**Files:**

**Editorial**:

This file contains editorial for the problem. Explanation of how the problem can be solved.

**Internal\_notes**:

This file contains notes for us. Like link to github repository, some useful articles etc.

**Problem\_statement**:

This file contains the problem statement.

**Folders:**

**Parsed\_test\_cases\_for\_hackerrank**:

This folder contains test cases for hackerrank. Question comes in our mind why only for\_hackerrank? On hackerrank we need to upload each test case separately like for first test case input000.txt and output000.txt for second test cases input001.txt and output001.txt and so on. And they suggest to use less number of test cases like 25-30 test cases. But in test\_case folder we have generated more test cases so that when we change from hackerrank to our own OJ we need not to create new test cases (basically to avoid rework I have generated more test cases initially). In test\_cases folder we have more test cases like 2-3 test cases in sample\_test\_cases\_input.txt file, 10-15 test cases in handmade\_test\_cases\_input.txt file, 100-150 test cases in small\_test\_cases\_input.txt file, 5-7 test cases in big\_test\_cases\_input.txt file. Also for problem setter it is always useful to have all test cases together, because it will be easy to generate them and test our solution. So, we have all test cases in only 3-4 files. Now we use some code and separate them. So these parsed test cases will be stored in this folder. File names in this folder will be like inputxxx.txt and outputxxx.txt.

**Solutions**:

This folder contains solution (code) for the problem. We have brute\_force\_solution, optimal\_solution, other\_solution files. Optimal\_solution will be our actual solution that needs to be posted for the problem. Brute\_forces solution is written to verify our optimal\_solution. When we have other solution that is sub-optimal etc. then we will write them in other\_solution.

**Test cases**:

This folder contains all the test cases required for this problem. There are many files. Each file has some special meaning and contains some special test cases. Sample\_test\_cases\_input.txt contains sample test cases which are trivial and we keep them visible (students can see the input and output). Number of test cases in sample\_test\_cases\_input.txt should be 2-4. Handmade\_test\_cases\_input.txt contains test cases which are manually created and should include all corner cases. Number of test cases in handmade\_test\_cases\_input.txt should be 10-15. Small\_test\_cases\_input.txt contains test cases which are generated by some code and their size are not too small like sample test cases and also not too big, our aim for these test cases is to run more number of test cases to check student's solution. Number of test cases in small\_test\_cases\_input.txt should be 100-150. Big\_test\_cases\_input.txt contains test cases which mainly focus on time complexity of the solution like if expected time complexity is O(N) then these test cases should be tight such that O(N^2) or other solution do not pass. Number of test cases in big\_test\_cases\_input.txt should be 5-7. Here generally we do not use more number of test cases because file size will be increased. When solution is O(N^2) etc then we can have more test cases but when O(N) then we should limit them to 5 to 7. Ignore.txt is for different purpose and you need not to worry. For brute\_force solution’s output I use them. There are other files like xxx\_output.txt. I use them to store optimal\_solution’s output. But they are not imp.

**Test\_cases\_generators**:

This folder contains two files. Small\_test\_cases\_generator contains code to generate small test cases which are stored in small\_test\_cases\_input.txt file. Big\_test\_cases\_generator contains code to generate big test cases which are stored in big\_test\_cases\_input.txt.

**Test\_cases\_parser\_for\_hackerrank:**

As we mentioned earlier, for hackerrank we have to separate our test cases. Test\_cases\_parser\_for\_hackerrank contains code for the same.