Arrays and Two Pointer

1 . [https://practice.geeksforgeeks.org/problems/equilibrium-point-1587115620/1/?problemStatus=solved&page=2&query=problemStatussolvedpage2#](https://practice.geeksforgeeks.org/problems/equilibrium-point-1587115620/1/?problemStatus=solved&page=2&query=problemStatussolvedpage2)

int equilibriumPoint(long long arr[], int n) {

// Your code here

int total\_sum = 0;

for(int i=0; i<n; i++)

{

total\_sum += arr[i];

}

int left\_sum = 0;

for(int i=0; i<n; i++)

{

total\_sum -= arr[i];

if(total\_sum == left\_sum)

{

return i+1;

}

left\_sum += arr[i];

}

return -1;

}

2. [https://practice.geeksforgeeks.org/problems/max-sum-without-adjacents2430/1#](https://practice.geeksforgeeks.org/problems/max-sum-without-adjacents2430/1)

int findMaxSum(int \*arr, int n) {

// code here

if(n==1)

{

return arr[0];

}

if(n==2)

{

return max(arr[0], arr[1]);

}

int a = arr[0];

int b = max(arr[0], arr[1]);

int c;

for(int i=2; i<n; i++)

{

c = max(b, arr[i]+a);

a = b;

b = c;

}

return c;

}

3. <https://leetcode.com/problems/sort-colors/>

class Solution {

public:

void sortColors(vector<int>& nums) {

int n = nums.size();

int i = 0;

int low = 0;

int high = n-1;

while(i<=high)

{

if(nums[i]==0)

{

swap(nums[i], nums[low]);

i++;

low++;

}

else if(nums[i]==1)

{

i++;

}

else

{

swap(nums[i], nums[high]);

high--;

}

}

}

};