

Implement Stack using an Array

```
class Solution {
public:
    vector<string> buildArray(vector<int>& target, int n) {
        vector<string> operations;
        int currNum = 1;
        int index = 0;

        while (currNum <= n && index < target.size()) {
            int num = target[index];
            if (num == currNum) {
                operations.push_back("Push");
                currNum++;
                index++;
            } else {
                operations.push_back("Push");
                operations.push_back("Pop");
                currNum++;
            }
        }

        return operations;
    }
};
```

The screenshot displays a coding platform interface with a problem list on the left, a code editor in the center, and a test result panel on the right.

Problem List: The left sidebar shows the problem "Implement Stack using an Array" as "Accepted". It indicates that 49 out of 49 test cases passed. The user "12ananya" submitted the solution on Mar 16, 2025, at 17:26. There are buttons for "Editorial" and "Solution".

Performance Metrics: The "Runtime" section shows "0 ms" and "Beats 100.00%". The "Memory" section shows "10.63 MB" and "Beats 82.89%". Below these is a bar chart showing the distribution of runtime performance across different time intervals (1ms, 2ms, 3ms, 4ms).

Code Editor: The center panel shows the C++ code for the solution. The code is as follows:

```
1 class Solution {
2 public:
3     vector<string> buildArray(vector<int>& target, int n) {
4         vector<string> operations;
5         int currNum = 1;
6         int index = 0;
7
8         while (currNum <= n && index < target.size()) {
9             int num = target[index];
10            if (num == currNum) {
11                operations.push_back("Push");
12                currNum++;
13                index++;
14            } else {
15                operations.push_back("Push");
16                operations.push_back("Pop");
17                currNum++;
18            }
19        }
20        return operations;
21    }
22 };
```

Test Result: The right panel shows the test result, which is "Accepted". It indicates a runtime of 0 ms. There are three test cases: Case 1, Case 2, and Case 3. The input for Case 2 is shown as:

```
target = [1,2,3]
```