

Լաբիրինթոս

Հերթական էկո-տուրիստական գրասանքի ժամանակ Մարտինը որոշել էր երեխաներին ցույց տալ տեղական եգիպտացորենի լաբիրինթոսը: Չնայած խիստ հրահանգներին՝ ոչ ոք չպետք է հեռանար խմբից, բայց բոլորը հաջողացրել էին կորել:



Մարտինը որոշ չափով կանխագագացել էր, որ նման բան կարող է պատահել, այդ իսկ պատճառով բոլորն ունեին հեռախոսներ և գիտեին միմյանց կոորդինատները: Սակայն դա բավարար չէր: Ոչ բոլորն էին այնքան անվախ, որ հեշտությամբ անցնեին խիտ եգիպտացորենի միջով, նույնիսկ Մարտինը:

Լաբիրինթոսը կարելի է ներկայացնել որպես N տողից և M սյունից բաղկացած վանդակավոր տարածք: Յուրաքանչյուր վանդակ բնորոշվում է իր $A_{i,j}$ խտությամբ, որը բնական թիվ է:

Խմբի յուրաքանչյուր անդամ ունի իր քաջության S_i աստիճանը, որը ցույց է տալիս, որ նա չի անցնի այն վանդակներով, որոնց խտությունը մեծ է S_i -ից:

Լաբիրինթոսում շարժվելը հնարավոր է միայն փոքր արահետներով, որոնք միացնում են հարակից, այսինքն, ընդհանուր կողմ ունեցող, վանդակները: Լաբիրինթոսից դուրս գալու համար բավական է հասնել որևէ եզրային վանդակի:

Եթե երկու տարբեր S_i ունեցող մարդիկ հանդիպեն, ապա պակաս քաջը ձեռք կբերի մյուսի քաջությունը և նրանք կշարժվեն միասին:

Մարտինն արագ հասկացել էր, որ այս կերպ նրանք չեն կարողանա դուրս գալ լաբիրինթոսից: Երկար քննարկումներից հետո նրանք որոշեցին, որ արդար կլինի, եթե յուրաքանչյուրը համարձակվի անցնել այն տարածքներով, որոնց խտությունը առավելագույնը D միավորով մեծ է իր քաջության աստիճանից: Այսինքն՝ մարդը կարող է անցնել այն տարածքներով, որոնց խտությունը **ոչ ավելի է, քան $S_i + D$ թիվը**:

Մարտինը ցանկանում է իմանալ այն **ամենափոքր D -ն**, որի դեպքում բոլորը կկարողանան ապահով դուրս գալ լաբիրինթոսից: Օգնեք նրան՝ գրելով ծրագիր, որը կգտնի անհրաժեշտ ամենափոքր D -ն:

Մուտքային տվյալներ

Առաջին տողում տրվում են երեք բնական թվեր՝ N , M և K , որոնք համապատասխանաբար նշում են լաբիրինթոսի չափերը և խմբի անդամների քանակը:

Հաջորդ N տողերում տրվում են M բնական թվեր, որոնք ցույց են տալիս տարածքների խտությունները:

Վերջին K տողերից յուրաքանչյուրը պարունակում է 3 բնական թիվ՝ X_i , Y_i և S_i , որոնք ցույց են տալիս i -րդ անդամի դիրքը (տող, սյուն) և նրա քաջության սահմանը:

Երաշխավորված է, որ տրված (X_i, Y_i) դիրքի խտությունը չի գերազանցում S_i -ն:

Հաշվարկները սկսվում են 1-ից, և լաբիրինթոսի վերին ձախ անկյունի կոորդինատներն են (1, 1):

Ելքային տվյալներ

Ելքում պետք է արտածել մեկ բնական D թիվ, որը ցույց է տալիս ամենափոքր հնարավոր արժեքը, որով պետք է ավելացնել բոլորի համարձակությունների աստիճանները, որպեսզի բոլորը կարողանան դուրս գալ լաբիրինթոսից:

Հավելում

C++ օգտագործելիս, եթե պահանջվի ռեկուրսիա, մեծացրեք ռեկուրսիայի stack-ի չափը հետևալ ֆունկցիայի միջոցով.

```
#include <sys/resource.h>

void increase_stack_size(size_t size_in_bytes) {
    struct rlimit rl;
    getrlimit(RLIMIT_STACK, &rl);
    rl.rlim_cur = size_in_bytes;
    setrlimit(RLIMIT_STACK, &rl);
}

int main() {
    increase_stack_size(512L * 1024L * 1024L);
    return 0;
}
```

Օրինակ

Մուտք	Ելք
5 10 5	
5 5 5 7 2 2 2 2 2	
5 1 1 1 7 1 1 1 1 2	
5 1 6 6 7 4 5 5 5 5	
5 1 6 1 7 1 5 1 1 2	
5 5 6 5 7 5 5 2 2 2	2
4 6 1	
2 9 2	
4 9 1	
4 4 2	
2 3 4	

Օրինակի պարզաբանումը

4-րդ երեխան, որի կոորդինատները (4, 4) է, շրջապատված է այնպիսի խիտ վանդակներով, որ իր համարձակությունը պետք է 3-ով մեծացնել, որպեսզի նա կարողանա դուրս գալ լաբիրինթոսից: Բայց $D = 2$ դեպքում, 5-րդ երեխան, որը գտնվում է (2, 3) վանդակում, կարող է հասնել նրա վանդակը և օգնել միասին դուրս գալ: Նույն կերպ, $D = 2$ դեպքում, (2, 9) վանդակում գտնվող երեխան կարողանում է հասնել (4, 6) վանդակ, և օգնել 1-ին երեխային դուրս գալ լաբիրինթոսից շարժելով վեր, մինչև (1, 6) վանդակը: Իսկ (4, 9) վանդակում գտնվողը կարողանում է ինքնուրույն դուրս գալ լաբիրինթոսից շարժելով դեպի ներքև կամ աջ:

Սահմանափակումներ

$$3 \leq N, M \leq 1000$$

$$1 \leq K \leq N.M$$

$1 \leq X_i \leq N, 1 \leq Y_i \leq M, (X_i, Y_i) \neq (X_j, Y_j)$, երբ $i \neq j$, այսինքն բոլոր K վանդակներն իրարից տարբեր են

$$1 \leq S_i, A_{i,j} \leq 10^9$$

Ենթախնդիրներ

Ենթախնդիր	Միավոր	N	M	Այլ սահմանափակումներ	Փնտրվող D -ի սահմանը
1	10	≤ 1000	≤ 1000	Բոլոր այն վանդակների խտությունները, որոնցում սկզբում խմբի անդամներից ոչ ոք չկա, իրար հավասար են	-
2	20	≤ 500	≤ 500	$S_1 = S_2 = \dots = S_k$	≤ 200
3	10	≤ 1000	≤ 1000	$S_1 = S_2 = \dots = S_k$	-
4	15	≤ 500	≤ 500	Գոյություն ունի օպտիմալ լուծում, որի ժամանակ ամբողջ խումբը հավաքվում է մի տեղում և դուրս գալիս	≤ 200
5	10	≤ 1000	≤ 1000	Գոյություն ունի օպտիմալ լուծում, որի ժամանակ ամբողջ խումբը հավաքվում է մի տեղում և դուրս գալիս	-
6	20	≤ 1000	≤ 1000	Չկան	≤ 200
7	15	≤ 1000	≤ 1000	Չկան	-