

Sanjsko mesto

Italijanski narodni šport je, seveda, urbanistika in arhitektura. Zato je povsem naravno, da se je poleg premnogih sodobnih mislecev in umetnikov s tem, kakopak, ukvarjal tudi zasanjani Leonardo.

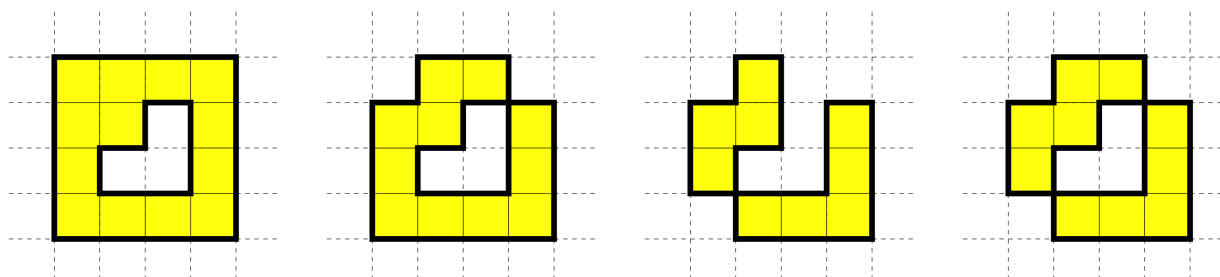
Sanjsko mesto

Mesto je sestavljeno iz N kock, postavljenih na namišljeno neskončno mrežo. Vsako polje te mreže označimo s parom koordinat (vrstica, stolpec). Danemu polju (i, j) so sosednja polja $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ in $(i, j + 1)$. Vsaka kocka, ki jo postavimo na mrežo, zaseda natanko eno polje. Kocko lahko na polje (i, j) postavimo le v primeru, da je $1 \leq i, j \leq 2^{31} - 2$. Koordinate polj bomo uporabljali tudi za označevanje kock na njih. Dve kocki sta sosednji, če sta postavljeni na sosednji polji. V sanjskem mestu so mestne kocke povezane tako, da ne vsebujejo „lukenj“ — to pomeni, da vsa polja zadoščajo spodnjima pogojema.

- Za vsaki dve *prazni* polji obstaja vsaj eno zaporedje sosednjih *praznih* polj, ki ju povezuje.
- Za vsaki dve *neprazni* polji obstaja vsaj eno zaporedje sosednjih *nepraznih* polj, ki ju povezuje.

1. primer

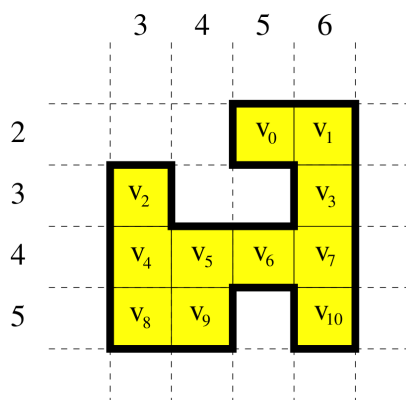
Noben od spodnjih zemljevidov ne predstavlja sanjskega mesta: prva dva (na levi) kršita prvi pogoj, tretji ne zadošča drugemu pogoju, zadnji (desni) pa se požvižga na oba.



Razdalje

Pri sprehajanju skozi mesto bomo prehodu s kocke na sosednjo kocko rekli *skok*. Prazna polja niso mikavna, zato se po njih sploh ne moremo sprehajati. Naj bodo v_0, v_1, \dots, v_{N-1} koordinate na mrežo postavljenih N kock. Za vsaki dve koordinati v_i in v_j je razdalja $d(v_i, v_j)$ med njima definirana kot najmanjše možno število skokov, potrebnih za sprehod od ene do druge.

Zemljevid spodaj predstavlja sanjsko mesto, sestavljeno iz $N = 11$ kock, na koordinatah $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$ in $v_{10} = (5, 6)$. Primeri razdalj: $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$ in $d(v_9, v_{10}) = 4$.



Naloga

Napiši program, ki bo v danem sanjskem mestu izračunal vsoto razdalj med vsemi takimi pari točk v_i in v_j , za katere je $i < j$. S formulo, tvoj program naj izračuna sledečo vsoto:

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

Natančneje, ustvariti moraš funkcijo `DistanceSum(N, X, Y)`, ki za dane N in seznama X in Y , ki opisujeta mesto, izračuna zgornjo formulo. Tako X kot Y imata N elementov; kocka i je na koordinatah $(X[i], Y[i])$, za vsak $0 \leq i \leq N - 1$, kjer $1 \leq X[i], Y[i] \leq 2^{31} - 2$. Ker bo rezultat morda prevelik za 32-bitni zapis, ga vrni po modulu 1 000 000 000 (ena milijarda).

V 2. primeru je $11 \times 10 / 2 = 55$ parov kock. Vsota razdalj med vsemi pari je 174.

1. podnaloga [11 točk]

Privzameš lahko, da je $N \leq 200$.

2. podnaloga [21 točk]

Privzameš lahko, da je $N \leq 2\,000$.

3. podnaloga [23 točk]

Privzameš lahko, da je $N \leq 100\,000$.

Dodatno bosta veljali še ti dve lastnosti: za vsaki dve neprazni polji i in j , pri katerih je $X[i] = X[j]$, bodo neprazna tudi vsa vmesna polja; za vsaki dve neprazni polji i in j , pri katerih je $Y[i] = Y[j]$, bodo tudi vsa vmesna polja neprazna.

4. podnaloga [45 točk]

Privzameš lahko, da je $N \leq 100\,000$.

Podrobnosti implementacije

Oddati moraš natanko eno datoteko, ki mora biti imenovana `city.c`, `city.cpp` ali `city.pas`. V njej mora biti implementacija funkcije, opisane zgoraj, s sledečimi podpisi.

C/C++ programi

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

Pascal programi

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

Ta funkcija se mora obnašati, kot je opisano zgoraj. Seveda lahko v lastno veselje in uporabo načaraš tudi kake druge prikladne funkcije. Oddaje ne smejo na noben način uporabljati standardnega vhoda/izhoda ali kakršnekoli druge datoteke.

Primer ocenjevalca

Nalogi priložen ocenjevalec bo vhodne podatke pričakoval v naslednji obliki:

- vrstica 1: N ;
- vrstice 2, ..., $N + 1$: $X[i]$, $Y[i]$.

Omejitve časa in porabe spomina

- Omejitev časa: 1 sekunda.
- Omejitev spomina: 256 MiB.