

Problem BinSearch

Girdi dosyası `stdin`
Çıktı dosyası `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;  
}
```

Çok iyi bilindiği üzere, p dizisinin sıralı olduğu durumda, verilen kod parçası sadece `target` değerinin p dizisi içinde olması halinde `true` sonucunu verecektir. Ancak, p dizisi sıralı değilse bu doğru olmayabilir.

Size pozitif tamsayı olan n değeri ve bir $b_1, \dots, b_n \in \{\text{true}, \text{false}\}$ serisi veriliyor. Bir pozitif tamsayı olan k için $n = 2^k - 1$ olduğu garanti ediliyor. $\{1, \dots, n\}$ için aşağıdaki şartları sağlayan p permütasyonunu oluşturmanız gerekmektedir. `binary_search(n, p, i)` fonksiyonunun b_i sonucunu **vermediği** pozisyon $i \in \{1, \dots, n\}$ olsun ve $S(p)$ kaç adet böyle pozisyon olduğu bilgisini versin. Siz $S(p)$ değerinin küçük olduğu p değerini vermelisiniz (“Kısıtlar” bölümünde verilen detayları görebilirsiniz)

(Not: $\{1, \dots, n\}$ değerlerinin bir permütasyonu, her bir değer 1 ile n arasında olduğu n tamsayı değerinden oluşan ve her bir değer *kesin olarak sadece* bir defa geçtiği dizi olarak ifade edilebilir.)

Girdi

Girdi birden fazla test içermektedir. İlk satır, verilen girdide kaç adet test olduğunu gösteren T değeridir. Bunu testler izleyecektir.

Her bir test için ilk satır n tamsayı değeri olacaktır. Testin ikinci satırı n uzunluğunda boşlukla ayrılmamış, sadece ‘0’ ve ‘1’ karakterlerinden oluşan bir string olacaktır. Eğer i nci karakter ‘1’ ise bu durumda $b_i = \text{true}$ olacaktır, eğer ‘0’ ise, $b_i = \text{false}$ olacaktır.

Çıktı

Çıktı her bir T testi için cevap bilgisi içerecektir. Her cevap, verilen özel test durumu için üretilebilecek p permütasyonunu içerecektir.

Kısıtlar

- $\sum n$ bir girdide verilen n değerlerinin tamamının toplamı olsun.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$, bazı $k \in \mathbb{N}$, $k > 0$ değerleri için.
- Eğer altgörev içindeki tüm testler için $S(p) \leq 1$ ise, o alt görev için 100% puan verilecektir.
- Eğer altgörev içindeki tüm testler için $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (örneğin, $1 \leq 2^{S(p)} \leq n + 1$) ise, o alt görev için 50% puan verilecektir.

#	Puanlar	Kısıtlar
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	$n = 2^{16} - 1$ ve her b_i , $\{\text{true}, \text{false}\}$ arasından bağımsız ve homojen olarak rastgele seçilmiştir.
6	30	Başka ek kısıt yoktur.

Örnekler

Girdi dosyası	Çıktı dosyası
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

Açıklamalar

Örnek 1. İlk iki test için $S(p) = 0$ olacaktır. Üçüncü testte $S(p) = 1$ olacaktır. Çünkü `binary_search(n, p, 2)` sonucu `true` olacaktır, fakat $b_2 = \text{false}$ durumundadır. Dördüncü testte $S(p) = 1$ olacaktır. Çünkü `binary_search(n, p, 4)` sonucu `true` olacaktır, fakat $b_4 = \text{false}$ durumundadır.

Örnek 2. Her iki test için de $S(p) = 0$ olacaktır.