

Na řešení úloh máte 4,5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb (tzn. knihy, kalkulačky, mobily, apod.).

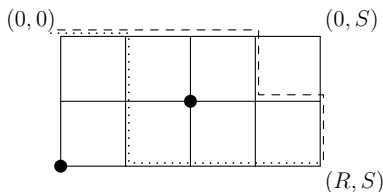
Řešením každého příkladu je zdrojový kód programu zapsaný v programovacím jazyce Pascal, C nebo C++. Po skončení soutěže bude pro každou úlohu váš poslední odevzdaný program automaticky otestován pomocí předem připravených 15 sad testovacích vstupních dat. Každá sada je tvořena jedním nebo několika testovacími vstupy. Jeden bod získáte za každou sadu vstupů, kterou váš program celou správně vyřeší – tzn. pro každý testovací vstup z této sady dá program správný výsledek ve stanoveném časovém a paměťovém limitu. Za každou úlohu tedy můžete získat 0 až 15 bodů.

Sady vstupů jsou navrženy tak, aby každé korektní řešení získalo nějaké body, i když je založeno na neefektivním algoritmu. Časové a paměťové limity zajišťují, že plný počet bodů získá pouze efektivní řešení. Podrobnější informace o testovacích datech najdete na konci zadání každé úlohy.

P-III-4 Mravenci

Program: `mravenci.pas / mravenci.c / mravenci.cpp`
Vstup: `mravenci.in`
Výstup: `mravenci.out`

Právě začíná jedna velmi podivná soutěž. Porota si připravila N stejných mřížek obdélníkového tvaru. Mřížka je tvořena $R + 1$ vodorovnými a $S + 1$ svislými čarami. Každý soutěžící dostane jednu mřížku a několik překážek. Jeho úkolem je rozmístit všechny překážky, které dostal, na různé mřížkové body. Takto upravenou mřížku pak odevzdá porotě.



Příklad mřížky pro $R = 2$ a $S = 4$
 Jsou vyznačeny 2 z 5 cest mravenců.

Porota do levého horního rohu mřížky vypustí speciální druh mravenců. Tito mravenci se pohybují pouze směrem dolů a doprava, a to jen po čarách tvořících mřížku. Mravenci se navíc navzájem nemají rádi, a proto žádní dva nepůjdou úplně stejnou cestou. Do pravého dolního rohu dorazí tudíž přesně tolik mravenců, kolik existuje různých cest mezi levým horním a pravým dolním rohem mřížky. Jelikož

tento počet může být velký, při vyhodnocování soutěže se uvažuje jenom zbytek, který dostaneme po dělení tohoto počtu číslem $10^9 + 9$.

Soutěžní úloha:

Je dáno N mřížek, všechny mají $R + 1$ vodorovných a $S + 1$ svislých čar. Pro každou mřížku je dán počet rozmístěných překážek a jejich souřadnice. Vaším úkolem je určit pro každou mřížku hodnotu $X \bmod 1\,000\,000\,009$, kde X je počet různých způsobů, jimiž se lze dostat z levého horního do pravého dolního rohu příslušné mřížky.

Poznámka: Je možné, že v některých výpočtech bude potřeba používat 64-bitová celá čísla (typ `long long` v C/C++, `int64` v Pascalu).

Formát vstupu:

Na prvním řádku vstupního souboru `mravenci.in` jsou tři kladná celá čísla R , S a N oddělená mezerami. Následuje N popisů jednotlivých mřížek.

Každý popis mřížky začíná řádkem obsahujícím jedno nezáporné celé číslo K , které představuje počet překážek rozmístěných na této mřížce. Další K řádků popisuje polohy překážek. Každý z nich obsahuje dvě mezerou oddělená čísla r_i , s_i ($0 \leq r_i \leq R$, $0 \leq s_i \leq S$), přičemž r_i je souřadnice řádku a s_i souřadnice sloupce, kde leží i -tá překážka. Žádné dvě překážky neleží na stejných souřadnicích a žádná překážka neleží na souřadnicích $(0, 0)$, kde mravenci začínají.

Omezení velikosti proměnných:

V testovacích datech budou proměnné R , S , N a K rovny nejvýše hodnotám uvedeným v následující tabulce:

test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	5	7	10	100	1000	1200	1000	1000	2000	15000	1000	2500	20000	100000	100000
S	5	7	10	100	1000	100000	1000	1000	2000	15000	1000	2500	10000	100000	100000
N	5	7	10	50	50	1	1000	1000	200	50	1000	200	200	2	100
K	5	7	10	15	15	15	2	10	15	10	50	100	50	30	50

Formát výstupu:

Pro každou mřížku vypište do souboru `mravenci.out` jeden řádek s jedním celým číslem: hodnotu $X \bmod 1\,000\,000\,009$, kde X je počet různých způsobů, jimiž lze dojít z levého horního do pravého dolního rohu této mřížky.

Příklad:

Vstup:
 2 4 2
 2
 1 2
 2 0
 2
 0 1
 1 1

Výstup:
 5
 1

První mřížka je zobrazena na obrázku. Ve druhé mřížce vede jediná možná cesta bodem $(2, 0)$.

P-III-5 Hurikán

Program: hurikan.pas / hurikan.c / hurikan.cpp

Vstup: hurikan.in

Výstup: hurikan.out

Souostroví Kiribati je tvořeno N ostrovy. Ještě především byla mezi těmito ostrovy postavena celá síť mostů, po nichž se dalo pohodlně přejít z každého ostrova na libovolný jiný. Včera se ale přehnal přes Kiribati hurikán Hermano a mnohé mosty zničil, takže jich zbylo pouze M . A co je ještě horší, K z těchto M mostů má narušenou statiku. Místní statik zjistil, že v každém z následujících K dní spadne jeden z těchto K poškozených mostů.

Místní vláda potřebuje zajistit, aby se obyvatelé i nadále mohli dostat každý den z libovolného ostrova na libovolný jiný. Rozhodla se proto najmout několik převozníků. Každý převozník zabezpečí dopravu mezi jednou konkrétní přidělenou dvojicí ostrovů.

Soutěžní úloha:

Program dostane na vstupu popis mostů, které zůstaly zachovány po řádění hurikánu, a pořadí, v němž spadnou ty z nich, které jsou poškozeny. Program určí pro dnešek i pro každý z následujících K dní, jaký nejmenší počet převozníků stačí na příslušný den najmout.

Formát vstupu:

První řádek vstupního souboru `hurikan.in` obsahuje dvě celá čísla N a M ($2 \leq N$, $1 \leq M$), která udávají počet ostrovů a počet mostů. Ostrovy jsou očíslovány od 1 do N .

Každý z následujících M řádků popisuje jeden most. Most je určen dvojicí celých čísel $a_i b_i$ ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$) – čísla ostrovů, které daný most spojuje. Mosty si očíslováme od 1 po M v pořadí, v němž jsou zadány na vstupu. Jednu dvojici ostrovů může spojoval i více mostů.

Následuje řádek obsahující celé číslo K ($1 \leq K \leq M$), které představuje počet poškozených mostů. Na posledním řádku vstupu je K mezerami oddělených čísel – čísla poškozených mostů v pořadí, v jakém spadnou.

Omezení velikosti proměnných:

Ve všech testovacích vstupech bude platit $M, N \leq 200\,000$. V sadách testovacích dat 1 až 4 bude platit $N \leq 1\,000$. V sadách 1 až 7 bude platit $K \leq 100$.

Formát výstupu:

Program vypíše do souboru `hurikan.out` $K + 1$ řádků obsahujících vždy jedno celé číslo. Číslo na i -tém řádku výstupu udává minimální počet převozníků, kteří jsou zapotřebí, když spadne prvních $i - 1$ poškozených mostů.

Příklad:*Vstup:*

4 4

1 2

3 2

1 3

3 4

3

2 4 3

Výstup:

0

0

1

2

Dnes je ještě možné přejít po mostech mezi každými dvěma ostrovy. Bude to možné i zítra, až spadne most 3–2. Až pozítří spadne most 3–4, bude již třeba najmout jednoho převozníka, aby nebyl ostrov 4 izolovaný od ostatních. Po pádu mostu 1–3 zůstane stát jediný most 1–2. Tehdy už bude zapotřebí zaměstnávat dva převozníky.