agile content

1) AdjacentMaxDistance

Integer V lies strictly between integers U and W if $U < V < W$ or if $U > V > W$.

A non-empty zero-indexed array A consisting of N integers is given. A pair of indices (P, Q), where $0 \leq P < Q < N$, is said to have adjacent values if no value in the array lies strictly between values A[P] and A[Q].

For example, in array A such that:

```
A[0] = 0
A[1] = 3
A[2] = 3
A[3] = 7
A[4] = 5
A[5] = 3
A[6] = 11
A[7] = 1
```

the following pairs of indices have adjacent values:

```
(0, 7), (1, 2), (1, 4),
(1, 5), (1, 7), (2, 4),
(2, 5), (2, 7), (3, 4),
(3, 6), (4, 5), (5, 7).
```

For example, indices 4 and 5 have adjacent values because there is no value in array. A that lies strictly between A[4] = 5 and A[5] = 3; the only such value could be the number 4, and it is not present in the array.

Given two indices P and Q, their distance is defined as abs(A[P] − A[Q]), where abs(X) = X for $X \geq 0$, and abs(X) = −X for $X < 0$. For example, the distance between indices 4 and 5 is 2 because (A[4] − A[5]) = (5 − 3) = 2.

Write a function:

class Solution { public int solution(int[] A); }

that, given a non-empty zero-indexed array A consisting of N integers, returns the maximum distance between indices of this array that have adjacent values. The function should return −2 if no adjacent indices exist.

Assume that:

- N is an integer within the range [1..40,000];
- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

For example, given array A such that:

```
A[0] = 0
A[1] = 3
A[2] = 3
A[3] = 12
A[4] = 5
A[5] = 3
A[6] = 7
A[7] = 1
```

the function should return 5 because:

- indices 3 and 6 are adjacent, because the array does not contain any value that lies strictly between A[3] = 12 and A[6] = 7;
- the distance between these indices is (A[3] − A[6]) = (12 - 7) = 5;
- no other pair of adjacent indices that has greater distance exists.

Performance:

- The best possible performance is expected.

2 - New CDN iTaas

Log files can tell much about a system’s behavior in a production environment. Extracting data from these files helps the decision-making process for both business and development roadmap.

iTaaS Solution is a company focused on content delivery, and one of its largest business challenges was CDN (Content Delivery Network) costs. Larger CDN costs increase the final pricing for its customers, reduce profits, and make it difficult to enter smaller markets.

After extensive research from their software engineers and financial team, they obtained an excellent deal from the company “MINHA CDN”, and signed a one-year contract with them.

iTaaS Solution already has a system capable of generating billing reports from logs, called “Agora”. However, it uses a specific log format, different from the format used by ”MINHA CDN”.

You have been hired by iTaaS Solution to develop a system that can convert log files to the desired format, which means that at this moment they need to convert them from the “MINHA CDN” format to the “Agora” format.

This is a sample log file in the “MINHA CDN” format:

```
312|200|HIT|"GET /robots.txt HTTP/1.1"|100.2
101|200|MISS|"POST /myImages HTTP/1.1"|319.4
199|404|MISS|"GET /not-found HTTP/1.1"|142.9
312|200|INVALIDATE|"GET /robots.txt HTTP/1.1"|245.1
```

The sample above should generate the following log in the “Agora” format:

```
#Version: 1.0
#Date: 15/12/2017 23:01:06
#Fields: provider http-method status-code uri-path time-taken
response-size cache-status
"MINHA CDN" GET 200 /robots.txt 100 312 HIT
"MINHA CDN" POST     200 /myImages 319 101 MISS
"MINHA CDN" GET 404 /not-found 143 199 MISS
"MINHA CDN" GET 200 /robots.txt 245 312 REFRESH_HIT
```

“MINHA CDN” will make log files through specific URLs.

The specification requires you to implement a Console Application that receives as input an URL (sourceUrl) and a destination file (targetPath). A sample call would be:

```
convert http://logstorage.com/minhaCdn1.txt ./output/minhaCdn1.txt
```

The sourceUrl parameter includes a file in the “MINHA CDN” format, and the output specified in the targetPath parameter corresponds to the file that needs to be created in the “Agora” format.

All namespaces should follow the pattern:
`CandidateTesting.YourFullName.YourNamespace`

Put all your solution in a single project. Unit Tests can be in a different project.

A sample log file that can be used for testing is available here:

https://s3.amazonaws.com/uux-itaas-static/minha-cdn-logs/input-01.txt

Beware that what will be analyzed on this exercise is the not only the correction of the code but also code best practices, like OOP, SOLID, unit tests and mocks.