

Ideální město

Leonardo a další italští vědci a umělci se zajímali o plánování měst. Chtěl navrhnout ideální město: pohodlné, prostorné a rozumně využívající dostupné zdroje. Zcela by se tak lišilo od běžných středověkých měst.

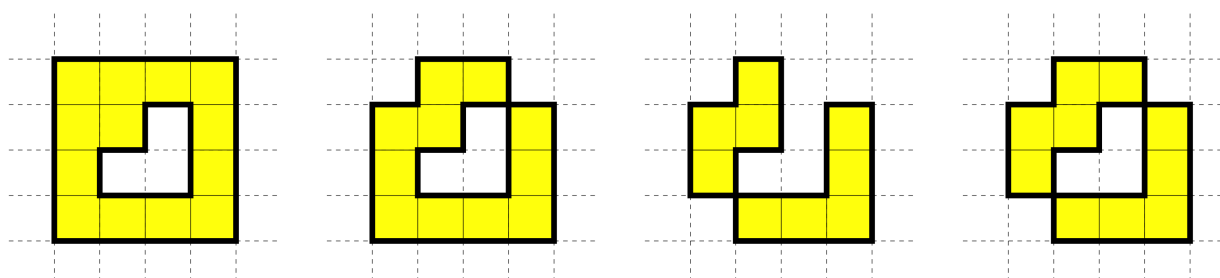
Ideální město

Město je tvořeno N bloky rozmístěnými na nekonečné čtvercové mřížce. Každé políčko mřížky je určeno souřadnicemi ve tvaru (řádek, sloupec). Políčko (i, j) sousedí s políčky $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ a $(i, j + 1)$. Každý blok pokrývá právě jedno políčko. Bloky mohou být pouze na políčkách se souřadnicemi (i, j) splňujícími $1 \leq i, j \leq 2^{31} - 2$. Souřadnice políček budeme používat i pro označení bloků, které jsou na nich umístěny. Dva bloky spolu sousedí, právě když sousedí políčka, na nichž jsou umístěny. V ideálním městě nejsou díry, tedy jeho bloky jsou rozmístěny tak, že platí obě následující podmínky.

- Každá dvě prázdná políčka lze propojit alespoň jednou posloupností sousedících prázdných políček.
- Každá dvě neprázdná políčka lze propojit alespoň jednou posloupností sousedících neprázdných políček.

Příklad 1

Města na následujícím obrázku nejsou ideální. První dvě zleva nesplňují první z podmínek. Třetí město nesplňuje druhou z podmínek. Nejpravější město nesplňuje ani jednu z podmínek.

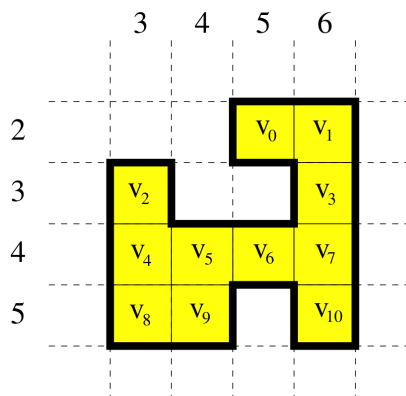


Vzdálenost

Nechť v_0, v_1, \dots, v_{N-1} jsou souřadnice všech N bloků. Vzdálenost $d(v_i, v_j)$ mezi dvěma bloky v_i a v_j je nejmenší počet kroků potřebných pro přesun z v_i do v_j . Krokem rozumíme přesun na sousedící políčko obsahující blok (přes prázdná políčka nelze chodit).

Příklad 2

Ideální město na následujícím obrázku se skládá z $N = 11$ bloků se souřadnicemi $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$ a $v_{10} = (5, 6)$. Vzdálenosti mezi bloky jsou například $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$ a $d(v_9, v_{10}) = 4$.



Zadání

Napište program, který pro zadané ideální město určí součet vzdáleností mezi všemi dvojicemi bloků v_i a v_j , kde $i < j$. Měli byste tedy určit následující součet:

$$\sum d(v_i, v_j), \text{ kde } 0 \leq i < j \leq N - 1$$

Implementujte funkci `DistanceSum(N, X, Y)`, kde X a Y jsou pole velikosti N určující pozice bloků. Pro $0 \leq i \leq N - 1$, blok i má souřadnice $(X[i], Y[i])$ splňující $1 \leq X[i], Y[i] \leq 2^{31} - 2$. Funkce `DistanceSum` by měla vrátit výše specifikovaný součet pro zadané město, a to modulo 1 000 000 000 (miliarda). Skutečná hodnota součtu může být příliš velká a nemusí ji tedy být možné reprezentovat v 32 bitech.

Počet dvojic bloků v příkladu 2 je $11 \times 10 / 2 = 55$. Součet všech vzdáleností dvojic bloků je 174.

Podúloha 1 [11 bodů]

Můžete předpokládat, že $N \leq 200$.

Podúloha 2 [21 bodů]

Můžete předpokládat, že $N \leq 2\,000$.

Podúloha 3 [23 bodů]

Můžete předpokládat, že $N \leq 100\,000$.

Dále pro libovolné dva bloky i a j platí následující podmínka. Jestliže $X[i] = X[j]$ nebo $Y[i] = Y[j]$, pak na každém políčku mezi těmito dvěma bloky (v příslušném řádku či sloupci) je také blok.

Podúloha 4 [45 bodů]

Můžete předpokládat, že $N \leq 100\,000$.

Implementace

Odevzdejte jeden soubor pojmenovaný `city.c`, `city.cpp` nebo `city.pas`. V tomto souboru implementujte výše popsanou funkci s následující deklarací.

C/C++

```
int DistanceSum(int N, int *X, int *Y);
```

Pascal

```
function DistanceSum(N : LongInt; var X, Y : array of LongInt) : LongInt;
```

Můžete samozřejmě implementovat i další pomocné procedury a funkce. Váš program nesmí vypisovat na standardní výstup, číst ze standardního vstupu ani jinak pracovat se soubory.

Ukázkový vyhodnocovač

Vyhodnocovač poskytnutý se zadáním úlohy očekává vstup v následujícím tvaru:

- řádka 1: N ;
- řádky 2, ..., $N + 1$: $X[i]$, $Y[i]$.

Omezení na čas a paměť

- Čas: 1 sekunda.
- Paměť: 256 MiB.