové hodnoty b_i mohou být libovolná celá čísla.

Samozřejmě také chceme, aby pračka po úpravě stále co nejlépe prala, takže se nová posloupnost b má co nejvíce podobat původní posloupnosti a. Formálně: pozic i, na kterých platí $a_i \neq b_i$, má být co nejméně.

Formát vstupu

Na prvním řádku vstupu jsou celá čísla n (počet minut programu, $1 \leq n \leq 200\,000$), k (koeficient přílišného zrychlování, $0 \leq k \leq 10^6$) a t (požadovaný typ výstupu, $t \in \{1, 2\}$). Na druhém řádku vstupu jsou kladná celá čísla $a_0, \ldots, a_{n-1}$ (program pračky, $1 \leq a_i \leq 10^9$ pro každé i).

Formát výstupu

Pokud $t = 1$, na výstup vypište jeden řádek a na něm jedno celé číslo p: nejmenší počet pozic, na kterých potřebujeme posloupnost a změnit.

Pokud $t = 2$, přidejte i druhý řádek a na něm jednu možnou posloupnost $b_0, \ldots, b_{n-1}$, která byla vyrobena z $a_0, \ldots, a_{n-1}$ tak, že jsme změnili přesně p hodnot, a při které nikdy nenastane přílišné zrychlení bubnu. Musí platit $1 \leq b_i \leq 10^9$ pro každé i; je zaručeno, že existuje řešení s optimální hodnotou p a s hodnotami b_i v tomto rozsahu.

Bodování

Je 10 sad vstupů s různými omezeními. Za každou, kterou váš program správně vyřeší, dostanete jeden bod. Sady 1 až 5 mají $t = 1$ (stačí vypočítat optimální počet změn). Dodatečná omezení pro tyto sady uvádíme níže. Sady 6 až 10 obsahují tytéž vstupy, jen s $t = 2$ (je potřeba i vypsat jedno řešení).

- V sadách 1 i 6 platí $n \leq 10$.
- V sadách 2 i 7 platí $n \leq 500$ a $k = 0$.

- V sadách 3 i 8 platí $n \leq 200\,000$ a $k = 0$.
- V sadách 4 i 9 platí $n \leq 500$.
- V sadách 5 i 10 platí $n \leq 200\,000$.

Příklady

Vstup:	*Výstup:*
`7 10 1` `40 50 60 70 80 90 100`	`0`

Už vstupní posloupnost je dobrá, není třeba nic měnit.

Vstup:	*Výstup:*
`7 10 2` `40 51 62 73 84 95 106`	`6` `47 53 62 67 69 32 7`

Změnili jsme všechny hodnoty kromě $a_2 = 62$. Existuje i mnoho jiných stejně dobrých řešení.

Vstup:	*Výstup:*
`7 10 2` `40 45 50 7 60 65 70`	`1` `40 45 50 52 60 65 70`

P-III-6 Šachisti

V Absurdistánu žije n šachistů, očíslovaných od 0 do $n-1$. Každý šachista i má svůj *rating* r_i, kladné celé číslo, které říká, jak dobře umí hrát šachy.

Některé dvojice šachistů se přátelí. Takových dvojic je m. Když se dva šachisté přátelí, jsou ochotni se navzájem trénovat. Když šachisté s ratingy x a y spolu trénují, na konci tréninku budou mít oba rating $\max(x, y)$.

Soutěžní úloha

Pro každého šachistu je dán cílový rating c_i. Zjistěte, zda existuje taková posloupnost tréninků, po které budou všichni šachisté mít přesně tyto cílové ratingy.

Formát vstupu

Vstup obsahuje postupně několik nezávislých testů. Na prvním řádku vstupu je číslo t udávající jejich počet.

Na prvním řádku každého testu jsou celá čísla n ($1 \leq n \leq 100\,000$) a m ($0 \leq m \leq \min(150\,000, n(n-1)/2)$). Na druhém řádku jsou čísla $r_0, \ldots, r_{n-1}$ ($1 \leq r_i \leq 100\,000$ pro každé i) a na třetím řádku jsou čísla $c_0, \ldots, c_{n-1}$ ($1 \leq c_i \leq 100\,000$ pro každé i). Zbytek testu pak tvoří m řádků a v každém z nich čísla dvou šachistů, kteří se přátelí. Je zaručeno, že každá neuspořádaná dvojice šachistů se v tomto seznamu objeví nanejvýš jednou.

Formát výstupu

Pro každý test vypište na výstup jeden řádek s textem `ANO`, pokud se cílových ratingů dá dosáhnout, resp. textem `NE`, pokud to nelze.

Bodování

Je deset sad vstupů s různými omezeními. Za každou sadu, kterou váš program správně vyřeší, dostanete jeden bod. Popisy sad a dodatečná omezení uvádíme v tabulce:

sada	$\max t$	$\max n$	m	další omezení
1	100	3	$m \leq 3$	
2	20	50	$m = n(n-1)/2$	každá dvojice šachistů se přátelí a navíc $r_i = i + 1$ pro všechny i
3	20	50	$m = n(n-1)/2$	každá dvojice šachistů se přátelí
4	3	100 000	$m = n - 1$	šachista 0 se přátelí se všemi ostatními
5	20	1000	$m = n - 1$	všechny dvojice $(i, i+1)$ se přátelí
6	3	100 000	$m = n - 1$	všechny dvojice $(i, i+1)$ se přátelí
7	20	1000	$m = n - 1$	graf všech přátelství je strom
8	3	100 000	$m = n - 1$	graf všech přátelství je strom a r_i jsou navzájem různé
9	20	2 500	$m \leq 100\,000$	v každém testu platí $n \cdot m \leq 100\,000$
10	3	100 000	$m \leq 150\,000$	

Příklady

Vstup:	*Výstup:*
2 3 1 900 1000 1100 1100 1000 1100 2 0 3 1 900 1000 1100 1100 1000 1100 1 2	ANO NE

Tento vstup by mohl být v sadě 1. V prvním testu stačí, aby šachisté 0 a 2 spolu trénovali. Ve druhém testu spolu umí trénovat jen šachisté 1 a 2, a to šachistovi 0 rating nezvýší.

Vstup:	*Výstup:*
1 4 3 900 900 1300 1100 1300 1100 1300 1100 0 1 0 2 0 3	ANO

Tento vstup by mohl být v sadě 4. Cílových ratingů umíme dosáhnout následovně: Nejdříve budou trénovat šachisté 0 a 3 (0 si zlepší rating na 1100), poté 0 a 1 (1 si zlepší rating na 1100) a poté 0 a 2 (0 si znovu zlepší rating z 1100 na 1300).

Vstup:	*Výstup:*
2 4 4 1000 1000 2200 1000 2200 1000 2200 1000 0 1 1 2 2 3 3 0 1 0 1001 1000	NE NE