**First and Last Occurrence**

class Solution

{

public:

vector<int> find(int arr[], int n , int x )

{

vector<int>ans;

int first = -1, last = -1;

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

if (x != arr[i])

continue;

if (first == -1)

first = i;

last = i;

}

if (first != -1){

ans.push\_back(first);

ans.push\_back(last);

}

else{

ans.push\_back(-1);

ans.push\_back(-1);

}

return ans;

}

};

**TC: O(N)**

**SC: O(1)**

**Find transition point**

class Solution

{

public:

vector<int> find(int arr[], int n , int x )

{

vector<int>ans;

int first = -1, last = -1;

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

if (x != arr[i])

continue;

if (first == -1)

first = i;

last = i;

}

if (first != -1){

ans.push\_back(first);

ans.push\_back(last);

}

else{

ans.push\_back(-1);

ans.push\_back(-1);

}

return ans;

}

};

**TC: O(N)**

**SC: O(1)**

**First Repeating Element**

class Solution

{

public:

vector<int> find(int arr[], int n , int x )

{

vector<int>ans;

int first = -1, last = -1;

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

if (x != arr[i])

continue;

if (first == -1)

first = i;

last = i;

}

if (first != -1){

ans.push\_back(first);

ans.push\_back(last);

}

else{

ans.push\_back(-1);

ans.push\_back(-1);

}

return ans;

}

};

**TC: O(2N)**

**SC: O(N)**

**Remove Duplicates from Sorted Array**

class Solution

{

public:

vector<int> find(int arr[], int n , int x )

{

vector<int>ans;

int first = -1, last = -1;

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

if (x != arr[i])

continue;

if (first == -1)

first = i;

last = i;

}

if (first != -1){

ans.push\_back(first);

ans.push\_back(last);

}

else{

ans.push\_back(-1);

ans.push\_back(-1);

}

return ans;

}

};

**TC: O(N)**

**SC: O(1)**