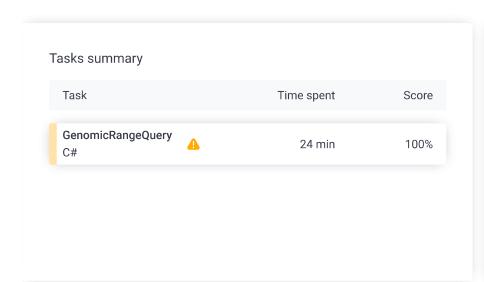
Codility_

CodeCheck Report: trainingM4VEHU-FBE

Test Name:

Summary Timeline Check out Codility training tasks





Tasks Details

1.

GenomicRangeQuery Find the minimal

nucleotide from a range of sequence DNA.

Task Score Performance Correctness 100% 100% 100%

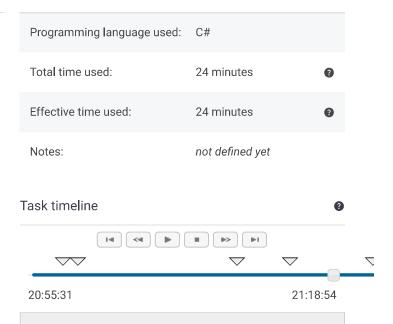
Task description

A DNA sequence can be represented as a string consisting of the letters A, C, G and T, which correspond to the types of successive nucleotides in the sequence. Each nucleotide has an impact factor, which is an integer. Nucleotides of types A, C, G and T have impact factors of 1, 2, 3 and 4, respectively. You are going to answer several queries of the form: What is the minimal impact factor of nucleotides contained in a particular part of the given DNA sequence?

The DNA sequence is given as a non-empty string S = S[0]S[1]...S[N-1] consisting of N characters. There are M queries, which are given in non-empty arrays P and Q, each consisting of M integers. The K-th query $(0 \le K < M)$ requires you to find the minimal impact factor of nucleotides contained in the DNA sequence between positions P[K] and Q[K] (inclusive).

For example, consider string S = CAGCCTA and arrays P, Q such that:

Solution



```
P[0] = 2 Q[0] = 4

P[1] = 5 Q[1] = 5

P[2] = 0 Q[2] = 6
```

The answers to these M = 3 queries are as follows:

- The part of the DNA between positions 2 and 4 contains nucleotides G and C (twice), whose impact factors are 3 and 2 respectively, so the answer is 2.
- The part between positions 5 and 5 contains a single nucleotide T, whose impact factor is 4, so the answer is 4.
- The part between positions 0 and 6 (the whole string) contains all nucleotides, in particular nucleotide A whose impact factor is 1, so the answer is 1.

Write a function:

```
class Solution { public int[] solution(string S,
int[] P, int[] Q); }
```

that, given a non-empty string S consisting of N characters and two non-empty arrays P and Q consisting of M integers, returns an array consisting of M integers specifying the consecutive answers to all queries.

Result array should be returned as an array of integers.

For example, given the string S = CAGCCTA and arrays P, Q such that:

```
P[0] = 2 Q[0] = 4

P[1] = 5 Q[1] = 5

P[2] = 0 Q[2] = 6
```

the function should return the values [2, 4, 1], as explained above.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- M is an integer within the range [1..50,000];
- each element of arrays P and Q is an integer within the range [0..N - 1];
- $P[K] \le Q[K]$, where $0 \le K < M$;
- string S consists only of upper-case English letters A, C, G, T.

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Code: 21:18:54 UTC, cs, final, show code in pop-up score: 100

```
1
     using System;
2
3
      * Lesson 5.3 - Genomic Range Query
4
5
      * Paulo Santos
6
      * 06.Dec.2022
7
      */
8
     class Solution {
         public int[] solution(string S, int[] P, int[]
9
10
         int len = S.Length;
11
         var answers = new int[P.Length];
12
         var occurrences = new int [len, 4];
13
14
15
          ^{st} For each pos, indicate which
16
          \ensuremath{^{*}} nucleotide it is, and perform
17
          * a running sum, after the
18
19
          * second position.
20
          * Eexample. if S = "CAGCCTA"
21
            occurrences will be:
22
23
          * {
24
25
            {0,1,0,0} // C
26
            {1,1,0,0} // A
27
            {1,1,1,0} // G
28
          * {1,2,0,0} // C
          * {1,3,0,0} // C
29
30
            \{1,3,1,1\} // T
31
             \{2,3,1,1\} // A
32
            }
33
          */
34
         for(var i = 0; i < len; i++) {
35
36
           switch (S[i]) {
37
             case 'A': { occurrences[i, 0] = 1; break; }
38
             case 'C': { occurrences[i, 1] = 1; break; }
39
             case 'G': { occurrences[i, 2] = 1; break; }
             default : { occurrences[i, 3] = 1; break; }
40
41
42
           if (i > 0) {
             occurrences[i, 0] += occurrences[i - 1, 0];
43
             occurrences[i, 1] += occurrences[i - 1, 1];
44
45
             occurrences[i, 2] += occurrences[i - 1, 2];
46
             occurrences[i, 3] += occurrences[i - 1, 3];
47
48
         }
49
50
          * Check each slice for min by simple subtracti
51
          */
52
53
         for(var i = 0; i < P.Length; i++) {</pre>
54
           var index1 = P[i];
55
           var index2 = Q[i];
56
57
           for(var j = 0; j < 4; j++) {
58
              * When index1 is not at the beginning
59
60
              * of the string, need to get occurrences
              * from just before beginning of the slice
61
              * to see if that nucleotide occurred
62
63
              * within the slice.
64
              ^{*} Example: is slice is (2, 4), need to che
65
66
              * for occurrences of A, C, G, T from index
67
68
             int lic = 0;
69
             if (index1 - 1 >= 0) {
70
               lic = occurrences[index1 - 1, j];
```

```
71
              }
72
              if(occurrences[index2, j] - lic > 0) {
73
74
                 \overset{'}{st} nucleotide value is 1 more than
75
                 * loop value (A=1, C=2, G=3, T=4)
76
77
                answers[i] = j + 1;
78
79
80
                 st no need to keep checking since
81
                 ^{st} always checking from smallest
82
83
                 * impact factor as soon as find occurren
84
                 * that has to be minimum, cause
                 \ ^*\ {\hbox{subsequent nucleotides have}}
85
                 * larger impact factor
86
                 */
87
88
                break;
89
90
91
          }
92
          return answers;
93
94
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(N + M)

expand all Example tests		
example example test		✓ OK
expand all Correctness tests		
extreme_sir single characte	3	✓ OK
extreme_do double charact		✓ OK
simple simple tests		✓ OK
small_lengt small length si	•	✓ OK
small_random	om string, length = ~300	√ OK
expand all	Performance t	ests
	same_letters gggggg??gggggg	√ OK
► large_rando		√ OK
extreme_lar all max ranges	_	✓ OK