1. Deque implementations

// Function to push element x to the back of the deque.

void push\_back\_pb(deque<int> &dq, int x) {

dq.push\_back(x);

}

// Function to pop element from back of the deque.

void pop\_back\_ppb(deque<int> &dq) {

if (!dq.empty())

dq.pop\_back();

else return;

}

// Function to return element from front of the deque.

int front\_dq(deque<int> &dq) {

if (!dq.empty())

return dq.front();

else return -1;

}

// Function to push element x to the front of the deque.

void push\_front\_pf(deque<int> &dq, int x) {

dq.push\_front(x);

}

1. Maximum of all subarrays of size k

vector <int> max\_of\_subarrays(int \*arr, int n, int k)

{

deque<int> dq;

vector<int> ans;

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

{

while (!dq.empty() && arr[i] >= arr[dq.back()])

dq.pop\_back();

dq.push\_back(i);

}

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

{

ans.push\_back(arr[dq.front()]);

while ((!dq.empty()) && dq.front() <= i - k)

dq.pop\_front();

while ((!dq.empty()) && arr[i] >= arr[dq.back()])

dq.pop\_back();

dq.push\_back(i);

}

ans.push\_back(arr[dq.front()]);

return ans;

}

1. Deque traversal

void printDeque(deque<int> Deq)

{

int n = Deq.size();

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

{

cout<<Deq.front()<<" ";

Deq.push\_back(Deq.front());

Deq.pop\_front();

}

cout<<endl;

}

1. Rotate deque by k

//Function to rotate deque by k places in anti-clockwise direction.

void left\_Rotate\_Deq\_ByK(deque<int> &deq, int n, int k)

{

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

{

deq.push\_back(deq.front());

deq.pop\_front();

}

}

//Function to rotate deque by k places in clockwise direction.

void right\_Rotate\_Deq\_ByK(deque<int> &deq, int n, int k)

{

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

{

deq.push\_front(deq.back());

deq.pop\_back();

}

}

1. Insertion in deque

deque<int> deque\_Init(int arr[], int n) {

deque<int> ans;

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

ans.push\_back(arr[i]);

return ans;

}

1. Deque deletion

//Function to erase the element from specified position X in deque.

void eraseAt(deque <int> &deq, int X)

{

auto it = deq.begin();

deq.erase(it+X);

}

//Function to erase the elements in range start (inclusive), end (exclusive).

void eraseInRange(deque<int> &deq, int start, int end)

{

auto it = deq.begin();

deq.erase(it+start, it+end);

}

//Function to erase all the elements in the deque.

void eraseAll(deque<int> &deq)

{

deq.clear();

}