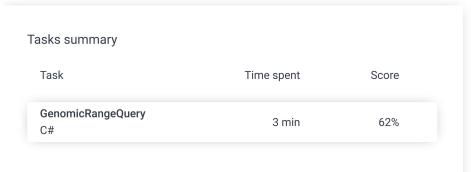
Codility_

Candidate Report: trainingWK98YQ-GSX

Check out Codility training tasks

Test Name:

Feedback Summary Timeline





Tasks Details

1. GenomicRangeQuery Task Score Correctness Performance Find the minimal nucleotide from a range of 62% 100% 0% sequence DNA.

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 nonempty arrays P and Q, each consisting of M integers. The K-th query (0 ≤ 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:

P[0] = 2Q[0] = 4P[1] = 5Q[1] = 5P[2] = 0Q[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

Solution

Programming language used: Total time used: 3 minutes Effective time used: 3 minutes Notes: not defined yet Task timeline ∇ 16:25:53 16:27:59 Code: 16:27:59 UTC, cs, final, show code in pop-up score: 62

- 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, 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 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
using System;
     using System.Collections.Generic;
     // you can also use other imports, for example:
3
     // using System.Collections.Generic;
 4
     // you can write to stdout for debugging purposes, e.g.
 6
     // Console.WriteLine("this is a debug message");
8
9
     class Solution {
10
              public int[] solution(String S, int[] P, int[] Q)
11
12
13
                  Dictionary<char, int> sequenceDictionary;
14
                  sequenceDictionary = new Dictionary<char, int>(
15
                  {
                      {'A',1},{'C',2},{'G',3},{'T',4}
16
17
                  };
18
                  int M = P.Length;
                  int[] retVal = new int[M];
19
20
21
                  for (int i = 0; i < M; i++)
22
23
                      int min = int.MaxValue;
24
                      for (int j=P[i];j<=Q[i];j++)</pre>
25
                      { int val = sequenceDictionary[S[j]];
26
                          if (val < min)</pre>
27
                              min = val:
28
                      retVal[i] = min;
29
30
                  }
31
                  return retVal;
32
             }
33
34
     }
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis 👩

Detected time complexity: O(N * M)

expar	nd all	Example tests	
•	example example test	✓	OK
expand all		Correctness tests	
•	extreme_sinlge single character string	√	OK
•	extreme_double double character string	√	OK
•	simple simple tests	√	OK
•	small_length_string	,	OK
•	small_random small random string, len	•	OK
expar	nd all	Performance tests	
•	almost_all_same_le	•	TIMEOUT ERROR Killed. Hard limit reached:

		6.000 sec.
•	large_random large random string, length	X TIMEOUT ERROR Killed. Hard limit reached: 6.000 sec.
•	extreme_large all max ranges	X TIMEOUT ERROR Killed. Hard limit reached:
		6.000 sec.

The PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.