Experiment-6

Student Name: Nikhil Niraj Singh
UID: 22BCS10133
Branch: BE-CSE
Section/Group: KPIT-901/B
Semester: 6th
Date of Performance: 8/03/25
Subject Name: Advanced Programming Lab - 2
Subject Code: 22CSP-351

1. Aim:

1. Problem: 6.1: To implement and analyze the Kadane's algorithm to find the contiguous subarray with the maximum sum in a given integer array.
2. Problem: 6.2: To determine if it is possible to reach the last index of an array by making jumps based on given values.

2. Objective:

1. Problem 6.1: To understand and implement Kadane's algorithm for solving the maximum subarray problem. To analyze the time complexity of the algorithm and optimize it for efficiency.
2. Problem 6.2: To implement a greedy approach or dynamic programming technique to solve the Jump Game problem. To analyze the problem constraints and optimize the solution for efficiency.

3. Implementation/Code:

1.)

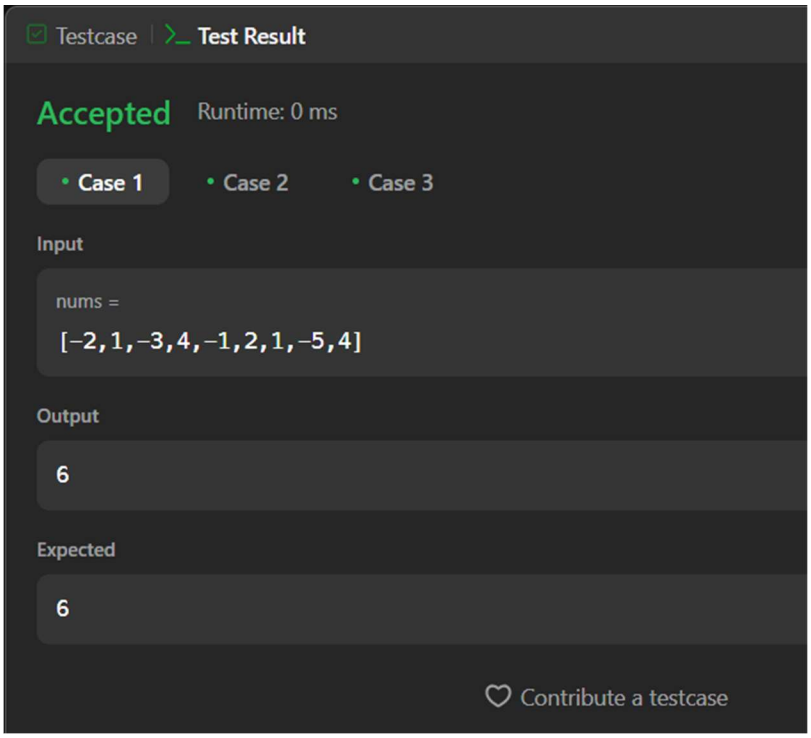
```
class Solution {
 public:
  int maxSubArray(vector<int>& nums) {
    // dp[i] := the maximum sum subarray ending in i
    vector<int> dp(nums.size());

    dp[0] = nums[0];
    for (int i = 1; i < nums.size(); ++i)
      dp[i] = max(nums[i], dp[i - 1] + nums[i]);

    return ranges::max(dp);
  }
};
```

2.)

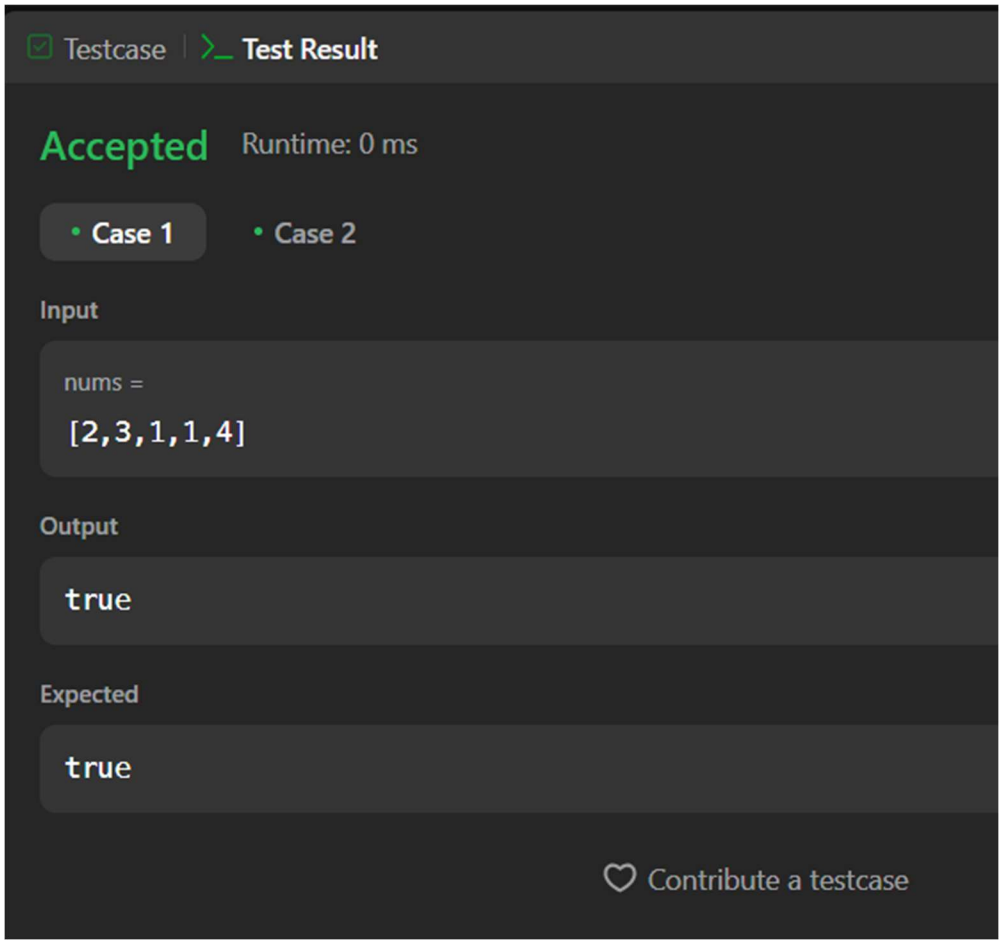
```
class Solution {
 public:
  bool canJump(vector<int>& nums) {
    int i = 0;

    for (int reach = 0; i < nums.size() && i <= reach; ++i)
      reach = max(reach, i + nums[i]);

    return i == nums.size();
  }
};
```

4. Output:

1.



2.


Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[2,3,1,1,4]
Output
true
Expected
true
Contribute a testcase

5. Learning Outcome:

1. Ability to apply Kadane's algorithm for solving subarray sum problems.
2. Understanding the importance of dynamic programming in optimizing array problems.
3. Understanding the application of greedy algorithms in pathfinding problems.
4. Ability to determine the feasibility of reaching a target index using given jump constraints.