Solution 17-11-23

30. Substring with Concatenation of All Words

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;


public class SubstringConcatenation {

    public static List<Integer> findSubstring(String s, String[] words) {

        List<Integer> result = new ArrayList<>();

        if (s == null || s.length() == 0 || words == null || words.length == 0) {

            return result;

        }


        int wordLength = words[0].length();

        int wordCount = words.length;

        int totalLength = wordLength * wordCount;


        Map<String, Integer> wordMap = new HashMap<>();

        for (String word : words) {

            wordMap.put(word, wordMap.getOrDefault(word, 0) + 1);

        }


        for (int i = 0; i <= s.length() - totalLength; i++) {

            Map<String, Integer> seen = new HashMap<>();

            int j = 0;

            while (j < wordCount) {
                int startIndex = i + j * wordLength;
                int endIndex = startIndex + wordLength;
                String currentWord = s.substring(startIndex, endIndex);
                if (wordMap.containsKey(currentWord)) {
                    seen.put(currentWord, seen.getOrDefault(currentWord, 0) + 1);
                    if (seen.get(currentWord) > wordMap.get(currentWord)) {
                        break;
                    }
                } else {
                    break;
                }
                j++;
            }
            if (j == wordCount) {
                result.add(i);
            }
        }

        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the input string:");
        String inputString = scanner.nextLine();

        System.out.println("Enter the words (separated by space):");
        String[] words = scanner.nextLine().split(" ");
        List<Integer> result = findSubstring(inputString, words);


        System.out.println("Indices of substring with concatenation of all words: " + result);
    }
}
```

Output:-

```
java -cp /tmp/eGGukgloUC SubstringConcatenation
Enter the input string:
foodtakenbycafe
Enter the words (separated by space):
food cafe
Indices of substring with concatenation of all words: []
1,2
dash: 2: 1,2: not found

java -cp /tmp/eGGukgloUC SubstringConcatenation
Enter the input string:
barfoothefoobarman
Enter the words (separated by space):foo bar
Indices of substring with concatenation of all words: [0, 9]
```

35. Search Insert Position

```
import java.util.Scanner;

public class SearchInsertPosition {

    public static int searchInsert(int[] nums, int target) {
        if (nums == null || nums.length == 0) {
```

```
        return 0;
    }

    int low = 0;
    int high = nums.length - 1;

    while (low <= high) {
        int mid = low + (high - low) / 2;

        if (nums[mid] == target) {
            return mid;
        } else if (nums[mid] < target) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }

    return low;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.println("Enter the size of the array:");
    int size = scanner.nextInt();

    int[] nums = new int[size];

    System.out.println("Enter the sorted array elements:");
    for (int i = 0; i < size; i++) {
            nums[i] = scanner.nextInt();
        }

        System.out.println("Enter the target element:");
        int target = scanner.nextInt();

        int result = searchInsert(nums, target);

        System.out.println("Index where the target would be inserted: " + result);
    }
}
```

Output:-

```
java -cp /tmp/VILs5TXjVN SearchInsertPosition
Enter the size of the array:
5
Enter the sorted array elements:
1
2
3
5
6
Enter the target element:
3
Index where the target would be inserted: 2
```