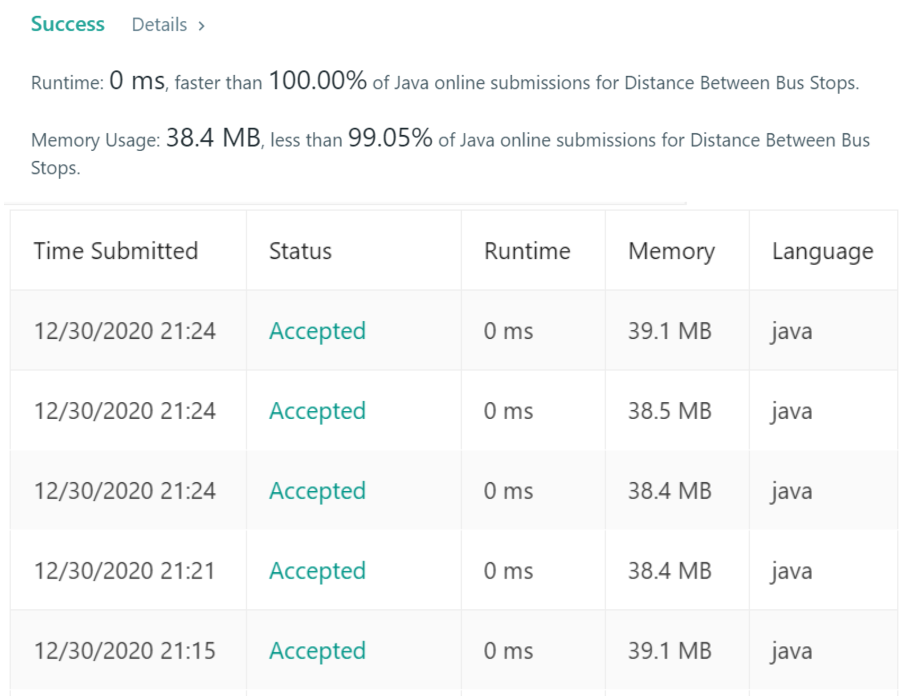
Maisha Aniqa
Student ID: 500961140

<u>Distance Between Bus Stops</u>

Screenshot of accepted solution:



Success Details ›

Runtime: 0 ms, faster than 100.00% of Java online submissions for Distance Between Bus Stops.

Memory Usage: 38.4 MB, less than 99.05% of Java online submissions for Distance Between Bus Stops.

Time Submitted	Status	Runtime	Memory	Language
12/30/2020 21:24	Accepted	0 ms	39.1 MB	java
12/30/2020 21:24	Accepted	0 ms	38.5 MB	java
12/30/2020 21:24	Accepted	0 ms	38.4 MB	java
12/30/2020 21:21	Accepted	0 ms	38.4 MB	java
12/30/2020 21:15	Accepted	0 ms	39.1 MB	java

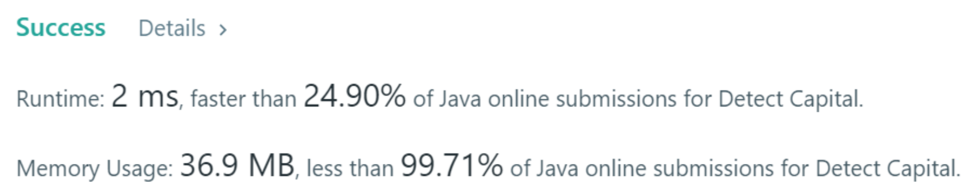
Explanation:

The question asks to determine the shortest distance between the start and destination stops. Clockwise direction is when the bus goes from start to the destination and counter-clockwise direction is when the bus goes from destination to start. In order to accomplish this, an index was assigned to start and destination, where the start index was greater than the destination index. In other words, the destination stop came after the start stop. Then, the distance in clockwise and counter-clockwise direction was found using for-loops. Finally, the minimum of the two distances was found using if-else statements. The code can be made more efficient by finding the total distance first including all the stops. Then, finding the clockwise distance and then subtracting the clockwise distance from the total distance to find the counter clockwise distance and comparing the two values using min() method to get the shortest distance.

Maisha Aniqa
Student ID: 500961140

Detect Capital

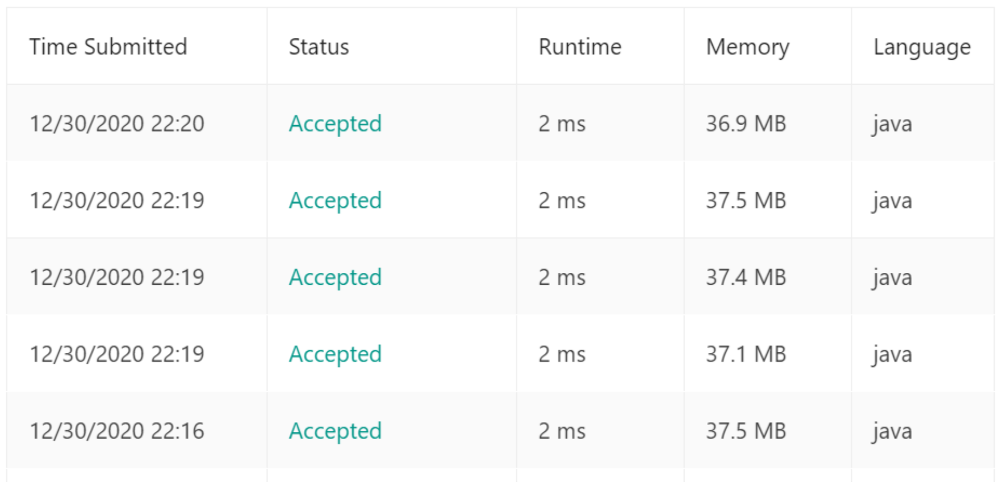
Screenshot of accepted solution:



Success Details ›

Runtime: 2 ms, faster than 24.90% of Java online submissions for Detect Capital.

Memory Usage: 36.9 MB, less than 99.71% of Java online submissions for Detect Capital.



Time Submitted	Status	Runtime	Memory	Language
12/30/2020 22:20	Accepted	2 ms	36.9 MB	java
12/30/2020 22:19	Accepted	2 ms	37.5 MB	java
12/30/2020 22:19	Accepted	2 ms	37.4 MB	java
12/30/2020 22:19	Accepted	2 ms	37.1 MB	java
12/30/2020 22:16	Accepted	2 ms	37.5 MB	java

Explanation:

If the usage of capitals in a given word is correct, the program returns true, otherwise, it returns false. Based on case 1 and case 2, when all the letters in the inputted word is the same case (lower-case or upper-case), the program returns true. This was implemented in the code using toLowerCase() and toUpperCase() functions. Based on case 3, when only the first letter in the word is upper-case, the program returns true. This was implemented in the code using a for-loop that goes over each letter and if the first letter (at index 0) is upper-case, it returns true. An else-statement was also used inside the for-loop that checks to see if any of remaining letters is in upper-case, in which case, it returns false. The charAt() function used in the else-if statements returns the character at an index specified by the loop. For case 1, it's important to check if the words have all upper-case letters **after** the for-loop, since the for-loop returns false if any of the letter (other than the first letter) is in upper-case. Thus, the algorithm needs to implement case 1 after case 3, otherwise the program will return false for any word with all upper-case letters. One way the program can be made more efficient is by writing an algorithm that returns true or false, regardless of the order each case is implemented in.