Zadatak BinSearch

Ulazna datoteka	`stdin`
Izlazna datoteka	`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;
}
```

Dobro je poznato da ako je `p` već sortiran onda će ovaj code vratiti `true` ako i samo ako se `target` nalazi negdje u `p`. S druge strane, to ne mora uvijek biti slučaj ako `p` nije već sortiran.

Dat je cijeli pozitivan broj n i niz $b_1, \ldots, b_n \in \{\texttt{true}, \texttt{false}\}$. Sigurno je da uvijek vrijedi $n = 2^k - 1$ za neki pozitivan cijeli broj k. Vaš zadatak je da generišete jednu permutaciju p skupa $\{1, \ldots, n\}$ koja poštuju odredjene uslove. Neka je $S(p)$ ukupan broj indeksa $i \in \{1, \ldots, n\}$ za koje `binary_search(n, p, i)` **ne** vraća b_i. Trebate izabrati p tako da je $S(p)$ mali (vidjeti detaljno objašnjenje u dijelu pod "Ograničenja").

(Važno: jedna permutacija skupa $\{1, \ldots, n\}$ je niz od n cijelih brojeva koji sadrži svaki broj od 1 do n *tačno* jednom.)

Ulazni podaci

Ulaz sadrži više testnih slučajeva. Prva linija sadrži broj T koji odredjuje broj testnih slučajeva. Testni slučajevi su definisani u narednim linijama.

Prva linija jednog testnog slučaja sadrži cijeli broj n. Druga linija ovog testnog slučaja sadrži jedan string dužine n koji sadrži samo '0' i '1' karaktere. Ovi karakteri nisu odvojeni razmakom. Ako je i^{ti} karakter '1' onda je $b_i = \texttt{true}$, a ako je i^{ti} karakter '0', onda je $b_i = \texttt{false}$.

Izlazni podaci

Izlazni podaci se sastoje od odgovora na svaki od T testnih slučajeva. Odgovor za odredjeni testni slučaj se sastoji od nadjene permutacije p za taj testni slučaj.

Ograničenja

- Neka je $\sum n$ svih vrijednosti od n u jednom ulazu.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$ za neki $k \in \mathbb{N}$, $k > 0$.
- Ako vrijedi $S(p) \leq 1$ za sve testne slučajeve u tom podzadatku onda ćete dobiti svih 100% bodova za taj podzadatak.
- U suprotnom i ako je $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (i.e. $1 \leq 2^{S(p)} \leq n + 1$) za sve testne slučajeve u tom podzadatku onda ćete dobiti pola, odnosno 50%, od svih bodova za taj podzadatak.

#	Bodovi	Ograničenja
1	3	$b_i =$ `true`.
2	4	$b_i =$ `false`.
3	16	$1 \leq n \leq 7$.
4	25	$1 \leq n \leq 15$.
5	22	$n = 2^{16} - 1$ i svaki b_i je izabran na slučajan način, uniformno i nezavisno, iz skupa {`true`, `false`}
6	30	Nema dodatnih ograničenja.

Primjeri

Ulazna datoteka	Izlazna datoteka
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

Objašnjenje

Primjer 1. U prva dva testna slučaja u prvom primjeru imamo da je $S(p) = 0$.

U trećem slučaju imamo da je $S(p) = 1$. Ovo je zbog toga što `binary_search(n, p, 2)` vraća `true`, iako je $b_2 =$ `false`.

U četvrtom testnom slučaju imamo da je $S(p) = 1$. Ovo je zbog toga što `binary_search(n, p, 4)` vraća `true` iako je $b_4 =$ `false`.

Primjer 2. Imamo da je $S(p) = 0$ za oba testna slučaja.