72. ročník Matematické olympiády – 2022/2023

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 a Java 11. U nich nicméně nezaručujeme, že bude možné získat plný počet bodů – je možné, že program nestihne doběhnout do časového limitu, byť používá algoritmus s optimální časovou složitostí.

P-III-4 Stavba dálnic

Ondra nastoupil na ministerstvo dopravy a dostal za úkol dohlížet na stavbu dálnice D42. Takové dohlížení není vůbec jednoduché, je potřeba dodržovat spoustu vyhlášek a vyplňovat všemožné úřední dokumenty a komunikovat s mnoha dalšími úřady.

Dálnici si můžeme představit jako úsečku, která je rozdělená na úseky očíslované 0 až n. Z vedení chodí dokumenty, že se staví úseky dálnice od a po b (včetně). Protože v úředních dokumentech je ale značný nepořádek, stejné úseky dálnice můžou být stavěny klidně vícekrát.

Ondrovi nadřízení chtějí po každém dokumentu vědět, kolik existuje rozestavěných částí dálnice, tedy intervalů po sobě jdoucích úseků, které jsou rozestavěné a na jejichž koncích je buď nerozestavěný úsek, nebo konec dálnice.

Formát vstupu

Na prvním řádku dostanete dvě mezerou oddělená přirozená čísla n a q. Následuje q řádků popisujících dokumenty, na i-tém z nich jsou dvě mezerou oddělená nezáporná celá čísla a_i a b_i značící, že se právě začalo stavět mezi úseky a_i a b_i (včetně úseků a_i a b_i).

Formát výstupu

Na výstupu vypište q řádků, přičemž i-tý z nich bude obsahovat jediné přirozené číslo: počet rozestavěných částí dálnice po zpracování dokumentu číslo i.

Příklady

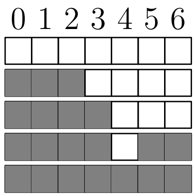
Vstup:

```
6 4
0 2
1 3
5 6
4 4
```

Výstup:

```
1
1
2
1
```

Jak se bude stavba dálnice vyvíjet, najdete na tomto obrázku. Bílé úseky ještě nejsou postavené, zatímco černé už ano:



Bodování

Ve všech vstupech platí $1 \leq n \leq 10^9$, $2 \leq q \leq 2 \cdot 10^5$ a $0 \leq a_i \leq b_i \leq n$.

body	*podmínky*
2	$n, q \leq 10^3$
4	$\forall a_i, b_i : b_i - a_i \leq 10$
4	bez dalších omezení

P-III-5 Malý princ

Osmá planeta byla velmi neobyčejná. Měla $n = 10^5$ vrcholů a každý vrchol byl spojený s právě dvěma jinými vrcholy, takže planeta tvořila cyklus a bydlil na ní fyzik, který zrovna zuřivě mačkal tlačítka na nějaké aparatuře.

„Co děláš?“ zeptal se Malý princ.

„Nedávno jsem četl knihu *Using the Borsuk–Ulam Theorem*, kde psali, že na obyčejných planetách jde vždy najít dvě protilehlá místa, na nichž je stejná teplota,“ odpověděl fyzik. „Snažím se zjistit, jestli to platí i na mé, neobyčejné planetě.“

„A co musíš udělat, abys to zjistil?“ odvětil Malý princ, který nerozuměl.

„Musím změřit teplotu v každém vrcholu,“ řekl fyzik. „Ale tenhle teploměr měří velmi pomalu a mě už nebaví na něj čekat.“

„Pomůžu ti,“ řekl Malý princ, pokynul fyzikovi, aby si na chvíli sedl do stínu, a chopil se přístroje. To je lepší než návštěva u byznysmana, pomyslel si Malý princ, a soustředil se, aby měřil přesně.

Ale když Malý princ měřil pět minut, unavil se jednotvárností činnosti. „Kéž bychom mohli postupovat nějak chytřeji, abychom nemuseli měřit teplotu všude.“

Soutěžní úloha

Zde přichází vaše šance zahrát si malou epizodní roli. Vrcholy planety jsou očíslované $0, \ldots, n-1$ proti směru hodinových ručiček. Pro každé i je ve vrcholu i teplota t_i a platí, že teplota v sousedních vrcholech se liší právě o 1 (je tedy $|t_{n-1} - t_0| = 1$ a $|t_i - t_{i+1}| = 1$ pro $0 \leq i < n-1$). Pomozte Malému princi na co nejméně měření najít $0 \leq i < n/2$ takové, že $t_i = t_{i+n/2}$, anebo zjistit, že takové i neexistuje.

Interakce

Vaším úkolem je napsat program, který bude komunikovat s „Malým princem“ (naším programem). Váš program bude střídavě psát údaje na standardní výstup a číst je ze standardního vstupu.

V každé *iteraci* položíte jednu otázku tak, že vypíšete řádek tvaru `0` x, kde $0 \leq x < n$ je číslo vrcholu, jehož teplotu má Malý princ změřit, a následně přečtete odpověď Malého prince, totiž celé číslo t_x. Vždy bude platit, že $-273 \leq t_x \leq 10^9$.

Pozor na to, že po každé otázce je potřeba vyprázdnit buffer standardního výstupu, aby se vypsaná data ihned dostala k Malému princi:

- V C nebo C++ se `stdio` pomocí `fflush(stdout);`
- V C++ s `iostreamy` pomocí `cout << flush;`
- V Pascalu pomocí `flush(output);`
- V Pythonu stačí funkci `print` přidat argument `flush=True`.
- V Javě vypisujte pomocí `System.out` a pak zavolejte metodu `flush()`.

Až si budete jisti odpovědí, vypište jeden řádek tvaru `1` x, kde buď $0 \leq x < n/2$ je číslo vrcholu takového, že $t_x = t_{x+n/2}$, anebo $x = -1$, pokud žádný takový vrchol neexistuje. Poté váš program musí skončit.

Ukázkový testovač

V adresáři `/mo/public/tasks/princ/` je k dispozici ukázková implementace „Malého prince“, kterou můžete použít při testování svých programů. Když program spustíte příkazem `./testovac`, nejprve přečte ze souboru `planeta.in` popis teplot ve vrcholech planety a potom výše uvedeným způsobem komunikuje pomocí svého standardního vstupu a výstupu. (Pokud chcete číst z jiného souboru, uveďte jeho jméno jako parametr příkazu.)

Soubor `planeta.in` má obsahovat dva řádky. Na prvním je přirozené číslo n popisující počet vrcholů planety. Číslo n musí být sudé (jinak neexistují protější vrcholy) a musí platit $2 \leq n \leq 10^5$. Na druhém řádku je n mezerou oddělených celých čísel mezi -273 a 10^9 takových, že i-té z nich popisuje teplotu ve vrcholu číslo i.

Nyní popíšeme příklad, jak může komunikace programů vypadat. Příslušný soubor `planeta.in` najdete v adresáři s testovačem.

Příklad

V tomto příkladu je $n = 10^5$, všechny vrcholy se sudým číslem mají teplotu 0 a všechny vrcholy s lichým číslem mají teplotu 1. Následuje příklad komunikace Malého prince s vaším programem. Čtěte ho shora dolů.

váš program	*Malý princ*	*komentář*
`0 0`		Jaká je teplota ve vrcholu 0?
	`0`	$t_0 = 0$
`0 1`		Jaká je teplota ve vrcholu 1?
	`1`	$t_1 = 1$
`0 50000`		Jaká je teplota ve vrcholu 50 000?
	`0`	$t_{50\,000} = 0$
`1 0`		Už znám řešení, vrcholy 0 a $0 + n/2 = 50\,000$ mají stejnou teplotu.

Bodování

Program, s nímž budete interagovat („Malý princ“) se nesnaží být zákeřný. V každém testu jsou teploty ve všech vrcholech pevně zvolené a Malý princ vám tyto teploty postupně prozrazuje. Můžete předpokládat, že na každou otázku umí odpovědět v zanedbatelném čase.

Ve všech testovacích vstupech má planeta $n = 10^5$ vrcholů a všechny teploty jsou celá čísla mezi -273 a 10^9. Váš program spustíme na několika testovacích vstupech. Pokud na některém z nich odpoví špatně, získá 0 bodů. Jinak označme jako d největší počet dotazů vašeho programu přes všechny testovací vstupy. Počet bodů, který dostanete, bude záviset na na hodnotě d. Tabulka vpravo popisuje, kolik bodů dostanete, pokud d bude nejvýše daná hodnota.

body	*max. d*
1	100 000
2	50 000
3	26 000
4	5 000
5	2 000
6	1 000
7	350
8	100
9	34
10	32

P-III-6 Bonbónovník

Dan v obchodě objevil *bonbónovník,* tedy strom, který se skládá z bonbónů. Aby strom v obchodě nádherně stál a lákal zákazníky, jsou bonbony spojené hůlkami. Obchod samozřejmě šetřil, strom tedy obsahuje nejmenší možný počet hůlek, a každé dva bonbony jsou tak spojeny právě jednou posloupností hůlek. Formálně si můžeme bonbónovník představit jako strom, jehož vrcholy jsou bonbony a hrany jsou spojovací hůlky.

Dan by si rád na stromu co nejvíce pochutnal. Zjistil ovšem, že strom obsahuje dva druhy bonbonů – lékorky a bonpari. Lékorky má Dan opravdu hodně rád, bonparky naopak moc nemusí. Rád by si tedy odnesl co nejvíce lékorek a a co nejméně bonparek. Přesněji řečeno, chce maximalizovat rozdíl počtu odnesených lékorek a bonparek. Ovšem není to tak jednoduché: Dan už má volnou pouze jednu ruku. Nemůže tedy ze stromu vytrhat všechny lékorky a ty odnést, musí si vzít nějakou část stromu, co drží pospolu.

Na některém z bonbónů je umístěná etiketa, a ten bonbón si tedy Dan musí nutně vzít, aby mohl zaplatit. Dan zatím ještě etiketu neobjevil, a tak by se rád připravil na všechny případy a pro každý bonbón si spočítal, který podstrom jej obsahující je nejlepší.

Soutěžní úloha

Je dán strom s n vrcholy. Každý vrchol obsahuje lékorku nebo bonparku. Chceme pro každý vrchol najít co nejlepší souvislou část stromu, jež daný vrchol obsahuje (jinak řečeno, množinu vrcholů obsahující daný vrchol takovou, že mezi jejími libovolnými dvěma prvky existuje cesta, která obsahuje pouze další prvky dané množiny). Kvalitu částí porovnáváme podle rozdílu počtu lékorek mínus počtu bonparek. Čím větší hodnota rozdílu je, tím lepší část to je. Není potřeba vypisovat, jak přesně část vypadá, stačí určit optimální hodnotu výše zmíněného rozdílu.

Formát vstupu

Na prvním řádku je číslo n – počet vrcholů. Na druhém řádku je posloupnost n znaků popisující postupně vrcholy. Pokud je i-tý znak `L`, tak vrchol i obsahuje lékorku, jinak je tam znak `B` a ve vrcholu je bonparka. Na následujících $n - 1$ řádcích jsou popsané jednotlivé hrany stromu. Na každé řádce je popsaná jedna hrana pomocí dvojice indexů jejich vrcholů. Vrcholy indexujeme od nuly.

Formát výstupu

Vypište n řádků, na i-tém z nich celé číslo popisující optimální rozdíl počtu lékorek mínus počtu bonparek v případě, že je etiketa na i-tém vrcholu.

Příklady

Vstup:	*Výstup:*
6	3
LLLBBL	3
0 1	3
0 2	2
0 3	2
3 4	2
4 5	

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

Bodování

Ve všech vstupech platí $2 \leq n \leq 10^6$.

Označme d maximální počet hůlek vycházejících z jednoho bonbonu (tedy maximální stupeň vrcholu).

body	*podmínky*
1	$n \leq 10$
3	$n \leq 5 \cdot 10^3$
2	$d \leq 2$
2	$d \leq 50$
2	bez dalších omezení