

AVAILABLE
LESSONS:*Lesson 1*
Iterations*Lesson 2*
Arrays*Lesson 3*
Time Complexity*Lesson 4*
Counting
Elements*Lesson 5*
Prefix Sums*Lesson 6*
Sorting*Lesson 7*
Stacks and
Queues*Lesson 8*
Leader*Lesson 9*
Maximum slice
problem*Lesson 10*
Prime and
composite
numbers*Lesson 11*

RESPECTABLE

GenomicRangeQuery

START

Find the minimal nucleotide from a range of sequence DNA.

Programming language: C++ ▼

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

Sieve of
Eratosthenes

Lesson 12
Euclidean
algorithm

Lesson 13
Fibonacci
numbers

Lesson 14
Binary search
algorithm

Lesson 15
Caterpillar
method

Lesson 16
Greedy
algorithms

Lesson 17
Dynamic
programming

Lesson 90
Tasks from
Indeed Prime
2016 challenge

Lesson 99
Future training

nucleotide A whose impact factor is 1, so the answer is 1.

Write a function:

```
vector<int> solution(string &S, vector<int> &P,
vector<int> &Q);
```

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

The sequence should be returned as:

- a Results structure (in C), or
- a vector of integers (in C++), or
- a Results record (in Pascal), or
- an array of integers (in any other programming language).

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.

Assume that:

- *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.

Complexity:

- expected worst-case time complexity is $O(N+M)$;
- expected worst-case space complexity is $O(N)$, beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

For programmers

[Lessons](#) [Challenges](#)
[Terms](#) [FAQ](#)

For companies

[Tour](#) [About us](#)
[Pricing](#) [Jobs](#)
[Blog](#) [Terms](#)
[Privacy](#) [Cookies](#)
[API](#)

Sign up for our newsletter:

Information about upcoming challenges, solutions and lessons directly in your inbox.

[Sign up](#)

Social:

[f](#) [t](#) [in](#)

Contact us:

For customer support queries:

UK +44 (0) 208 970 78 68

US 1-415-466-8085
support@codility.com

For sales queries:

UK +44 (0) 208 970 78 67

US 1-415-466-8085
sales@codility.com

© 2009–2016 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID GB981191408. Registered office: 107 Cheapside, London EC2V 6DN