

Məsələ İkili axtarış

Giriş faylı `stdin`
Çıxış faylı `stdout`

```
bool binary_search(int n, int p[], int target){
    int left = 1, right = n;
    while(left < right){
        int mid = (left + right) / 2;
        if(p[mid] == target)
            return true;
        else if(p[mid] < target)
            left = mid + 1;
        else
            right = mid - 1;
    }
    if(p[left] == target) return true;
    else return false;
}
```

Aydındır ki, əgər p artan şəkildə sıralanmış olarsa, yuxarıdakı kod yalnız `target` ədədi p 'nin içində varsa `true` qaytaracaq. Amma əgər p sıralanmayıbsa, o zaman bu fakt düzgün olmaya bilər.

Sizə n ədədi və $b_1, \dots, b_n \in \{\text{true}, \text{false}\}$ massivi verilib. $n = 2^k - 1$ şərti həmişə ödənilir (k hər hansı bir ədəd ola bilər). Siz $\{1, \dots, n\}$ ədədlərinin növbəti şərti ödəyən bir permutasiyasını yaratmalısınız. Gəlin $i \in \{1, \dots, n\}$ ədədləri üçün `binary_search(n, p, i)` funksiyasının b_i dəyərini **qaytarmadığı** i 'lərin sayını $S(p)$ ilə göstərək. Siz p tapmalısınız ki, $S(p)$ kiçik olsun (daha ətraflı “Məhdudiyyətlər” bölməsində).

(Qeyd: $\{1, \dots, n\}$ ədədlərinin permutasiyası p bir n uzunluqlu ədədlər ardıcılığıdır ki, 1 'dən n 'ə qədər bütün tam ədədlər bu ardıcılığın içində tam olaraq bir dəfə olsun.)

Giriş verilənləri

Girişə bir neçə test verilir. İlk sətirdə testlərin sayını göstərən T ədədi var. Sonra testlər gəlir. Testlərin ilk sətirində n ədədi var. İkinci sətirdə n uzunluqlu 0 və 1 'lərdən ibarət string gəlir. Bu rəqəmlər arasında boşluq yoxdur. i 'ci hərf 1 olarsa $b_i = \text{true}$, əks halda $b_i = \text{false}$ olur.

Çıxış verilənləri

Çıxışa T testin hər biri üçün tapdığınız cavabı verirsiniz. Test üçün cavab p 'nin bir permutasiyası olur.

Məhdudiyyətlər

- $\sum n$ bütün testlərdəki n dəyərlərinin cəmi olsun
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- Həmişə bir $k \in \mathbb{N}$, $k > 0$ üçün $n = 2^k - 1$.
- əgər alt tapşırığın bütün testləri üçün $S(p) \leq 1$ olarsa, o zaman həmin alt tapşırığın sizə verdiyi xalın 100%-ni qazanırsınız
- əks halda əgər alt tapşırığın bütün testləri üçün $S(p) \leq \lceil \log_2 n \rceil$ (yəni $2^{S(p)} \leq n + 1$) olarsa, o zaman həmin alt tapşırığın sizə verdiyi xalın 50%-ni qazanırsınız

#	Xal	Məhdudiyyətlər
1	3	$b_i = \text{true}$.
2	4	$b_i = \text{false}$.
3	16	$1 \leq n \leq 7$.
4	25	$1 \leq n \leq 15$.
5	22	$n = 2^{16} - 1$ və b_i dəyərləri bir-birindən asılı olmadan $\{\text{true}, \text{false}\}$ arasından bərabər təsadüfi seçilib.
6	30	lavə məhdudiyyət yoxdur.

Nümunə

Giriş faylı	Çıxış faylı
4 3 111 7 1111111 3 000 7 000000000	1 2 3 1 2 3 4 5 6 7 3 2 1 7 6 5 4 3 2 1
2 3 010 7 0010110	3 2 1 7 3 1 5 2 4 6

İzah

Nümunə 1. İlk iki testdə $S(p) = 0$ olur.

Üçüncü testdə $S(p) = 1$ olur, çünki `binary_search(n, p, 2)` bizə `true` qaytarır, amma $b_2 = \text{false}$.

Dördüncü testdə $S(p) = 1$ olur, çünki `binary_search(n, p, 4)` bizə `true` qaytarır, amma $b_4 = \text{false}$.

Nümunə 2. Hər iki test üçün $S(p) = 0$ olur.