

Իդեալ քաղաք

Լեոնարդոն, իր ժամանակի շատ այլ իտալացի գիտնականների և նկարիչների նման, անչափ շատ էր հետաքրքրվում քաղաքների նախագծումով և քաղաքաշինությամբ: Նա նպատակ դրեց նախագծել իդեալ քաղաք՝ հարմարավետ, բնդարձակ և ռեսուրսների ռացիոնալ օգտագործումով, միջին դարերի նեղ քաղաքներից հեռու:

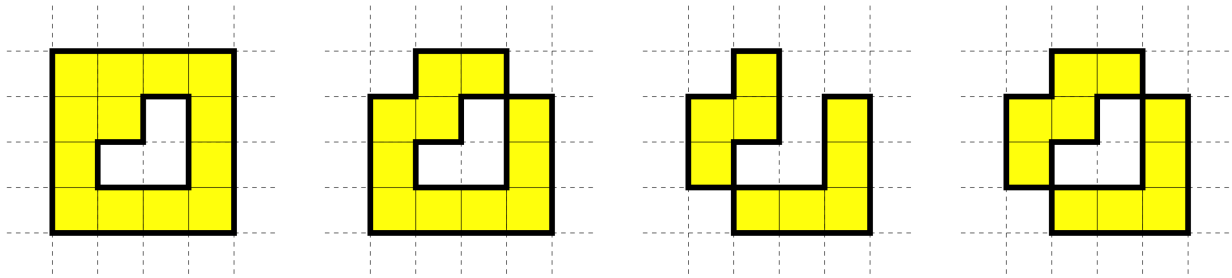
Իդեալ քաղաք

Քաղաքը պատրաստված է N բլուկներից, որոնք տեղադրված են վանդակավոր անվերջ ցանցում: Յուրաքանչյուր վանդակ որոշվում է կոորդինատների զույգով (տող, սյուն): (i, j) վանդակի հարևան վանդակները $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ և $(i, j + 1)$ վանդակներն են: Յուրաքանչյուր բլուկ ցանցի վրա տեղադրելուց ծածկում է ճիշտ մեկ վանդակ: Բլուկը կարող է տեղադրվել (i, j) վանդակում այն և միայն այն դեպքում, եթե $1 \leq i, j \leq 2^{31} - 2$. Մենք բլուկի համար կօգտագործենք այն վանդակի կոորդինատները, որի վրա այն դրված է: Երկու բլուկ հարևան են, եթե նրանք դրված են հարևան վանդակների վրա: Իդեալ քաղաքում բլուկները կապակցված են իրար այնպես, որ քաղաքի սահմաններում "անցքեր" չկան, այսինքն վանդակները պետք է բավարարեն ստորև բերված երկու պայմաններին:

- Ցանկացած երկու "դատարկ" վանդակների համար գոյություն ունի այդ վանդակներն իրար միացնող հարևան *դատարկ* վանդակների առնվազն մեկ հաջորդականություն:
- Ցանկացած երկու *ոչ-դատարկ* վանդակների համար գոյություն ունի այդ վանդակներն իրար միացնող հարևան *ոչ-դատարկ* վանդակների առնվազն մեկ հաջորդականություն:

Օրինակ 1

Ստորև պատկերված բլուկների կոնֆիգուրացիաներից ոչ մեկը իդեալական քաղաք չէ. ձախից առաջին երկուսը չեն բավարարում առաջին պայմանին, երրորդը չի բավարարում երկրորդ պայմանին, և չորրորդը երկու պայմանին էլ չի բավարարում:

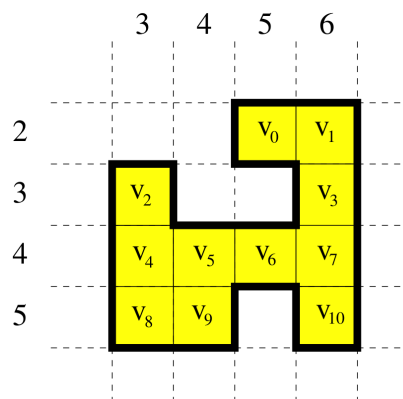


Հեռավորություն

Քաղաքում շրջելիս մի վանդակից մյուսին անցնելը անվանենք *ցատկ*: Դատարկ վանդակների վրայով հնարավոր չէ անցնել: Դիցուք v_0, v_1, \dots, v_{N-1} -ը ցանցում տեղադրված N բլոկների կոորդինատներն են: v_i և v_j երկու տարբեր կոորդինատներով բլոկների համար, նրանց $d(v_i, v_j)$ հեռավորությունը մեկից մյուսին անցնելու համար մինիմալ ցատկերի քանակն է:

Օրինակ 2

Ստորև բերված կոնֆիգուրացիան ներկայացնում է իդեալ քաղաք կազմված $N = 11$ բլոկներից, որոնց կոորդինատներն են $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$, և $v_{10} = (5, 6)$: Օրինակ, $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$, and $d(v_9, v_{10}) = 4$:



Խնդիր

Ձեր խնդիրն է տրված իդեալ քաղաքի համար գրել ծրագիր, որը հաշվում է, բոլոր $i < j$ համար, v_i և v_j բլոկների միջև եղած հեռավորությունների գումարը: Այսինքն, ձեր ծրագիրը պետք է հաշվի հետևյալ գումարը.

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

Մասնավորապես, դուք պետք է իրականացնեք $\text{DistanceSum}(N, X, Y)$ ֆունկցիան, քաղաքը նկարագրող տրված N -ի և երկու X և Y զանգվածների համար հաշվում է վերևի բանաձևը: X -ի և Y -ի երկուսի չափն էլ N է: i բլոկի կոորդինատներն են $(X[i], Y[i])$, $0 \leq i \leq N - 1$, և $1 \leq X[i], Y[i] \leq 2^{31} - 2$: Քանի որ արդյունքը կարող շատ մեծ լինել 32 բիթում պահելու համար, դուք պետք է

տաք այն 1 000 000 000-ի (մեկ միլիարդ) վրա բաժանելուց մնացորդը:

Օրինակ 2-ում կա բլոկների $11 \times 10 / 2 = 55$ գույգ: Բոլոր գույգերի հեռավորությունների գումարը 174 է:

Ենթախնդիր 1 [11 միավոր]

Կարող եք համարել, որ $N \leq 200$:

Ենթախնդիր 2 [21 միավոր]

Կարող եք համարել, որ $N \leq 2\,000$:

Ենթախնդիր 3 [23 միավոր]

Կարող եք համարել, որ $N \leq 100\,000$:

Լրացուցիչ, հետևյալ երկու պայմանները տեղի ունեն. Ցանկացած երկու i և j ոչ-դատարկ վանդակների համար, եթե $X[i] = X[j]$, նրանց միջև եղած բոլոր վանդակները նույնպես ոչ-դատարկ են; ցանկացած երկու i և j ոչ-դատարկ վանդակների համար, եթե $Y[i] = Y[j]$, նրանց միջև եղած բոլոր վանդակները նույնպես ոչ-դատարկ են:

Ենթախնդիր 4 [45 միավոր]

Կարող եք համարել, որ $N \leq 100\,000$:

Իրականացման մանրամասներ

Դուք պետք է submit անեք ճիշտ մեկ ֆայլ, called `city.c`, `city.cpp` կամ `city.pas` անունով: Այդ ֆայլում պետք է իրականացնել վերևում նկարագրված ֆունկցիան, օգտագործելով հետևյալ սիգնատուրը.

C/C++ ծրագրեր

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

Pascal ծրագրեր

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

Այդ ֆունկցիան պետք է իրեն պահի այնպես, ինչպես նկարագրված է վերևում: Իհարկե, դուք կարող եք այլ ֆունկցիաներ ևս իրականացնել ներքին գործածման համար: Ձեր ծրագրերը ստանդարտ մուտք/ելքը չպիտի որևէ կերպ օգտագործեն, նաև չպիտի այլ ֆայլերում փորձեն գրել:

Գրեյդերի օրինակ

Խնդրի միջավայրում տրվող գրեյդերի օրինակը ենթադրում է հետևյալ ձևաչափի մուտք.

- տող 1: N ;
- տողեր 2, ..., $N + 1$: $X[i]$, $Y[i]$.

Ժամանակի և հիշողության սահմանափակումները

- Ժամանակի սահմանափակումը. 1 վայրկյան:
- Հիշողության սահմանափակումը. 256 MiB: