Online Interview Questions - Problems Wise

deepwaterooo

February 5, 2015

Contents

1 4 Sum, wrong answers 5

2 2 Sum: O(1) Space, O(nlogn) Time undone 7
2.1 O(1) Space, O(nlogn) Time 7
2.2 O(n) Space, O(n) Time 7

3 3 Sum 9

4 3 Sum Closest 11

5 A + B Problem, no idea;;; 13

6 Anagrams My Submissions 15

7 Backpack: still feeling difficult for me now··· 17

8 Balanced Binary Tree 19

Chapter 1

4 Sum, wrong answers

Given an array S of n integers, are there elements a, b, c, and d in S such that a + b + c + d = target? Find all unique quadruplets in the array which gives the sum of target.

Note

Elements in a quadruplet (a,b,c,d) must be in non-descending order. (ie, a ⩽ b ⩽ c ⩽ d)

The solution set must not contain duplicate quadruplets.

Example

For example, given array S = {1 0 -1 0 -2 2}, and target = 0. A solution set is:

```
(-1, 0, 0, 1)
(-2, -1, 1, 2)
(-2, 0, 0, 2)
```

```java
/**
 * @param numbers : Give an array numbersbers of n integer
 * @param target : you need to find four elements that's sum of target
 * @return : Find all unique quadruplets in the array which gives the sum of
 *           zero.
 */
public ArrayList<ArrayList<Integer>> fourSum(int[] numbers, int target) {
    int n = numbers.length;
    HashSet<ArrayList<Integer>> set = new HashSet<ArrayList<Integer>>();
    ArrayList<ArrayList<Integer>> res = new ArrayList<ArrayList<Integer>>();
    if (n < 4) return res;
    Arrays.sort(numbers);
    int x = 0, y = 0, tmp = 0;
    for (int i = 0; i < n - 3; i++) {
        if (i > 0 && numbers[i] == numbers[i - 1]) continue;
        for (int j = i + 1; j < n - 2; j++) {
            x = j + 1;
            y = n - 1;
            while (x < y) {
                tmp = target - numbers[i] - numbers[j];
                if (numbers[x] + numbers[y] == tmp) {
                    set.add(new ArrayList<Integer>(Arrays.asList(numbers[i], numbers[j]
                    x++;
                    y--;
                }
                else if (numbers[x] + numbers[y] < tmp) x++;
                else y--;
            }
        }
    }
    res.addAll(set);
    return res;
}
```

Chapter 2

2 Sum: O(1) Space, O(nlogn) Time undone

25% Accepted

Given an array of integers, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

Note

You may assume that each input would have exactly one solution

Example

numbers=[2, 7, 11, 15], target=9

return [1, 2]

Challenge

2.1 O(1) Space, O(nlogn) Time

2.2 O(n) Space, O(n) Time

```java
public int[] twoSum(int[] numbers, int target) { // O(nlogn), O(1)
    Arrays.sort(numbers);
    int [] res = new int[2];
    int i = 0, j = numbers.length - 1;
    while (i < j) {
        if (numbers[i] + numbers[j] == target) {
            res[0] = i + 1;
            res[1] = j + 1;
            return res;
        }
        if (numbers[i] + numbers[j] < target) i++;
        else j--;
    }
    return res;
}
```

Chapter 3

3 Sum

19% Accepted

Given an array S of n integers, are there elements a, b, c in S such that a + b + c = 0? Find all unique triplets in the array which gives the sum of zero.

Note

Elements in a triplet (a,b,c) must be in non-descending order. (ie, $a \leqslant b \leqslant c$)

The solution set must not contain duplicate triplets.

Example

For example, given array S = {-1 0 1 2 -1 -4}, A solution set is:

```
(-1, 0, 1)
(-1, -1, 2)
```

```
public ArrayList<ArrayList<Integer>> threeSum(int[] numbers) {
    int n = numbers.length;
    ArrayList<ArrayList<Integer>> res = new ArrayList<ArrayList<Integer>>();
    HashSet<List<Integer>> set = new HashSet<List<Integer>>();
    Integer [] one = new Integer[3];
    int k = 0;
    for (int i = 0; i < n - 2; i++) {
        for (int j = i + 1; j < n - 1; j++) {
            k = j + 1;
            while (k < n - 1 && numbers[i] + numbers[j] + numbers[k] != 0) k++;
            if (k <= n - 1 && numbers[i] + numbers[j] + numbers[k] == 0) {
                one[0] = numbers[i];
                one[1] = numbers[j];
                one[2] = numbers[k];
                Arrays.sort(one);
                set.add(new ArrayList<Integer>(Arrays.asList(one)));
                k++;
            }
        }
    }
    for(List<Integer> i : set)
        res.add(new ArrayList(i));
    return res;
}
```

Chapter 4

3 Sum Closest

30% Accepted

Given an array S of n integers, find three integers in S such that the sum is closest to a given number, target. Return the sum of the three integers.

Note

You may assume that each input would have exactly one solution.

Example

For example, given array S = {-1 2 1 -4}, and target = 1. The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

```java
public int threeSumClosest(int[] numbers ,int target) {
    int n = numbers.length;
    int res = Integer.MAX_VALUE;
    int k = 0;
    for (int i = 0; i < n - 2; i++) {
        for (int j = i + 1; j < n - 1; j++) {
            k = j + 1;
            while (k <= n - 1) {
                if (Math.abs(numbers[i] + numbers[j] + numbers[k] - target) < Math.abs(
                    res = numbers[i] + numbers[j] + numbers[k];
                k++;
            }
        }
    }
    return res;
}
```

Chapter 5

A + B Problem, no idea;;;

60% Accepted
For given numbers a and b in function aplusb, return the sum of them.
Note
You don't need to parse the input and output. Just calculate and return.
Example
If a = 1 and b = 2 return 3
Challenge
Can you do it without + operation?
Clarification
Are a and b both 32-bit integers?

- Yes.

Chapter 6

Anagrams My Submissions

28% Accepted

Given an array of strings, return all groups of strings that are anagrams.

Note

All inputs will be in lower-case

Example

```
Given a string list: ["lint","intl","inlt","code"]
return ["lint","inlt","intl"]

public String mySort(String s) {
    char [] tmp = s.toCharArray();
    Arrays.sort(tmp);
    return new String(tmp);   //tmp.toString(); doesn't work
}

public List<String> anagrams(String[] strs) {
    Map<String, List<Integer>> m = new HashMap<String, List<Integer>>();
    ArrayList<String> res = new ArrayList<String>();
    for (int i = 0; i < strs.length; i++) {
        String tmp = mySort(strs[i]);
        if (!m.containsKey(tmp))
            m.put(tmp, new ArrayList<Integer>(Arrays.asList(i)));
        else m.get(tmp).add(i);
    }
    for (String key : m.keySet())
        if (m.get(key).size() > 1)
            for (int i = 0; i < m.get(key).size(); i++)
                res.add(strs[m.get(key).get(i)]);
    return res;
}
```

Chapter 7

Backpack: still feeling difficult for me now ...

17% Accepted

Given n items with size A[i], an integer m denotes the size of a backpack. How full you can fill this backpack?

Note

You can not divide any item into small pieces.

Example

If we have 4 items with size [2, 3, 5, 7], the backpack size is 11, we can select 2, 3 and 5, so that the max size we can fill this backpack is 10. If the backpack size is 12. we can select [2, 3, 7] so that we can fulfill the backpack.

You function should return the max size we can fill in the given backpack.

Chapter 8

Balanced Binary Tree

46% Accepted

Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of every node never differ by more than 1.

Example

Given binary tree A={3,9,20,#,#,15,7}, B={3,#,20,15,7}

```
A)  3            B)    3
   / \                  \
  9  20                 20
    /  \                / \
   15   7              15  7
```

The binary tree A is a height-balanced binary tree, but B is not.