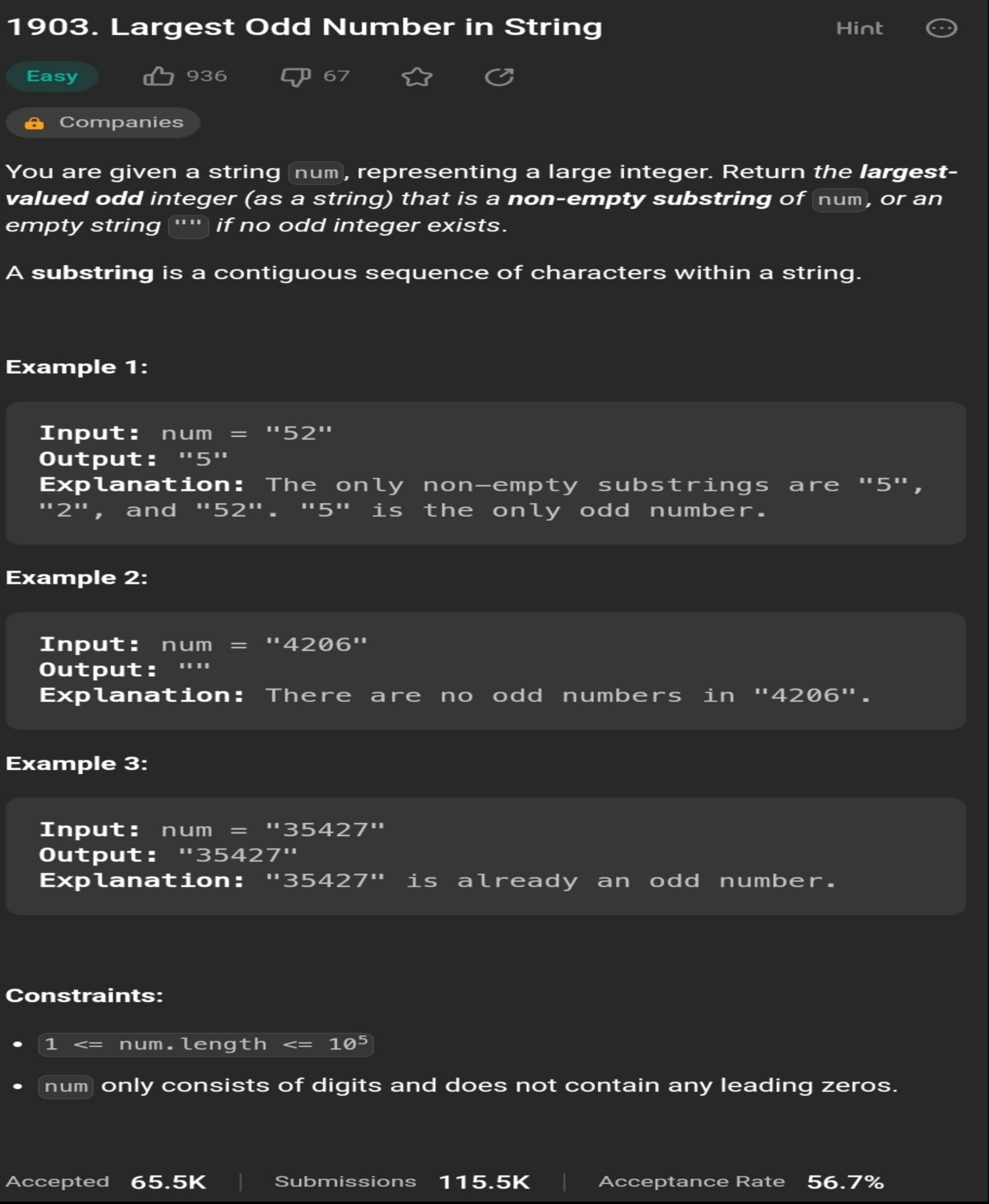


1903. Largest Odd Number in String

You are given a string `num`, representing a large integer. Return *the* ***largest-valued odd*** *integer (as a string) that is a* ***non-empty substring*** *of* `num`*, or an empty string* `""` *if no odd integer exists*.

A **substring** is a contiguous sequence of characters within a string.

Example 1:

```
Input: num = "52"
Output: "5"
Explanation: The only non-empty substrings are "5", "2", and "52". "5" is the only odd number.
```

Example 2:

```
Input: num = "4206"
Output: ""
Explanation: There are no odd numbers in "4206".
```

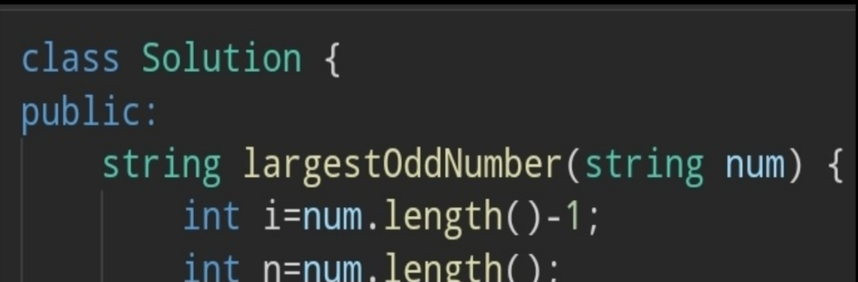
Example 3:

```
Input: num = "35427"
Output: "35427"
Explanation: "35427" is already an odd number.
```

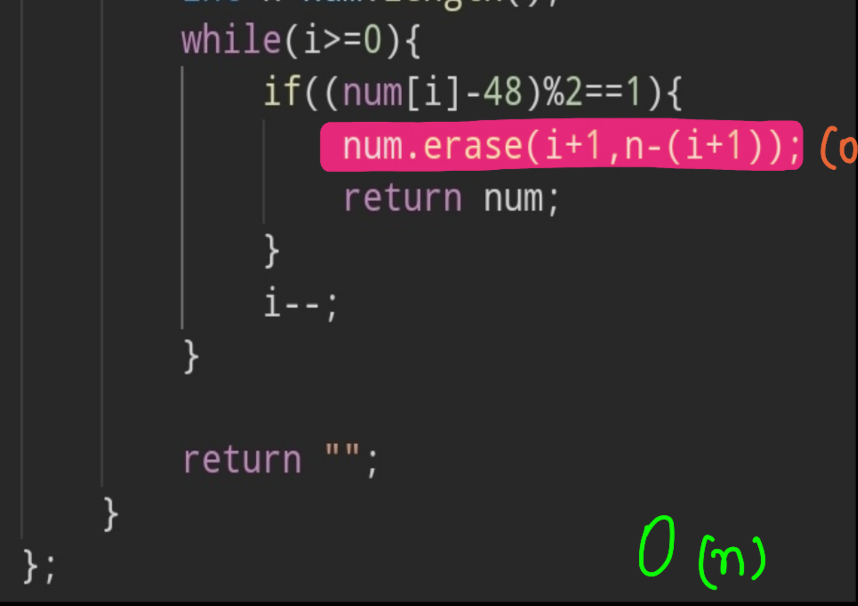
Constraints:

- `1 <= num.length <= 10`^{5}
- `num` only consists of digits and does not contain any leading zeros.

Accepted 65.5K | Submissions 115.5K | Acceptance Rate 56.7%



```
class Solution {
public:
    string largestOddNumber(string num) {
        int i=num.length()-1;
        int n=num.length();
```



```
        int n = num.length();
        while(i>=0){
            if((num[i]-48)%2==1){
                num.erase(i+1,n-(i+1));
                return num;
            }
            i--;
        }

        return "";
    }
};
```

O(n)

(or) num.resize(i+1)

Both functions complexity are O(n) only where n is size of new array.

resize(k) :

The resize function makes the array to be size k starting from index 0 and if k is less than the initial array size then remaining elements are set to '\0'.

$$\{9,8,7,6,5\} \xrightarrow{\text{num.resize}(3)} \{9,8,7\}$$

erase(m, n) :

The 'm' denotes the index from which the elements needs to be erased and 'n' denotes the no. of elements to be erased.

$$\{\overset{0}{9},\overset{1}{8},\overset{2}{7},\overset{3}{6},\overset{4}{5}\} \xrightarrow{\text{num.erase}(3,2)} \{9,8,7\}$$

note:

we can also pop_back if we slightly modify the code.

```
while ( i ≥0)
{
    if((nums[i]-48)%2 ==0)
```

```
            nums.pop_back( )
    else
        return nums

     i--
}
```