

Winter Domain Camp Day-5

Name-Abhiraj Patel

UID-22BCS11329

Date-26-12-24

Q.1. Find Minimum in the rotated sorted array

```
#include <iostream>

#include <vector>

using namespace std;

int findMin(vector<int>& nums) {
    int left = 0, right = nums.size() - 1;
    while (left < right) {
        int mid = left + (right - left) / 2;
        if (nums[mid] > nums[right])
            left = mid + 1;
        else
            right = mid;
    }
    return nums[left];
}

int main() {
    vector<int> nums = {4, 5, 6, 7, 0, 1, 2};
    cout << "Minimum element: " << findMin(nums) << endl;
    return 0;
}
```

Output

```
Minimum element: 0
PS A:\S2D\Coding\C++\wintercamp\Day5>
```

Q2 Median of 2 sorted array

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
```

```
    if (nums1.size() > nums2.size()) swap(nums1, nums2);
```

```
    int x = nums1.size(), y = nums2.size();
```

```
    int low = 0, high = x;
```

```
    while (low <= high) {
```

```
        int partitionX = (low + high) / 2;
```

```
        int partitionY = (x + y + 1) / 2 - partitionX;
```

```
        int maxX = (partitionX == 0) ? INT_MIN : nums1[partitionX - 1];
```

```
        int minX = (partitionX == x) ? INT_MAX : nums1[partitionX];
```

```
        int maxY = (partitionY == 0) ? INT_MIN : nums2[partitionY - 1];
```

```
        int minY = (partitionY == y) ? INT_MAX : nums2[partitionY];
```

```
        if (maxX <= minY && maxY <= minX) {
```

```
            if ((x + y) % 2 == 0) {
```

```
                return (max(maxX, maxY) + min(minX, minY)) / 2.0;
```

```
            } else {
```

```
                return max(maxX, maxY);
```

```
            }
```

```
        } else if (maxX > minY) {
```

```
            high = partitionX - 1;
```

```
        } else {
```

```
            low = partitionX + 1;
```

```
        }
```

```
    }
```

```

        return -1.0; // Error case
    }

int main() {
    vector<int> nums1 = {1, 3};
    vector<int> nums2 = {2};
    cout << "Median: " << findMedianSortedArrays(nums1, nums2) << endl;
    return 0;
}

```

Output:

```

Median: 2
PS A:\S2D\Coding\C++\wintercamp\Day5>

```

Q.3 Merge K sorted lists

```

#include <iostream>

#include <vector>

#include <queue>

using namespace std;

struct ListNode {
    int val;
    ListNode* next;
    ListNode(int x) : val(x), next(nullptr) {}
};

struct Compare {
    bool operator()(ListNode* a, ListNode* b) {
        return a->val > b->val;
    }
};

```

```

ListNode* mergeKLists(vector<ListNode*>& lists) {
    priority_queue<ListNode*, vector<ListNode*>, Compare> pq;
    for (auto list : lists) {
        if (list) pq.push(list);
    }
    ListNode dummy(0), *tail = &dummy;
    while (!pq.empty()) {
        ListNode* node = pq.top();
        pq.pop();
        tail->next = node;
        tail = tail->next;
        if (node->next) pq.push(node->next);
    }
    return dummy.next;
}

```

Output:

Accepted 134 / 134 testcases passed
 Abhiraj Patel submitted at Dec 26, 2024 23:28

Runtime: 0 ms | Beats 100.00%
 Memory: 18.53 MB | Beats 46.80%

```

1 #include <vector>
2 using namespace std;
3 class Solution {
4 public:
5     ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
6         if (!l1) return l2;
7         if (!l2) return l1;
8
9         if (l1->val < l2->val) {
10             l1->next = mergeTwoLists(l1->next, l2);
11             return l1;
12         } else {
13             l2->next = mergeTwoLists(l1, l2->next);
14             return l2;
15         }
16     }
17 }

```

Q.4 Minimum number of moves to seat

```

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

```

```

int minMovesToSeat(vector<int>& seats, vector<int>& students) {

    sort(seats.begin(), seats.end());

    sort(students.begin(), students.end());

    int moves = 0;

    for (int i = 0; i < seats.size(); ++i) {

        moves += abs(seats[i] - students[i]);

    }

    return moves;

}

int main() {

    vector<int> seats = {3, 1, 5};

    vector<int> students = {2, 7, 4};

    cout << "Minimum moves: " << minMovesToSeat(seats, students) << endl;

    return 0;

}

```

Output:



The screenshot displays a LeetCode submission page. On the left, a status bar indicates 'Accepted' with '262 / 262 testcases passed'. Below this, the user 'Abhiraj Patel' is noted as having submitted the solution on 'Dec 26, 2024 23:31'. A 'Runtime' section shows a performance of '4 ms' and 'Beats 9.10%', with a link to 'Analyze Complexity'. A 'Memory' section shows '21.65 MB' and 'Beats 37.37%'. On the right, the C++ code for the 'minMovesToSeat' function is displayed within a 'Solution' class, matching the code provided in the previous block.

Q5. Search a Number

```

#include<iostream.h>

using namespace std;

int search(vector<int>& nums, int target) {

    int l=0,r=nums.size()-1;

    while(l<=r)

```

```

{
    int mid=l+(r-l)/2;

    if(nums[mid]==target) return mid;

    else if(nums[mid]>target) r=mid-1;

    else l=mid+1;

}

return -1;

}

```

```

int main() {

    vector<int> arr = {9, 7, 16, 16, 4};

    int k = 16;

    cout << "Position of " << k << ": " << search(arr, k) << endl;

    return 0;

}

```

Output:

The screenshot shows a code editor interface with a dark theme. On the left, a sidebar displays the submission status: 'Accepted' with '47 / 47 testcases passed'. Below this, it says 'Abhiraj Patel submitted at Dec 26, 2024 23:35'. There are buttons for 'Editorial' and 'Solution'. The 'Runtime' section shows '0 ms' and 'Beats 100.00%' with a green leaf icon. Below that is a link to 'Analyze Complexity'. The 'Memory' section shows '31.24 MB' and 'Beats 40.29%'. On the right, the code is displayed in a C++ class structure:

```

1  class Solution {
2  public:
3      int search(vector<int>& nums, int target) {
4          int l=0,r=nums.size()-1;
5          while(l<=r)
6          {
7              int mid=l+(r-l)/2;
8              if(nums[mid]==target) return mid;
9              else if(nums[mid]>target) r=mid-1;
10             else l=mid+1;
11         }
12         return -1;
13     }
14 };

```