

Problem BinSearch

Влезна датотека stdin
Излезна датотека 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;
}
```

Добро е познато дека ако p е сортирана (подредена) низа, тогаш овој код враќа true ако и само ако $target$ се појавува во p . Од друга страна, кодот нема да работи како што треба ако p НЕ е сортирана.

Даден е позитивен цел број n и низа $b_1, \dots, b_n \in \{true, false\}$. Се гарантира дека $n = 2^k - 1$ за некој позитивен цел број k . Треба да генерирате пермутација p на $\{1, \dots, n\}$ којашто задоволува одредени услови. Нека $S(p)$ е бројот на индекси $i \in \{1, \dots, n\}$ за кои $binary_search(n, p, i)$ НЕ го враќа b_i . Треба да го поставите p така што $S(p)$ е мал број (како што е подетално објаснето во делот “Restrictions”).

(Забелешка: пермутација на $\{1, \dots, n\}$ е низа од n цели броеви којашто го содржи секој од целите броеви од 1 до n *точно* по еднаш.)

Input data

Влезот содржи повеќе тест случаи. Првата линија од влезот го содржи T , бројот на тест случаи. Тест случаите следуваат во продолжение.

Првата линија од секој тест случај го содржи целиот број n . Втората линија од секој тест случај содржи стринг со должина n кој ги содржи само знаците '0' и '1'. Овие знаци НЕ се разделени со празни места. Ако $i^{\text{тиот}}$ знак е '1', тогаш $b_i = true$, а ако е '0', тогаш $b_i = false$.

Output data

Излезот се состои од одговорите за секој од T -те тест случаи. Одговорот за конкретен тест случај се состои од пермутацијата p генерирана за тој тест случај.

Restrictions

- Нека $\sum n$ е збирот од сите вредности за n во еден влез.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$ за некое $k \in \mathbb{N}$, $k > 0$.
- Ако $S(p) \leq 1$ за сите тест случаи во рамките на една подзадача, тогаш ви се доделуваат 100% од поените за таа подзадача.
- Инаку, ако $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (т.е. $1 \leq 2^{S(p)} \leq n + 1$) за сите тест случаи во рамките на една подзадача, тогаш ви се доделуваат 50% од поените за таа подзадача.

#	Поени	Restrictions
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	$n = 2^{16} - 1$ и секое b_i се избира рамномерно и независно на random од $\{\text{true}, \text{false}\}$.
6	30	Нема дополнителни ограничувања.

Primeri

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

Explanations

Пример 1. Во првите два тест случаи од првиот пример, имаме $S(p) = 0$.

Во третиот тест случај, имаме $S(p) = 1$. Ова е така затоа што `binary_search(n, p, 2)` враќа `true`, иако $b_2 = \text{false}$.

Во четвртиот тест случај, имаме $S(p) = 1$. Ова е така затоа што `binary_search(n, p, 4)` враќа `true`, иако $b_4 = \text{false}$.

Пример 2. Имам $S(p) = 0$ за двата тест случаи.