

Задача Двоично търсене

Входен файл stdin
Изходен файл 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;
}
```

Добре известно е, че когато p е сортиран масив, горният код връща `true` тогава и само тогава, когато `target` се среща в p . Но когато p не е сортиран, това може да не е вярно.

Дадено е положително цяло число n и редица $b_1, \dots, b_n \in \{\text{true}, \text{false}\}$. Гарантирано е, че $n = 2^k - 1$ за някое положително цяло число k . Вие трябва да генерирате пермутация p на $\{1, \dots, n\}$, която изпълнява дадени изисквания. Нека $S(p)$ е броят на индексите $i \in \{1, \dots, n\}$, за които `binary_search(n, p, i)` **не** връща b_i . Вие трябва да определите p така, че $S(p)$ да е малко (вижте за подробности секцията “Ограничения”).

(Забележка: пермутация на $\{1, \dots, n\}$ е редица от n цели числа, която съдържа всяко число от 1 до n *точно* веднъж.)

Вход

Входът съдържа няколко теста. На първия ред на входа е записан броят T на тестовете. Следват тестовете.

Първият ред за всеки тест съдържа цялото число n . На втория ред е записан низ с дължина n , съдържащ само знаците '0' и '1'. Тези знаци не са отделени с интервали. Ако i -тият знак е '1', тогава $b_i = \text{true}$ и ако той е '0', тогава $b_i = \text{false}$.

Изход

Изходът трябва да съдържа отговорите на всеки от тестовете от входа. Отговорът за всеки тест е пермутацията p , генерирана за този тест.

Ограничения

- Нека $\sum n$ е сумата от всички стойности на n във входния файл.
- $1 \leq \sum n \leq 100\,000$.
- $1 \leq T \leq 7\,000$.
- $n = 2^k - 1$ за някое $k \in \mathbb{N}$, $k > 0$.
- Ако $S(p) \leq 1$ за всички тестове за една подзадача, тогава вие получавате 100% от точките за тази подзадача.
- В противен случай, ако $0 \leq S(p) \leq \lceil \log_2 n \rceil$ (т.е. $1 \leq 2^{S(p)} \leq n + 1$) за всички тестове за една подзадача, тогава вие получавате 50% от точките за тази подзадача.

Подзадачи

#	Точки	Ограничения
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 всяко b_i е избрано случайно от $\{\text{true}, \text{false}\}$.
6	30	Няма допълнителни ограничения.

Примери

Входен файл	Изходен файл
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

Обяснения

Пример 1. За първите два теста имаме $S(p) = 0$.

За третия тест имаме $S(p) = 1$, защото `binary_search(n, p, 2)` връща `true`, но $b_2 = \text{false}$.

За четвъртия тест имаме $S(p) = 1$, защото `binary_search(n, p, 4)` връща `true`, но $b_4 = \text{false}$.

Пример 2. Имам $S(p) = 0$ и за двата теста.