

Kitablar və robot

Zaman limiti: 2 s

Yaddaş limiti: 256 MB

Məktəb kitabxanasında $1, 2, \dots, N$ ədədləri ilə nömrələnmiş N sayda kitab var. Kitablar rəfdə soldan sağa doğru bir cərgədə düzülüb. Lakin kitabların rəfdə hansı sıra ilə düzüldüyü bəlli deyil.

Kitabxanada oxuculara xidmət göstərən ağıllı bir robot var. Bu robota oxumaq istədiyiniz kitabların nömrələrini bildirdikdə (sifariş etdikdə), o, sizə sifariş etdiyiniz kitabları minimum neçə gedişdə götürə biləcəyini deyir. Belə ki, bu robot **bir-birinin ardınca yerləşdirilən kitabları bir gedişdə götürə bilir**.

Məsələn, $N = 5$ kitabdan ibarət rəfdə kitablar 3, 5, 1, 4, 2 sırası ilə düzülərsə və siz 1, 2 və 5 nömrəli kitabları sifariş etsəniz, o, sizə bu kitabları minimum iki gedişdə götürə biləcəyini deyəcək (3, 5, 1, 4, 2 – burada sifariş etdiyiniz kitablar arasında bir-birinin ardınca yerləşdirilmiş olanlar fərqli rənglərlə göstərilmişdir).

Robota ən çox 20000 dəfə sifariş verə bilərsiniz. Hər bir sifarişdə robot sizə bu sifarişdəki kitabları minimum neçə gedişə götürə biləcəyini deyəcək. Bundan istifadə edərək, kitabların rəfdə soldan sağa və ya sağdan sola hansı sıra ilə düzüldüyünü müəyyən edin. Qeyd edək ki, kitabların sırasını tərsinə çevirsək, robotun sorğulara verəcəyi cavablar dəyişməyəcək. Buna görə də soldan sağa və ya sağdan sola hər iki cavab düzgün sayılır. Məsələn, kitablar yuxarıda verilmiş nümunədəki kimi soldan sağa 3, 5, 1, 4, 2 sırasında olarsa, 2, 4, 1, 5, 3 – də düzgün cavabdır.

Onu da qeyd edək ki, bu tapşırıqda robot sifarişləri icra etmir, sadəcə minimum necə gedişə sifariş etdiyiniz kitabları götürə biləcəyini deyir.

İmplementasiya detalları

Bu interaktiv tapşırıqdır. Sizə `"books.cpp"` faylı veriləcək və siz onun içərisindəki `Solve(int N)` funksiyasını icra etməlisiniz.

void Solve(int N); Bu funksiya bir dəfə çağırılacaq. Burada N kitabxanadakı kitabların sayını bildirir. Bu funksiyanın daxilində **Query(S)** funksiyasını ən çoxu 20000 dəfə çağırmaqla bilərsiniz və sonda **Answer(R)** funksiyasını bir dəfə çağırmaqla cavabı bildirməlisiniz.

int Query(const std::vector& S); Bu funksiya robotun sifariş verdiyiniz kitabları minimum neçə gedişdə götürə biləcəyini qaytarır. Burada S , N ölçülü vektordur və sizin sifariş verdiyiniz kitabları bildirir. Hər bir i ($1 \leq i \leq N$) üçün, əgər $S[i - 1] = 0$ olarsa, bu o deməkdir ki, i -ci kitab sifarişinizə daxil deyil, yox əgər $S[i - 1] = 1$ olarsa, bu halda i -ci kitab sifarişinizə

daxildir. Hər bir sorğuda $S[i - 1] = 1$ olan ən azı bir i ($1 \leq i \leq N$) olmalıdır. Yəni ki, boş sifariş vermək olmaz. Boş sifariş verdiyiniz, S vektorunda 0 və 1-dən başqa dəyəərə rast gəldiyi və yaxud S -in ölçüsü N -dən fərqli olduğu təqdirdə sistem tərəfindən **Wrong Answer** nəticəsi alacaqsınız. Bu sorğudan ən çox 20000 dəfə istifadə edə bilərsiniz, daha çox istifadə etdiyiniz təqdirdə sistem tərəfindən **Wrong Answer** nəticəsi alacaqsınız.

void Answer(const std::vector& R); Bu funksiyanı sonda **bir dəfə** çağırmalısınız. Burada R , N ölçülü vektordur və sizin cavabınızı, yəni kitabların rəfdəki sırasını (soldan sağa və ya sağdan sola doğru olması fərq etməz) bildirir. Hər bir i ($1 \leq i \leq N$) üçün, $R[i - 1]$ rəfdə i -ci yerdə duran kitabın nömrəsini bildirir. Əgər cavabınız yanlış olarsa, yaxud R vektorunun ölçüsü N -dən fərqli olarsa sistem tərəfindən **Wrong Answer** nəticəsi alacaqsınız.

Qeydlər:

- Əlavələr bölməsində *books.cpp*, *books.h* və *grader.cpp* faylları verilmişdir. Əgər siz verilən *books.cpp* faylından istifadə etməsəniz, öz yaratdığınız kod faylında *books.h* başlıq faylını daxil etməyi unutmayın, əks halda sistem tərəfindən **Compilation failed** nəticəsi alacaqsınız.
- Proqramınızda əlavə funksiya və ya metodlar yazma, qlobal dəyişənlərdən istifadə edə bilərsiniz.
- Proqramınız standart giriş və çıxış fayllarından istifadə etməməlidir.
- Proqramınız başqa fayllar ilə hər hansı yolla əlaqə yaratmamalıdır.

Kompilyasiya etmək və işlətmək

books.cpp, *books.h* və *grader.cpp* fayllarını eyni qovluğa yerləşdirib aşağıdakı komanda vasitəsilə kompilyasiya edə bilərsiniz:

```
g++ -std=c++11 -o grader grader.cpp books.cpp
```

Kompilyasiya uğurla başa çatarsa, bu zaman *grader* adlı *“exe”* faylı yaranacaq.

Nümunə qreyder

Nümunə qreyder giriş verilənlərini standart girişdən aşağıdakı formatda oxuyur:

- Birinci sətirdə bir tam ədəd, N – kitabxanadakı kitabların sayı verilir.
- Növbəti N sətirin hər birində bir tam ədəd A_1, A_2, \dots, A_N verilir. Burada A_i rəfdə soldan i -ci kitabın nömrəsini bildirir.

Əgər proqramınız doğru olarsa nümunə qreyder **Accepted [sorğu sayı]**, əks halda isə **Wrong Answer** çap edəcəkdir. Burada **[sorğu sayı]** etdiyiniz sorğuların sayını bildirir.

Nümunə interaksiya

Nümunə qreyder standart girişdən $N = 5$ və $A = \{3, 5, 1, 4, 2\}$ oxuyur. Daha sonra $Solve(5)$ çağırılır. Proqramınız $Query(\{1, 1, 0, 0, 1\})$ çağırır və nəticə olaraq 2 qaydır. Sonda proqramınız $Answer(\{3, 5, 1, 4, 2\})$ çağırır və bitir. Qeyd edək ki, proqramınız $Answer(\{2, 4, 1, 5, 3\})$ çağırırsa, bu da düzgün sayılacaq.

Məhdudiyyətlər

N və A_i -nin mənasını anlamaq üçün nümunə qreyder bölməsinə və tapşırığın şərtinə baxın.

- $1 \leq N \leq 1000$
- $1 \leq A_i \leq N$ ($1 \leq i \leq N$)
- $A_i \neq A_j$ ($1 \leq i < j \leq N$)

Alt tapşırıqlar

Bu məsələ aşağıdakı kimi 2 alt tapşırıqdan ibarətdir:

Alt Tapşırıq	Əlavə məhdudiyyətlər	Qiymətləndirmə
1	$N \leq 200$	19 bal
2	Əlavə məhdudiyyət yoxdur	81 bal