Binārā meklēšana (BinSearch)

Ievaddati	`stdin`
Izvaddati	`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;
}
```

Ir skaidri redzams, ka, ja `p` ir sakārtots, tad šis kods atgriež `true` tad un tikai tad, ja `target` atrodas `p`. No otras puses, tas tā var nebūt, ja `p` nav sakārtots.

Ir dots naturāls skaitlis n un virkne $b_1, \ldots, b_n \in \{\texttt{true}, \texttt{false}\}$. Tiek garantēts, ka $n = 2^k - 1$ un k ir naturāls skaitlis. Uzdevums ir ģenerēt permutāciju p no $\{1, \ldots, n\}$, kas atbilst noteiktiem nosacījumiem. Ar $S(p)$ tiek apzīmēts indeksu skaits $i \in \{1, \ldots, n\}$, kuram `binary_search(n, p, i)` **neatgriež** b_i. Ir jāatrod tāda permutācija p, lai $S(p)$ ir mazs (tas ir paskaidrots "Ierobežojumi" sadaļā).

Piezīme. Skaitļu $\{1, \ldots, n\}$ permutācija ir n skaitļu virkne, kurā katrs skaitlis no 1 līdz n ir sastopams *tieši* vienu reizi.

Ievaddati

Ievaddatos ir vairāki testpiemēri. Pirmajā rindā ir testpiemēru skaits T. Nākamajās rindās ir dots testpiemēru apraksts. Katra testpiemēra apraksta pirmajā rindā ir vesels skaitlis n, bet otrajā rindā ir n simbolus gara virkne, kurā ir tikai simboli '0' un '1'. Šie simboli nav atdalīti ar atstarpēm. Ja i-tais simbols virknē ir '1', tad $b_i = \texttt{true}$, un, ja tas ir '0', tad $b_i = \texttt{false}$.

Izvaddati

Izvaddatos ir jābūt T rindām - visu testpiemēru atbildēm. Katrā rindā jābūt permutācijai p, kas ir ģenerēta atbilstošajam testpiemēram.

Ierobežojumi

- Ar $\sum n$ ir apzīmēta viena testa visu n vērtību summa.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$ un $k \in \mathbb{N}$, $k > 0$.
- Ja $S(p) \leq 1$ visiem viena apakšuzdevuma testiem, tad tiek piešķirti 100% no apakšuzdevuma maksimālā punktu skaita.
- Citādi, ja $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (tas ir $1 \leq 2^{S(p)} \leq n + 1$) visiem viena apakšuzdevuma testiem, tad tiek piešķirti 50% no apakšuzdevuma maksimālā punktu skaita.

#	Punkti	Ierobežojumi
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	$n = 2^{16} - 1$ un katrs b_i tiek izvēlēts vienmērīgi, neatkarīgi un nejauši no {`true`, `false`}.
6	30	Bez papildu ierobežojumiem.

Piemēri

Ievaddati	Izvaddati
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

Paskaidrojumi

1. piemērs Pirmajos divos testpiemēros $S(p) = 0$.

Trešajā testpiemērā $S(p) = 1$. Tas ir tādēļ, ka `binary_search(n, p, 2)` atgriež `true`, lai gan $b_2 =$ `false`.

Ceturtajā testpiemērā $S(p) = 1$. Tas ir tādēļ, ka `binary_search(n, p, 4)` atgriež `true`, lai gan $b_4 =$ `false`.

2. piemērs Abos testpiemēros $S(p) = 0$.