

Scarnum Solution

Subtank1: Thuật toán có độ phức tạp $O(n*m)$

Đây cũng là thuật toán đơn giản, “đề bảo sao thì mình làm vậy”.

Đầu tiên với mỗi truy vấn thì chi phí là $O(m)$, với mỗi số n thì ta duyệt từ $I = 2 \rightarrow n$ và kiểm tra xem:

- I có phải là ước của n hay không
- Nếu I là ước của n , kiểm tra xem nó có phải là số may mắn hay không. Nếu I là số may mắn thì ghi ra “YES”. Nếu không, tăng I lên và kiểm tra tiếp
- Nếu chạy hết vòng lặp mà vẫn không có I thỏa mãn yêu cầu, ghi ra “NO”

Thuật toán kiểm tra:

```
For( $I = 2 \rightarrow n$ )
```

```
If( $n \bmod I = 0$  and  $luckynum(i) = true$ ) write(“YES”), break;
```

```
If( $I > n$ ) write(“NO”);
```

Hàm kiểm tra số I có phải là số may mắn không $lucknum()$ các bạn có thể viết như sau:

```
While( $I > 0$ ) {
```

```
If( $I \bmod 10 \neq 2$  and  $I \bmod 10 \neq 5$ ) return false;
```

```
 $I = I \div 10$ ;
```

```
}
```

```
Return true;
```

Độ phức tạp $O(n)$

Subtank2: Ý tưởng tương tự như Subtank1 nhưng lần này cải tiến phần kiểm tra ước của n có phải là số may mắn không chỉ với thuật toán có chi phí $O(\sqrt{n}) \rightarrow$ Chi phí tổng là $O(m*\sqrt{n})$

Chạy một $\text{for}(I = 1 \rightarrow \sqrt{n})$ kiểm tra xem:

- I có phải là ước của n không
- Nếu I là ước của $n \rightarrow n/I$ cũng là ước của n ta chỉ cần kiểm tra xem, hoặc I là số may mắn hoặc i/n là số may mắn. Nếu đúng thì ghi ra “YES” và thoát vòng lặp
- Nếu chạy hết vòng lặp mà không có I nào thỏa mãn ghi ra “NO”

Thuật toán kiểm tra:

```
For( $I = 2 \rightarrow \sqrt{n}$ )
```

```

If( $n \bmod I = 0$ ){
    If( $luckynum(i) = true$  or  $luckynum(n/i) = true$ ) write("YES"), break;
}
If( $I > \sqrt{n}$ ) write("NO");

```

Subtask3: Thuật toán có chi phí $O(m + n \log n)$

Ở Subtask này ta nhận thấy rằng, chi phí $O(m)$ là không thể thay đổi nên ta cần tìm một thuật toán kiểm tra có chi phí $O(1)$

Khai báo mảng `lnum[]` là mảng chứa những số may mắn.

Ta chọn ra trước các số may mắn đưa vào mảng trước như sau bằng cách $\text{for}(I = 2 \rightarrow 10^7)$, với mỗi số I ta kiểm tra xem nó có phải là số may mắn không và đưa vào mảng `lnum` (phần này các bạn có thể tự viết)

Gọi `size` là biến lưu số lượng số may mắn được lưu trong mảng `lnum`. Ta viết một thuật toán tương tự sàng nguyên tố như sau:

```

for( $x = 1 \rightarrow size$ ){
     $i = lnum[x]$ ;
    for( $j = 1 \rightarrow 10^7/i$ )
         $scarnum[i*j] = true$ ;
}

```

Với mảng `scarnum[c]` ý nghĩa rằng c có phải là số gần may mắn hay không, `true` là may mắn, `false` là không phải (nhớ hãy khai báo `scarnum[]` với tất cả các phần tử là `false`)

Thuật toán trên với độ phức tạp xấp xỉ $O(n \log n)$

Với mảng `scarnum[]` ta có thể kiểm tra chỉ với chi phí $O(1)$ → Chi phí tổng là $O(m + n \log n)$