

# CSC 382 Analysis of Algorithms

## Exam I

Name: \_\_\_\_\_

### 1. (5 points) **Palindrome**

Given an Array `arr` of letters, return `true` if `arr` is a **palindrome**.

An array is a **palindrome** when it reads the same backward as forward.

For example, `abcba` is a palindrome while `abcda` is not.

-- Example 1 --

Input: `arr = ['a', 'b', 'c', 'b', 'a']`

Output: `true`

-- Example 2 --

Input: `arr = ['2', '0', '2', '2']`

Output: `false`

2. (5 points) **Count Negative Numbers in a Sorted Array**

Given a sorted array `nums` (ascending order), find the number of negative numbers in it.

-- Example --

Input: `nums = [-101, -99, -54, -21, -3, 6, 12, 19, 20, 74, 92]`

Output: 5

3. (5 points) **Prepend Function of a Vector**

**Vectors** (or **Dynamic Arrays**) are sequence containers representing arrays that can change in size.

Write the `prepend` (add a new element to the beginning) function of a vector.

-- Example --

```
cout << vec << endl;    // output: 3, 1, 4, 2, 8
vec.prepend(5);
cout << vec << endl;    // output: 5, 3, 1, 4, 2, 8
```

4. (5 points) **3/4 Place of a Linked List**

Find the node at 3/4 place of a linked list.

-- Example --

Input: head->(3)->(1)->(4)->(2)->(8)->(9)->(7)->(6)->null

Output: (9) ^

Explanation: There are 8 nodes in all, and  $8 * (3/4) = 6$ .

So, the 3/4 place node is the 6th node which is 9.

5. (5 points) **Group Anagrams**

Given an array of strings `strs` (consist of lowercase English letters), group the **anagrams** together. You can return the answer in **any order**.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

-- Example --

Input: `strs = ["eat","tea","tan","ate","nat","bat"]`

Output: `[["bat"],["nat","tan"],["ate","eat","tea"]]`

6. (5 points) **Range Sum of BST**

Given a binary search tree `bst` and two integers `low` and `high`, return the sum of values of all nodes with a value in the inclusive range `[low, high]`. (All node values are unique.)

-- Example 1 --

Input: `bst = [10,5,15,3,7,null,18]`, `low = 7`, `high = 15`

(see Figure 1)

Output: 32

Explanation: Nodes 7, 10, and 15 are in the range `[7, 15]`.  $7 + 10 + 15 = 32$ .

-- Example 2 --

Input: `bst = [10,5,15,3,7,13,18,1,null,6]`, `low = 6`, `high = 10`

(see Figure 2)

Output: 23

Explanation: Nodes 6, 7, and 10 are in the range `[6, 10]`.  $6 + 7 + 10 = 23$ .

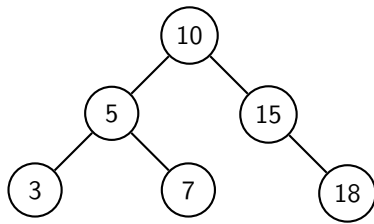


Figure 1

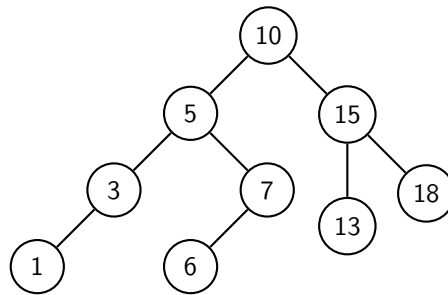


Figure 2