

## CS222 Homework 5

Exercises for Algorithm Design and Analysis by Li Jiang, 2016 Autumn Semester

1. Given a positive integer  $n$  and you can do operations as follow:

1. If  $n$  is even, replace  $n$  with  $n/2$ .
2. If  $n$  is odd, you can replace  $n$  with either  $n + 1$  or  $n - 1$ .

What is the minimum number of replacements needed for  $n$  to become 1?

Example 1:

Input: 8

Output: 3

Explanation:  $8 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1$

Example 2:

Input:7

Output:4

Explanation:  $7 \Rightarrow 8 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1$  or  $7 \Rightarrow 6 \Rightarrow 3 \Rightarrow 2 \Rightarrow 1$

Input: int  $n$ ;

Output: int minNum;

2. Implement a basic calculator to evaluate a simple expression string.

The expression string may contain open ( and closing parentheses ), the plus + or minus sign -, non-negative integers and empty spaces.

You may assume that the given expression is always valid.

Some examples:

"1 + 1" = 2

" 2-1 + 2 " = 3

"(1+(4+5+2)-3)+(6+8)" = 23

Note: Do not use the eval built-in library function.

Input: string str;

Output: int result;

3. In the computer world, use restricted resource you have to generate maximum benefit is what we always want to pursue.

For now, suppose you are a dominator of  $m$  0s and  $n$  1s respectively. On the other hand, there is an array with strings consisting of only 0s and 1s.

Now your task is to find the maximum number of strings that you can form with given  $m$  0s and  $n$  1s. Each 0 and 1 can be used at most once.

Note:

1. The given numbers of 0s and 1s will both not exceed 100.
2. The size of given string array won't exceed 600.

Example 1:

Input: Array = "10", "0001", "111001", "1", "0",  $m = 5$ ,  $n = 3$ .

Output: 4.

Explanation: This are totally 4 strings can be formed by the using of 5 0s and 3 1s, which are "10," "0001", "1", "0".

Example 2:

Input: Array = "10", "0", "1",  $m = 1$ ,  $n = 1$ .

Output: 2.

Explanation: You could form "10", but then you'd have nothing left. Better form "0" and "1".

Input: string array[]; int N; int m; int n; // N: number of strings in array.

Output: int findMaxForm;