P2
300202727
Omar Khattab

P2 Report

Part 1: Document Analyser – 11860

I was trying to approach a solution for this problem where I would get the range, p and q, at the time of building my HashMap<String, ArrayList<Integer>, where the String was the words, and the ArrayList was an integer list of indices where the words occurred in the document. My solution worked for the first test file, testDocAnalyser1.txt, but for the second test file, testDocAnalyserBig.txt, I realized that in the range of p and q there could be duplications of some of the words in document number 4, and then I realized that I would need to change my approach as I was not counting for the duplicates in my solution. I tried running the solution I had on the online judge website, and I was getting a runtime of 3.900 which made me think that even if I could fix my code, it would most probably exceed time limit.

My Submissions

#	Problem	Verdict	Language	Run Time	Submission Date
27030351	11860 Document Analyzer	Accepted	JAVA	1.430	2021-12-06 02:02:15
27030237	11860 Document Analyzer	Wrong answer	JAVA	1.690	2021-12-06 01:31:28
27030235	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-06 01:31:00
27024561	11860 Document Analyzer	Accepted	JAVA	1.600	2021-12-04 08:35:11
27024545	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-04 08:30:12
27024542	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-04 08:29:11
27021750	11094 Continents	Accepted	JAVA	0.050	2021-12-03 11:50:12
27016404	11094 Continents	Accepted	JAVA	0.050	2021-12-01 22:07:55
27014569	11860 Document Analyzer	Wrong answer	JAVA	3.900	2021-12-01 09:46:23
27014429	11860 Document Analyzer	Wrong answer	JAVA	0.070	2021-12-01 08:59:42
27014424	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-01 08:58:45
27014322	11860 Document Analyzer	Runtime error	JAVA	0.000	2021-12-01 08:20:25

So I decided to change my approach and I realized that would need to number all words first before even looking for the range, I did that by switching the keys and

values of my HashMap words so that the keys would be the numbering of the words, and the values would be the word associated to that index, then I used an algorithm where I would check if my range starting from index 1 in my HashMap words and looping over the size of it, checking if each word is already in the range or not and by using another HashMap<String, Integer> I was storing the words in the range as the keys and their values were the number of occurrences in the range, once my range have all the different words in it I try to shrink the range such that if I the first word in my range occurred more than once I would increment the begging of the range and decrement the number of occurrences of the word of the begging of the range, then by comparing the length of the current range with length of the my previous range I was able to determine which range was the smallest and was including all words ignoring the duplicate case that I did not account for in my first solution. The new solution runtime was more than two times faster than the first one, 1.430, and it passed the verdict testing of the online judge.

My Submissions

#	Problem	Verdict	Language	Run Time	Submission Date
27030351	11860 Document Analyzer	Accepted	JAVA	1.430	2021-12-06 02:02:15
27030237	11860 Document Analyzer	Wrong answer	JAVA	1.690	2021-12-06 01:31:28
27030235	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-06 01:31:00
27024561	11860 Document Analyzer	Accepted	JAVA	1.600	2021-12-04 08:35:11
27024545	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-04 08:30:12
27024542	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-04 08:29:11
27021750	11094 Continents	Accepted	JAVA	0.050	2021-12-03 11:50:12
27016404	11094 Continents	Accepted	JAVA	0.050	2021-12-01 22:07:55
27014569	11860 Document Analyzer	Wrong answer	JAVA	3.900	2021-12-01 09:46:23
27014429	11860 Document Analyzer	Wrong answer	JAVA	0.070	2021-12-01 08:59:42
27014424	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-01 08:58:45
27014322	11860 Document Analyzer	Runtime error	JAVA	0.000	2021-12-01 08:20:25

Part 2: Continents – 11094

I found part 2 easier than the first part as it was pretty straight forward, also because I solved a similar question before (Mazes), by using recursions and by knowing in which direction Mijid can move I was able to build the recursive method getContinentSize and get the size of different continents not conquered by Mijid and returning the biggest one of them. Mijid could only move in four directions; North, east, south and west, which was why I had two look in 4 different directions in my recursive method, after debugging I was able to identify the edge cases and my run time on the online judge was actually 0.05!

My Submissions

#	Problem	Verdict	Language	Run Time	Submission Date
27030351	11860 Document Analyzer	Accepted	JAVA	1.430	2021-12-06 02:02:15
27030237	11860 Document Analyzer	Wrong answer	JAVA	1.690	2021-12-06 01:31:28
27030235	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-06 01:31:00
27024561	11860 Document Analyzer	Accepted	JAVA	1.600	2021-12-04 08:35:11
27024545	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-04 08:30:12
27024542	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-04 08:29:11
27021750	11094 Continents	Accepted	JAVA	0.050	2021-12-03 11:50:12
27016404	11094 Continents	Accepted	JAVA	0.050	2021-12-01 22:07:55
27014569	11860 Document Analyzer	Wrong answer	JAVA	3.900	2021-12-01 09:46:23
27014429	11860 Document Analyzer	Wrong answer	JAVA	0.070	2021-12-01 08:59:42
27014424	11860 Document Analyzer	Compilation error	JAVA	0.000	2021-12-01 08:58:45
27014322	11860 Document Analyzer	Runtime error	JAVA	0.000	2021-12-01 08:20:25

Program Development

In believe this assignment was the one I learned stuff from the most as it taught me how to account for memory and use it wisely, also the way of solving difficult problems, as we would first go for the silly solution, then improve, then when the solution is correct, we would try to optimize it to our best. Overall great assignment for learning the differences of different time complexity and debugging.

_____(This concludes P2 report_____