

Optimizer Bug

Time Limit: 5s

Your supervisor is completely in love with neural nets. He has discovered a magical optimizer which somehow works for any kind of problem one can throw at it. An optimizer is used to train a neural network by updating the weight variables according to the given input data.

However, he has given you the optimizer and a problem of counting sheep. But no matter how hard you try you can not get it to work on the given dataset. So after digging deeper, you've discovered a weird property of the optimizer. That is, no matter what data you try to train it on, it always follows a simple rule. The rule is so weird that when you explained it to your supervisor he laughed his *** off for straight two and a half minute.

In the meantime, you've come up with a better idea. You've challenged him that you could tell him the updated weights without even running the optimizer.

You've summarized the rules in following pseudocode,

```
optimizerFunction(f1, f2, n) {
    if (n == 1) {
        return f1;
    } else if (n == 2) {
        return f2;
    } else {
        return optimizerFunction(f1, f2, n-1) + 2*optimizerFunction(f1, f2, n-2) + 1;
    }
}

optimize(n, &weights) {
    maxValInd = //get max value & rightmost index of max value from weights.
    minValInd = //get min value & leftmost index of min value from weights.

    // % indicates mod operation
    indU = optimizerFunction(minValInd.index, maxValInd.index, n) % weights.size();
    valU = optimizerFunction(minValInd.value, maxValInd.value, n);
    weights[indU] = (weights[indU] + valU) % 1000000007;
}

main()
{
    n = 1;
    weights = {2, 5, 1, 9, 9};
    for (i = 0; i < n; i++)
    {
        optimize(i+1, weights);
    }
}
```

```
}
```

n, number of iterations.

weights, vector of integer numbers.

You call the **optimize()** method above on every step of the **n** iterations. Note: the **weights** vector is updated in every step.

All the numbers are non negative integers.

All index values are 0 based.

In each test case given the number of iterations **n**, the size of the **weights** array and all the initial values of the **weights**. Find out the summation of all the weights after **n** iterations of the optimizer and output in a single line. There will be no more than **5** test cases. See the sample test cases for more details.

Constraints:

$$1 \leq n \leq 10000$$

$$5 \leq \text{length of weights} \leq 100000$$

$$0 \leq \text{each elements of initial weights} \leq 1000000000$$

Sample Input:

1 5

2 5 1 9 9

3 5

2 5 1 9 9

Sample Output:

27

59