

## Kick Start 2020 - Round G

# Analysis: Kick\_Start

The given problem statement can be rephrased as counting the number of substrings (fragments) which begin with string 'KICK' and end with string 'START'. Another thing to note is that a substring starting with 'KICK' can form multiple lucky fragments ending at different 'START'. For example, consider the string 'KICKSTARTSTART'. This string contains two lucky fragments, first being 'KICKSTART' and second being 'KICKSTARTSTART'.

### Test Set 1

For a substring to be considered as a lucky fragment, we can check if it starts with 'KICK' and ends with 'START'. For every substring from  $i$ -th to  $j$ -th index, where  $0 \leq i < j < N$ , we can check if the substring begins with 'KICK' by comparing whether  $i$ -th index is equal to 'K',  $(i+1)$ -th index is equal to 'I', ... ,  $(i+3)$ -th index is equal to 'K'. Similarly we can check if the substring ends with 'START' by comparing whether  $(j-4)$ -th index is equal to 'S',  $(j-3)$ -th index is equal to 'T', ... ,  $j$ -th index is equal to 'T'. Note that we only check the characters if the indices are within the bounds of the substring. Hence, it takes constant number of operations to check whether a substring is lucky fragment or not. In total, we have  $O(N^2)$  substrings to check. Therefore, total time complexity for the approach is  $O(N^2)$ .

Another solution could be to use two loops iterating over the string. Outer loop is used to scan the string for 'KICK'. Whenever 'KICK' is encountered, we use the inner loop to scan the remaining string i.e. substring to the right of 'KICK' to count the occurrences of string 'START'. We repeat this for every 'KICK' encountered using the outer loop and keep summing up the count of occurrences of string 'START' to get the total number of lucky fragments. Rationale is that every 'START' encountered can be paired up with the 'KICK' found using outer loop to create a lucky fragment for Ksenia.

As we iterate over the string using two loops, the overall time complexity for this solution is  $O(N^2)$ .

### Test Set 2

We take a single pass over the string. During this pass, we perform following two checks:

- If we encounter string 'KICK', we increase a counter, let it be  $X$ .
- If we encounter string 'START', we add the number of 'KICK' encountered till now to the final answer, in other words add the current value of  $X$  to the final answer.

Rationale is that every 'START' we encounter can be paired up with any 'KICK' encountered before to create a lucky fragment.

Time complexity is  $O(N)$  for the solution as we take a single pass over the string.