

## 486. Predict the Winner

Medium 4.8K 224

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`.

i Java Auto

```
1 class Solution {
2     public int solve(int[][] dp,int []v,int i,int j,int k){
3         if(i>j)return 0;
4         if(dp[i][j]!=-1)return dp[i][j];
5         int first=0,last=0,ans=0;
6         if(k%2==0){
7             first=v[i]+solve(dp,v,i+1,j,k+1);
8             last=v[j]+solve(dp,v,i,j-1,k+1);
9             ans=Math.max(first,last);
10        }
11        else{
12            first=-v[i]+solve(dp,v,i+1,j,k+1);
13            last=-v[j]+solve(dp,v,i,j-1,k+1);
14            ans=Math.min(first,last);
15        }
16        return dp[i][j]=ans;
17    }
18    public boolean PredictTheWinner(int[] v) {
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =  
[1,5,2]

Output

false

Console



Run

Submit

[Description](#)[Editorial](#)[Solutions \(4.5K\)](#)[Submissions](#)

## 412. Fizz Buzz

Easy

2.2K

293



Companies

Given an integer `n`, return a string array `answer` (**1-indexed**) where:

- `answer[i] == "FizzBuzz"` if `i` is divisible by 3 and 5.
- `answer[i] == "Fizz"` if `i` is divisible by 3.
- `answer[i] == "Buzz"` if `i` is divisible by 5.
- `answer[i] == i` (as a string) if none of the above conditions are true.

### Example 1:

**Input:** `n = 3`**Output:** `["1","2","Fizz"]`

### Example 2:

**Input:** `n = 5`**Output:** `["1","2","Fizz","4","Buzz"]`

### Example 3:

**Input:** `n = 15`**Output:**  
`["1","2","Fizz","4","Buzz","Fizz","7","8","Fizz","Buzz","11","Fizz","13","14","FizzBuzz"]`

Java

Auto

```
1 class Solution {
2     public List<String> fizzBuzz(int n) {
3         List<String> arr = new ArrayList<>();
4         for(int i=1;i<n+1;i++){
5             if(i%3==0 && i%5==0)
6                 arr.add("FizzBuzz");
7             else if(i%5==0)
8                 arr.add("Buzz");
9             else if(i%3 == 0)
10                arr.add("Fizz");
11             else{
12                 String s =String.valueOf(i);
13                 arr.add(s);
14             }
15         }
16         return arr;
17     }
18 }
```

Testcase

Result

Accepted Runtime: 0 ms

Case 1

Case 2

Case 3

Input

`n =``3`

Output

`["1","2","Fizz"]`

Console



Run

Submit