

Task 1. Hint

Դոկ Բռաունը կարողացավ վերածել իր DeLorean-ը ժամանակի մեքենայի: Իսկ այժմ նա սկսել է մտածել ավելի մեծ խնդրի վրա. ամենաերկար ընդհանուր ենթահաջորդականությունը: Երբ տրված են երկու A և B թվերի հաջորդականություններ, N և M երկարություններով, նա ցանկանում է գտնել ամենաերկար (կամ ամենաերկարներից մեկը) C հաջորդականությունը, այնպես որ C -ի բոլոր էլեմենտները հանդիպում են և A -ում և B -ում նույն հերթականությամբ (բայց անհրաժեշտ չէ հաջորդական) ինչպես C -ում: Նա կարողացավ գրել ինչ-որ դանդաղ ծրագիր, որը կաշխատի մի քանի օր մինչև կգտնի լուծումը: Սակայն, նրան անհրաժեշտ է պատասխանը ինչքան հնարավոր է շուտ: Նրա սկզբնական գաղափարն էր թողնել, որ իր ծրագիրը աշխատի և հետո մի քանի օրից իր ստացված արդյունքները ուղարկել իր իսկ ներկա ժամանակ: Խնդիրն այն է, որ ժամանակի ճանապարհորդությունը պահանջում է հսկայական էներգիա, ուստի ամբողջական լուծումը ժամանակով հետ ուղարկելը չափազանց թանկ կարծենա:

Հիմա Դոկը ունի նոր գաղափար, բայց նրան անհրաժեշտ է ձեր օգնությունը իրականացնելու համար դա: Նա ցանկանում է ուղարկել լուծման մասին կարճ հուշում ապագայից ներկա և հետո օգտագործելով այդ հուշումը կառուցել օպտիմալ լուծումը: Նկատե՞ք որ անհրաժեշտ չէ, որ օպտիմալ լուծումը լինի նույնը ինչ ապագայի լուծումը:

Դուք պետք է գրե՞ք `hint.cpp` ծրագիր, որը կիրականացնի երկու ֆունկցիաներ. `genHint` և `solve`, որոնք կկատարեն Դոկի գաղափարը:

Իրականացման մանրամասներ

Ձեր `genHint` ֆունկցիան պետք է ունենա հետևյալ նախատիպը.

```
std::vector<bool> genHint(const std::vector<int>& a, const std::vector<int>& b,
    const std::vector<int>& sol);
```

Դա կկանչվի միայն մեկ անգամ և կստանա, որպես արգումենտներ երկու տրված հաջորդականությունները և օպտիմալ լուծումը: Դա պետք է վերադարձնի հուշում, որը կուղարկվի ձեր մյուս ֆունկցիային:

Ձեր `solve` ֆունկցիան պետք է ունենա հետևյալ նախատիպը.

```
std::vector<int> solve(const std::vector<int>& a, const std::vector<int>& b,
    const std::vector<bool>& hint);
```

Դա կկանչվի միայն մեկ անգամ և կստանա, որպես արգումենտներ երկու տրված հաջորդականությունները և հուշումը, որը ձեր մյուս ֆունկցիան վերադարձնում է: Դա պետք է վերադարձնի օպտիմալ լուծումը:

Բացի այս երկու ֆունկցիաներից, ձեր ծրագիրը կարող է ունենալ կամայական տիպի օժանդակ ֆունկցիաներ, կլասսներ, փոփոխականներ և այլն: Սակայն, **չպետք է** պարունակի `main` ֆունկցիան և **պետք է** պարունակի `hint.h` գլխավոր ֆայլը: **Նկատի ունեցե՞ք, որ համակարգում ձեր երկու ֆունկցիաները կանչվելու են ձեր ծրագրի հիման վրա ստեղծված երկու տրաբեր պրոցեսներում:** Սա նշանակում է, որ ձեր երկու ֆունկցիաները չեն կարող ընդհանուր գլոբալ փոփոխական ունենալ, գլոբալ փոփոխականներ չպիտի օգտագործե՞ք:

Լոկալ թեստավորում

Տրված է `Lgrader.cpp` ֆայլը, որը դուք կարող եք կոմպիլացնել ձեր ծրագրի հետ միասին թեստավորելու համար: Դա կկարդա N -ը և M -ը, followed by A and B . Որից հետո դա կկարդա օպտիմալ լուծման

Երկարությունը K -ն և օպտիմալ լուծումը C -ն: Դա կարտածի ձեր hint ֆունկցիայի հուշման երկարությունը, ինչպես նաև ձեր solve ֆունկցիայի վերադարձրած լուծումը: Նկատե՛ք որ, ի տարբերություն գնահատման համակարգի, ձեր լուրջ համակարգը չի աշխատացնի ձեր ֆունկցիաները տարբեր պրոցեսներում:

Սահմանափակումներ

$$1 \leq N, M \leq 10^5, 0 \leq A_i, B_j < (N, M)$$

Ենթախնդիրներ

Ենթախնդիր	Միավորներ	N, M
1	10	$\leq 10^4$
2	90	$\leq 10^5$

Ձեր լուծումը կստանա միավոր ենթախնդրի համար միայն եթե այն անցնի ենթախնդրի բոլոր թեստերը:

Գնահատում

Միավորը, որը դուք կստանաք տրված ենթախնդրի համար կախված է ձեր ծրագրի գեներացրած հուշման ամենաերկար L երկարությունից: Միավորը, որը դուք կստանաք ենթախնդրի համար կլինի.

$$\left(\left(\frac{640}{L+1} \right)^{0.3}, 1 \right)$$