Experiment-2.2

Student Name: Nabha Varshney **UID: 20BCS4995**

Branch: CSE **Section/Group: 20BCS-DM-704 (A)**

Semester: 6th **Date of Performance:12th Apr 2023**

Subject Name: Competitive Coding II **Subject Code: 20CSP- 351**

Aim – To demonstrate the concept of Graphs

Objective-

- The objective is to build problem solving capability and to learn the basic concepts of data structures.
- The implementation of find the difference which shows and brushes up the concept of Graphs and can be solved through various approaches.
- The implementation of predict the winner problem in C++.

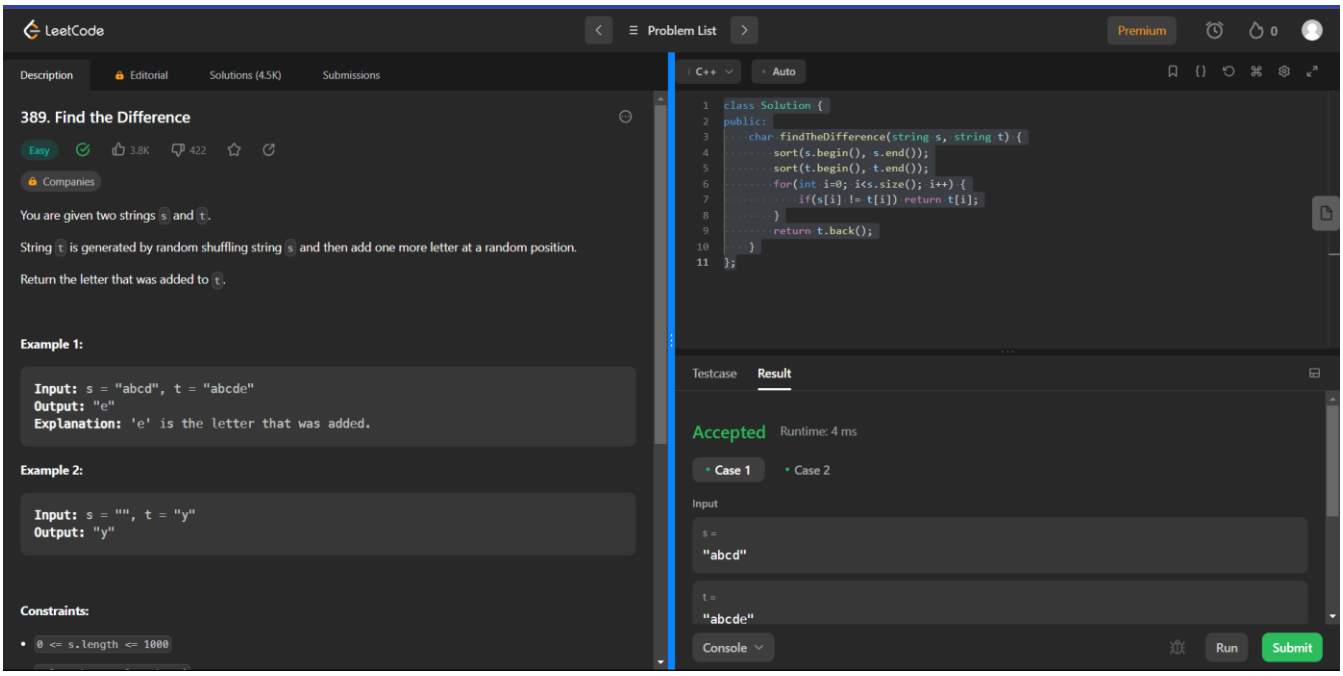
1) Find the difference

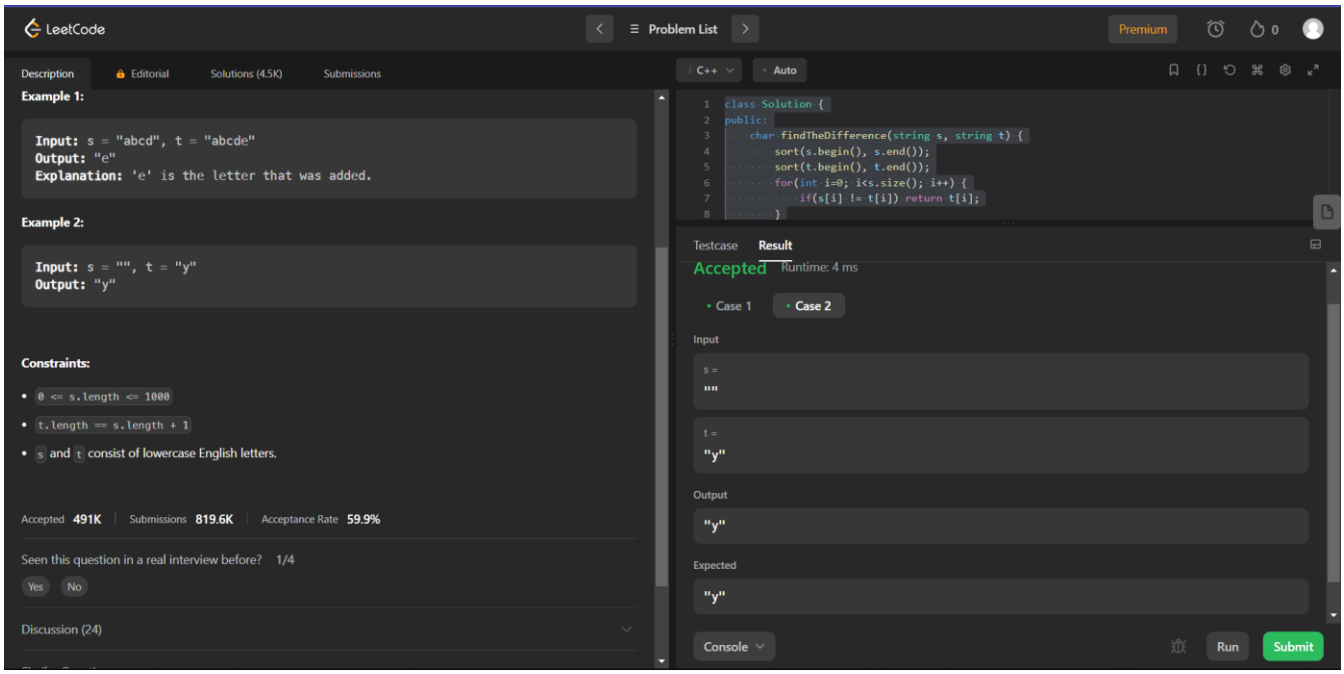
https://leetcode.com/problems/find-the-difference/

Code –

```
class Solution {
public:
    char findTheDifference(string s, string t) {
        sort(s.begin(), s.end());
        sort(t.begin(), t.end());
        for(int i=0; i<s.size(); i++) {
            if(s[i] != t[i]) return t[i];
        }
        return t.back();
    }
};
```

Output -


LeetCode
Problem List
Premium
Description
Editorial
Solutions (4.5K)
Submissions
C++
Auto
389. Find the Difference
Easy
3.8K
422
Companies
You are given two strings s and t.
String t is generated by random shuffling string s and then add one more letter at a random position.
Return the letter that was added to t.
Example 1:
Input: s = "abcd", t = "abcde"
Output: "e"
Explanation: 'e' is the letter that was added.
Example 2:
Input: s = "", t = "y"
Output: "y"
Constraints:
0 <= s.length <= 1000
class Solution {
public:
char findTheDifference(string s, string t) {
sort(s.begin(), s.end());
sort(t.begin(), t.end());
for(int i=0; i<s.size(); i++) {
if(s[i] != t[i]) return t[i];
}
return t.back();
}
};
Testcase
Result
Accepted
Runtime: 4 ms
Case 1
Case 2
Input
s =
"abcd"
t =
"abcde"
Console
Run
Submit


LeetCode
Problem List
Premium
Description
Editorial
Solutions (4.5K)
Submissions
C++
Auto
Example 1:
Input: s = "abcd", t = "abcde"
Output: "e"
Explanation: 'e' is the letter that was added.
Example 2:
Input: s = "", t = "y"
Output: "y"
Constraints:
0 <= s.length <= 1000
t.length == s.length + 1
s and t consist of lowercase English letters.
Accepted 491K
Submissions 819.6K
Acceptance Rate 59.9%
Seen this question in a real interview before? 1/4
Yes
No
Discussion (24)
class Solution {
public:
char findTheDifference(string s, string t) {
sort(s.begin(), s.end());
sort(t.begin(), t.end());
for(int i=0; i<s.size(); i++) {
if(s[i] != t[i]) return t[i];
}
Testcase
Result
Accepted
Runtime: 4 ms
Case 1
Case 2
Input
s =
""
t =
"y"
Output
"y"
Expected
"y"
Console
Run
Submit

2) Predict the winner

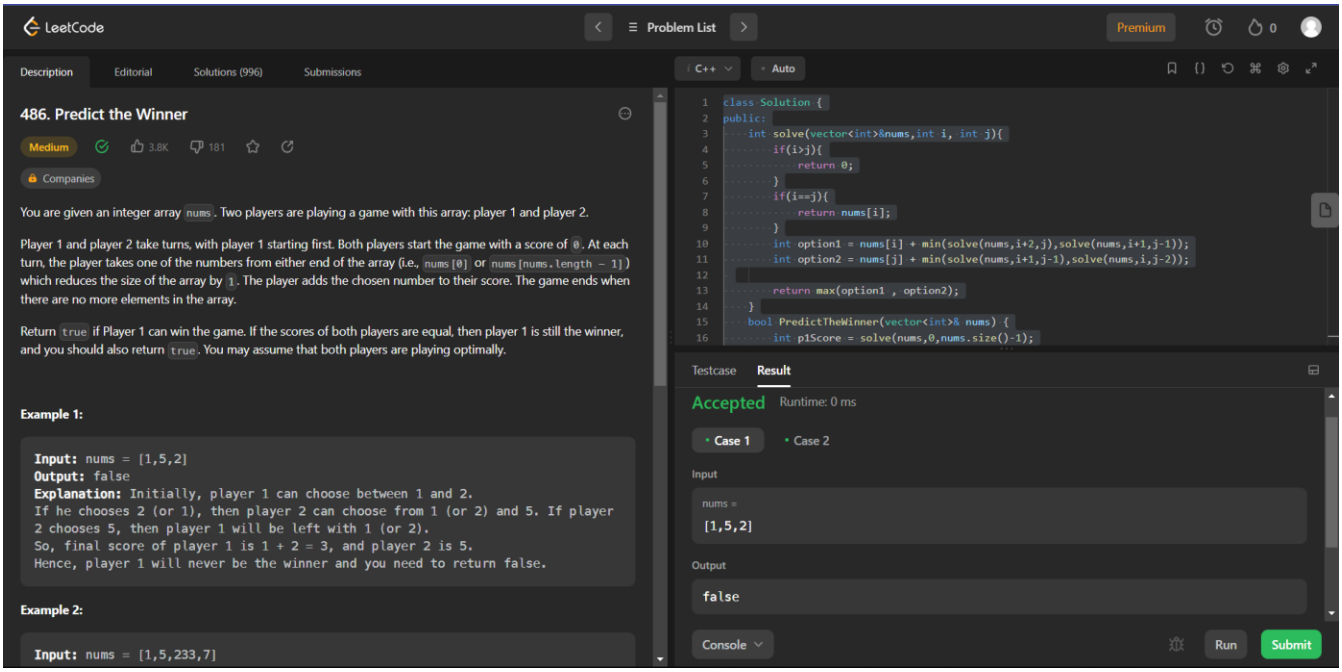
https://leetcode.com/problems/predict-the-winner/

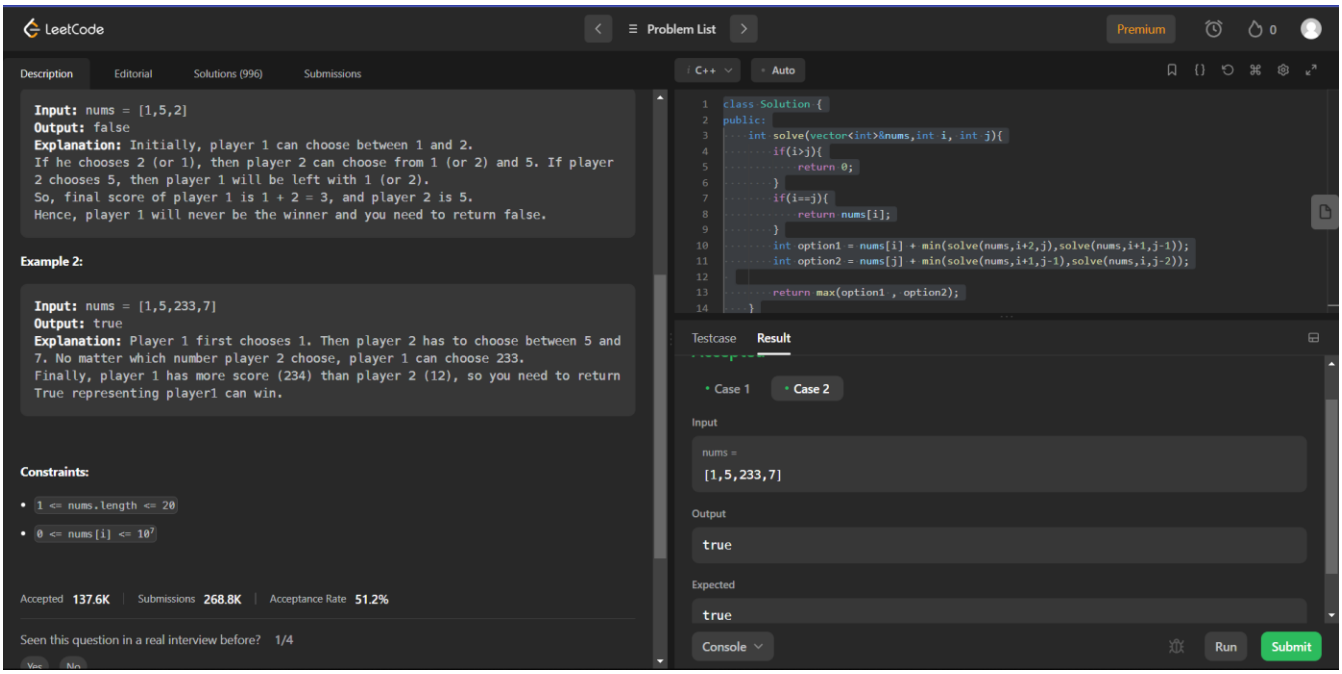
Code -

```
class Solution {
public:
    int solve(vector<int>&nums,int i, int j){
        if(i>j){
            return 0;
        }
        if(i==j){
            return nums[i];
        }
        int option1 = nums[i] + min(solve(nums,i+2,j),solve(nums,i+1,j-1));
        int option2 = nums[j] + min(solve(nums,i+1,j-1),solve(nums,i,j-2));

        return max(option1 , option2);
    }
    bool PredictTheWinner(vector<int>& nums) {
        int p1Score = solve(nums,0,nums.size()-1);
        int total_Score = 0;
        for(int i=0;i<nums.size();i++){
            total_Score += nums[i];
        }
        int p2Score = total_Score - p1Score;
        if(p1Score>=p2Score){
            return true;
        }
        else{
            return false;
        }
    }
};
```

Output –


LeetCode
Problem List
Premium
Description
Editorial
Solutions (996)
Submissions
C++
Auto
486. Predict the Winner
Medium
3.8K
181
Companies
You are given an integer array nums. Two players are playing a game with this array: player 1 and player 2.
Player 1 and player 2 take turns, with player 1 starting first. Both players start the game with a score of 0. At each turn, the player takes one of the numbers from either end of the array (i.e., nums[0] or nums[nums.length - 1]) which reduces the size of the array by 1. The player adds the chosen number to their score. The game ends when there are no more elements in the array.
Return true if Player 1 can win the game. If the scores of both players are equal, then player 1 is still the winner, and you should also return true. You may assume that both players are playing optimally.
Example 1:
Input: nums = [1,5,2]
Output: false
Explanation: Initially, player 1 can choose between 1 and 2.
If he chooses 2 (or 1), then player 2 can choose from 1 (or 2) and 5. If player 2 chooses 5, then player 1 will be left with 1 (or 2).
So, final score of player 1 is 1 + 2 = 3, and player 2 is 5.
Hence, player 1 will never be the winner and you need to return false.
Example 2:
Input: nums = [1,5,233,7]
class Solution {
public:
int solve(vector<int>&nums,int i, int j){
if(i>j){
return 0;
}
if(i==j){
return nums[i];
}
int option1 = nums[i] + min(solve(nums,i+2,j),solve(nums,i+1,j-1));
int option2 = nums[j] + min(solve(nums,i+1,j-1),solve(nums,i,j-2));
return max(option1 , option2);
}
bool PredictTheWinner(vector<int>& nums) {
int p1Score = solve(nums,0,nums.size()-1);
Testcase
Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[1,5,2]
Output
false
Console
Run
Submit


LeetCode
Problem List
Premium
Description
Editorial
Solutions (996)
Submissions
C++
Auto
Input: nums = [1,5,2]
Output: false
Explanation: Initially, player 1 can choose between 1 and 2.
If he chooses 2 (or 1), then player 2 can choose from 1 (or 2) and 5. If player 2 chooses 5, then player 1 will be left with 1 (or 2).
So, final score of player 1 is 1 + 2 = 3, and player 2 is 5.
Hence, player 1 will never be the winner and you need to return false.
Example 2:
Input: nums = [1,5,233,7]
Output: true
Explanation: Player 1 first chooses 1. Then player 2 has to choose between 5 and 7. No matter which number player 2 choose, player 1 can choose 233.
Finally, player 1 has more score (234) than player 2 (12), so you need to return True representing player1 can win.
Constraints:
1 <= nums.length <= 20
0 <= nums[i] <= 10^7
Accepted 137.6K
Submissions 268.8K
Acceptance Rate 51.2%
Seen this question in a real interview before? 1/4
class Solution {
public:
int solve(vector<int>&nums,int i, int j){
if(i>j){
return 0;
}
if(i==j){
return nums[i];
}
int option1 = nums[i] + min(solve(nums,i+2,j),solve(nums,i+1,j-1));
int option2 = nums[j] + min(solve(nums,i+1,j-1),solve(nums,i,j-2));
return max(option1 , option2);
}
Testcase
Result
Case 1
Case 2
Input
nums =
[1,5,233,7]
Output
true
Expected
true
Console
Run
Submit