

486 Predict The Winner

Given an array of scores that are non-negative integers. Player 1 picks one of the numbers from either end of the array followed by the player 2 and then player 1 and so on. Each time a player picks a number, that number will not be available for the next player. This continues until all the scores have been chosen. The player with the maximum score wins.

Given an array of scores, predict whether player 1 is the winner. You can assume each player plays to maximize his score.

Example 1:

Input: [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: [1, 5, 233, 7]

Output: True

Explanation: Player 1 first chooses 1. Then player 2 have 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.

Note:

1. $1 \leq \text{length of the array} \leq 20$.
2. Any scores in the given array are non-negative integers and will not exceed 10,000,000.
3. If the scores of both players are equal, then player 1 is still the winner.

Intution :

Both players are playing **smartly**

let $A = \{1, 5, 9, 3\}$ // let start index = s , end index = e

score of players :

choice1 = $A[s] + \min(\text{findScore}(A, s+2, e), \text{findScore}(A, s+1, e-1))$; // player1 selects $A[s]$ and then player 2 will maximise his score such that player1 is left with min of remaining

choice2 = $A[e] + \min(\text{findScore}(A, s+1, e-1), \text{findScore}(A, s, e-2))$; //player1 selects $A[e]$ and then player 2 will maximise his score such that player1 is left with min of remaining

player1 score = $\max(\text{choice1}, \text{choice2})$;

CODE :

```
int dp[21][21];
```

```
Solution(){ //constructor
```

```
    for(int i=0; i<21; i++)
```

```
        for(int j=0; j<21; j++)
```

```
            dp[i][j] = -1;
```

```
}
```

```
int findScore(vector<int>& nums, int s, int e){
```

```
    if(s > e) return 0;
```

```
    if(dp[s][e] != -1) return dp[s][e];
```

```
    dp[s][e] = max(nums[s] + min(findScore(nums, s+2, e), findScore(nums, s+1, e-1)), nums[e] + min(findScore(nums, s+1, e-1), findScore(nums, s, e-2)));
```

```
    return dp[s][e];
```

```
}
```

```
bool PredictTheWinner(vector<int>& nums) {
```

```
    if(nums.size() <= 1) return true;
```

```
    return 2*findScore(nums, 0, nums.size()-1) >= accumulate(nums.begin(), nums.end(), 0);
```

}