

Ասպետական մրցաշար

Բեատրիչե դը Էստեյի հետ իր ամուսնության համար Միլանի դուքս Լոդովիկո Սֆորցան խնդրեց Լեոնարդոյին կազմակերպել հարսանեական արարողությունը ներառյալ ասպետների մեծ մրցույթը, որը պետք է երեք օր տևեր: Բայց ամենահայտնի ասպետը ուշանում էր...

Մրցաշարը

Մրցաշարի սկզբում N ասպետներ կանգնում են մի գծի վրա, և նրանց դիրքերը համարակալված են 0-ից $N - 1$ թվերով, ըստ շարքում նրանց դասավորման կարգի: Մրցավարը սկսում է հերթական *փուլը* ազդարարելով երկու S և E դիրքեր (որտեղ $0 \leq S < E \leq N - 1$): Բոլոր ասպետները, որոնց դիրքերը S -ի և E -ի մեջն են, ներառյալ նաև իրենց, մրցում են իրար միջև: Հաղթողը շարունակում է մրցաշարը հետ վերադառնալով շարքում իր նախկին տեղը, իսկ բոլոր պարտվածները խաղից դուրս են մնում լքելով ասպարեզը: Խաղի մեջ մնացած բոլոր ասպետները խտանում են պահպանելով իրենց հարաբերական դասավորությունը շարքի մեջ այնպես, որ արդյունքում դիրքերը լինեն 0-ից $N - (E - S) - 1$: Այնուհետև մրցավարը հայտարարում է նոր փուլ ու շարունակում այնքան, մինչև մնա մեկ հաղթող:

Լեոնարդոն գիտի, որ բոլոր ասպետները տարբեր ուժերի են, որ ներկայացվում են 0-ից (ամենաթույլը) մինչև $N - 1$ (ամենաուժեղը) վարկանիշով: Նա նաև ճշգրիտ գիտի, թե մրցավարը C փուլերում ինչ հրամաններ է տալու. ի վերջո Լեոնարդոն է... նաև համոզված է, ու այդ փուլերից յուրաքանչյուրում ավելի բարձր վարկանիշով ասպետը կհաղթի:

Ուշացած ասպետը

N ասպետներից $N-1$ -ն արդեն շարքի են կանգնած, միայն ամենահանրաճանաչը չկա: Այդ ասպետի վարկանիշը R է, և նա մի քիչ կուշանա: Խաղի սրության համար Լեոնարդոն որոշում է օգտվելով ուշացած ասպետի հանրաճանաչությունից նրա համար շարքում այնպիսի տեղ ընտրել, որ մաքսիմիզացնի այն փուլերի քանակը, որոնցում նա կհաղթի: Նկատենք, որ մենք հետաքրքրված չենք այն փուլերով, որոնցում ուշացած ասպետը չի մասնակցում: Հաշվվում են միայն այն փուլերը, որոնցում նա մասնակցում է և հաղթում է:

Օրինակ

$N = 5$ համար , շարքում տեղավորված $N - 1$ ասպետներն ունեն

համապատասխանաբար [1, 0, 2, 4] վարկանիշները: Հետևաբար ուշացած ասպետի վարկանիշը $R = 3$ է: $C = 3$ փուլերի համար մրցավարը մտադիր է հայտարարել (S, E) դիրքերը հետևյալ հերթականությամբ : (1, 3), (0, 1), (0, 1):

Եթե Լեոնարդոն ուշացած ասպետին տեղավորի առաջին դիրքում, ապա շարքի մեջ ասպետների վարկանիշները կլինեն [3, 1, 0, 2, 4]: Առաջին փուլում 1, 2, 3 դիրքերում կհայտնվեն 1, 0, 2 վարկանիշներով ասպետները: Կհաղթի 2 ռանգով ասպետը: Նոր շարքը կլինի [3, 2, 4]: Հաջորդ փուլը կլինի 0 և 1 դիրքերում գտնվող 3 և 2 վարկանիշներով ասպետների միջև: Կհաղթի $R = 3$ վարկանիշով ասպետը և կառաջանա [3, 4] շարքը: Վերջին փուլում կմասնակցեն 0 և 1 դիրքերում գտնվողները: Կհաղթի 4 վարկանիշովը: Այսպիսով ուշացած ասպետը կհաղթի միայն մեկ փուլում (երկրորդում):

Եթե Լեոնարդոն ընտրի այլ տարբերակ, ուշացած ասպետին տեղավորելով 1 և 0 վարկանիշներով ասպետների մեջև, ապա սկզբնական շարքը կունենա հետևյալ տեսքը [1, 3, 0, 2, 4]: Այժմ իրար հետ կմրցեն 3, 0, 2 վարկանիշներով ասպետները և կհաղթի $R = 3$ վարկանիշովը: Ստացված շարքը կունենա հետևյալ տեսքը [1, 3, 4] , և հաջորդ փուլում (1 և 3 վարկանիշներով ասպետների միջև) կհաղթի $R = 3$ վարկանիշովը: Արդյունքում կստացվի [3, 4] շարքը և կհաղթի 4 վարկանիշով ասպետը: Այսպիսով, ուշացած ասպետը կհաղթի երկու փուլերում, և դա ամենահաջող դիրքն է քանի որ նա երկուսից ավելի անգամ հաղթել չի կարող:

Խնդիր

Պահանջվում է գրել ծրագիր, որ ուշացած ասպետի համար շարքում ընտրի լավագույն դիրք այնպես, որ մաքսիմիզացնի այն փուլերի քանակը, որոնցում նա կհաղթի, ինչպես ցանկանում էր Լեոնարդոն: Ավելի ճիշտ դուք պիտի իրականացնեք `GetBestPosition(N, C, R, K, S, E)` անունով պրոցեդուրան, որտեղ

- N -ը ասպետների քանակն է;
- C -ն մրցավարի կողմից նախատեսված անցկացվելիք փուլերի քանակն է ($1 \leq C \leq N - 1$);
- R -ը ուշացած ասպետի վարկանիշն է ; բոլոր ասպետների վարկանիշները (ինչպես ուշացածին, այնպես էլ շարք կանգնածներին) իրարից տարբեր են և ընտրվում են 0, ..., $N - 1$, միջակայքից, և R վարկանիշը բացահայտ տրվում է, չնայած այն կարելի եր հաշվել ;
- K -ն $N - 1$ ամբողջ թվերի զանգված է, որոնք ներկայացնում են արդեն շարքի կանգնած $N - 1$ ասպետների վարկանիշները;
- S -ը և E -ն C չափի երկու զանգվածներ են: ամեն մի i -ի համար 0-ից $C - 1$ -ը ներառյալ, $(i + 1)$ -րդ փուլում մասնակցում են $S[i]$ -ից մինչև n $E[i]$ -ը ներառյալ համարներով ասպետները: Երաշխավորվում է, որ ամեն մի i -ի համար, $S[i] < E[i]$:

Այս ֆունկցիայի պարամետրերը կոռեկտ են. $E[i]$ -ն փոքր է ասպետների ընթացիկ քանակից, որոնք կարող են մասնակցել $(i + 1)$ -րդ փուլում, և բոլոր C հրամաններից հետո մնում է ճիշտ մեկ ասպետ

`GetBestPosition(N, C, R, K, S, E)`-ը պիտի վերադարձնի ամենալավ P դիրքը, որում L եռնարդոն պիտի տեղավորի ուշացած ասպետին ($0 \leq P \leq N - 1$). Եթե կան մի քանի համարժեք դիրքեր, ապա պիտի վերադարձնի *ամենափոքրը* (դիրքի համարը կարելի է որոշել որպես օպտիմալ լուծման մեջ ուշացած ասպետին նախորդող ասպետների քանակ): Մասնավորապես, $P = 0$ նշանակում է, որ ուշացած ասպետը գտնվում է շարքի սկզբում և $P = N - 1$ նշանակում է, որ այն գտնվում է շարքի վեջում:

Ենթախնդիր 1 [17 միավոր]

Ենթադրվում է, որ $N \leq 500$.

Ենթախնդիր 2 [32 միավոր]

Ենթադրվում է, որ $N \leq 5\,000$.

Ենթախնդիր 3 [51 միավոր]

Ենթադրվում է, որ $N \leq 100\,000$.

Իրականացման մանրանասները

Դուք պիտի պատրաստեք մեկ ֆայլ, որը պիտի կոչվի `tournament.c`, `tournament.cpp` կա `tournament.pas`: Այն պիտի իրականացնի վերոհիշյալ պրոցեդուրան, օգտագործելով այսպիսի սիգնատուրներ

C/C++ ծրագրերի համար

```
int GetBestPosition(int N, int C, int R, int *K, int *S, int *E);
```

Pascal ծրագրերի համար

```
function GetBestPosition(N, C, R : LongInt; var K, S, E : array of LongInt) : LongInt;
```

Այդ ֆունկցիաները պետք է իրենց պահեն 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:

Գերյոդերի օրինակ

Խնդրի միջավայրում տրվող գրեյդերի օրինակը ենթադրում է հետևյալ ձևաչափի մուտք.

- Տող 1: N, C, R ;
- Տողեր 2, ..., N : $K[i]$;
- Տողեր $N + 1, \dots, N + C$: $S[i], E[i]$.

Ժամանակի և հիշողության սահմանափակումները

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