

Candidate Report: training34BJQQ-FZD

Check out Codility training tasks

Test Name:

Summary

Timeline

Tasks summary

Task	Time spent	Score
GenomicRangeQuery Java 8	1 min	62%

Total score

62%

Tasks Details

Medium	1. <a href="#">GenomicRangeQuery</a>	Task Score	Correctness	Performance
	Find the minimal nucleotide from a range of sequence DNA.	62%	100%	0%

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  $S = S[0]S[1] \dots S[N-1]$  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 ( $0 \leq K < M$ ) 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  $S = \text{CAGCCTA}$  and arrays  $P, Q$  such that:

P[0] = 2      Q[0] = 4



P[1] = 5      Q[1] = 5

P[2] = 0      Q[2] = 6

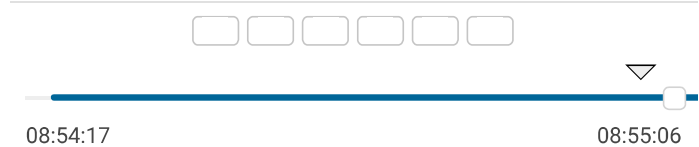
The answers to these  $M = 3$  queries are as follows:

- 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

Solution

Programming language used:	Java 8	
Total time used:	1 minutes	
Effective time used:	1 minutes	
Notes:	not defined yet	

Task timeline



Code: 08:55:06 UTC, java, final, score: 62

[show code in pop-up](#)

```
1 // you can also use imports, for example:
2 // import java.util.*;
3
4 // you can write to stdout for debugging purposes, e.g.
5 // System.out.println("this is a debug message");
6 import java.util.List;
7 import java.util.ArrayList;
8 class Solution {
```

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];
- $P[K] \leq Q[K]$ , where  $0 \leq K < M$ ;
- 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.

```
9      public int[] solution(String S, int[] P, int[] Q) {
10          // write your code in Java SE 8
11          int[] result = new int[P.length];
12
13          if (S != null) {
14              if (!S.isEmpty()) {
15                  List<Integer> nucleotides
16                      for (Character c : S.toCharArray()) {
17                          switch (c) {
18                              case 'A':
19                                  nucleotide
20                                  break;
21                              case 'C':
22                                  nucleotide
23                                  break;
24                              case 'G':
25                                  nucleotide
26                                  break;
27                              case 'T':
28                                  nucleotide
29                                  break;
30                              default:
31                                  System.err
32                                  }
33                          }
34                      for(int i=0;i<P.length;i++)
35                          if(P[i]==Q[i])
36                              result[i]
37                          else
38                              result[i]
39                      }
40                  }
41              }
42          }
43
44          return result;
45
46      }
47  }
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis ?

Detected time complexity: **O(N \* M)**

Example tests	
▶ example	✓ OK
example test	
Correctness tests	
▶ extreme_single	✓ OK
single character string	
▶ extreme_double	✓ OK
double character string	
▶ simple	✓ OK
simple tests	
▶ small_length_string	✓ OK
small length simple string	
▶ small_random	✓ OK
small random string, length = ~300	
Performance tests	
▼ almost_all_same_letters	✗ TIMEOUT ERROR
GGGGGG..??..GGGGGG..??..GGGGGG	
Killed. Hard limit reached:	
8.000 sec.	

1.	8.000 s	TIMEOUT ERROR, Killed. Hard limit reached: 8.000 sec.
2.	0.248 s	OK
<hr/>		
<div>▼ large_random</div> <div>large random string, length</div> <div>8.000 sec.</div> <div>TIMEOUT ERROR</div> <div>Killed. Hard limit reached: 8.000 sec.</div>		
<hr/>		
1.	8.000 s	TIMEOUT ERROR, Killed. Hard limit reached: 8.000 sec.
<hr/>		
<div>▼ extreme_large</div> <div>all max ranges</div> <div>9.000 sec.</div> <div>TIMEOUT ERROR</div> <div>Killed. Hard limit reached: 9.000 sec.</div>		
<hr/>		
1.	9.000 s	TIMEOUT ERROR, Killed. Hard limit reached: 9.000 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.