**Assignment16**

**1.**

function findNearestGreaterFrequency(arr) {

const n = arr.length;

const stack = [];

const frequency = {};

const result = Array(n).fill(-1);

for (let i = n - 1; i >= 0; i--) {

const num = arr[i];

frequency[num] = (frequency[num] || 0) + 1;

while (stack.length > 0 && frequency[arr[stack[stack.length - 1]]] <= frequency[num]) {

const index = stack.pop();

result[index] = num;

}

stack.push(i);

}

return result;

}

Example:

const arr1 = [1, 1, 2, 3, 4, 2, 1];

console.log(findNearestGreaterFrequency(arr1));

Output: [-1, -1, 1, 2, 2, 1, -1]

const arr2 = [1, 1, 1, 2, 2, 2, 2, 11, 3, 3];

console.log(findNearestGreaterFrequency(arr2));

Output: [2, 2, 2, -1, -1, -1, -1, 3, -1, -1]

**2.**

function sortStack(stack) {

const tempStack = [];

while (stack.length > 0) {

const temp = stack.pop();

while (tempStack.length > 0 && tempStack[tempStack.length - 1] > temp) {

stack.push(tempStack.pop());

}

tempStack.push(temp);

}

while (tempStack.length > 0) {

stack.push(tempStack.pop());

}

return stack;

}

Example:

const stack1 = [34, 3, 31, 98, 92, 23];

console.log(sortStack(stack1));

Output: [3, 23, 31, 34, 92, 98]

const stack2 = [3, 5, 1, 4, 2, 8];

console.log(sortStack(stack2));

Output: [1, 2, 3, 4, 5, 8]

**3.**

function deleteMiddle(stack) {

if (stack.length === 0) {

return stack;

}

const size = Math.floor(stack.length / 2) + 1;

deleteMiddleUtil(stack, size);

return stack;

}

function deleteMiddleUtil(stack, size) {

if (size === 1) {

stack.pop();

return;

}

const temp = stack.pop();

deleteMiddleUtil(stack, size - 1);

stack.push(temp);

}

Example:

const stack1 = [1, 2, 3, 4, 5];

console.log(deleteMiddle(stack1));

Output: [1, 2, 4, 5]

const stack2 = [1, 2, 3, 4, 5, 6];

console.log(deleteMiddle(stack2));

Output: [1, 2, 4, 5, 6]

**4.**

function canArrangeInIncreasingOrder(queue) {

const stack = [];

const secondQueue = [];

let expected = 1;

while (queue.length > 0) {

if (queue[0] === expected) {

queue.shift();

expected++;

} else if (stack.length > 0 && stack[stack.length - 1] === expected) {

stack.pop();

expected++;

} else {

stack.push(queue.shift());

}

}

while (stack.length > 0 && stack[stack.length - 1] === expected) {

stack.pop();

expected++;

}

return queue.length === 0 && stack.length === 0 ? "Yes" : "No";

}

Example:

const queue1 = [5, 1, 2, 3, 4];

console.log(canArrangeInIncreasingOrder(queue1));

Output: Yes

const queue2 = [5, 1, 2, 6, 3, 4];

console.log(canArrangeInIncreasingOrder(queue2));

Output: No

**5.**

function reverseNumber(num) {

const numStr = String(num);

const stack = [];

for (let i = 0; i < numStr.length; i++) {

stack.push(numStr[i]);

}

let reversedStr = '';

while (stack.length > 0) {

reversedStr += stack.pop();

}

const reversedNum = parseInt(reversedStr);

return reversedNum;

}

Example:

console.log(reverseNumber(365));

Output: 563

console.log(reverseNumber(6899));

Output: 9986

**6.**

function reverseKElements(k, queue) {

const stack = [];

const tempQueue = [];

for (let i = 0; i < k; i++) {

stack.push(queue.dequeue());

}

while (queue.size() > 0) {

tempQueue.push(queue.dequeue());

}

while (stack.length > 0) {

tempQueue.push(stack.pop());

}

while (tempQueue.length > 0) {

queue.enqueue(tempQueue.shift());

}

}

Example:

const queue = {

data: [1, 2, 3, 4, 5],

enqueue: function (x) {

this.data.push(x);

},

dequeue: function () {

return this.data.shift();

},

size: function () {

return this.data.length;

},

front: function () {

return this.data[0];

}

};

reverseKElements(3, queue);

console.log(queue.data);

Output: [3, 2, 1, 4, 5]

**7.**

function countRemainingWords(sequence) {

const stack = [];

for (let i = 0; i < sequence.length; i++) {

const word = sequence[i];

if (stack.length === 0) {

stack.push(word);

} else {

const top = stack[stack.length - 1];

if (top === word) {

stack.pop();

} else {

stack.push(word);

}

}

}

return stack.length;

}

Example:

const sequence = ['ab', 'aa', 'aa', 'bcd', 'ab'];

const remainingWords = countRemainingWords(sequence);

console.log(remainingWords);

Output: 3

const sequence = ['tom', 'jerry', 'jerry', 'tom'];

const remainingWords = countRemainingWords(sequence);

console.log(remainingWords);

Output: 0

**8.**

function maxAbsoluteDiff(arr) {

const n = arr.length;

const LS = new Array(n).fill(0);

const stack = [];

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

while (stack.length > 0 && stack[stack.length - 1] >= arr[i]) {

stack.pop();

}

LS[i] = stack.length > 0 ? stack[stack.length - 1] : 0;

stack.push(arr[i]);

}

const RS = new Array(n).fill(0);

stack.length = 0; // Clear the stack

for (let i = n - 1; i >= 0; i--) {

while (stack.length > 0 && stack[stack.length - 1] >= arr[i]) {

stack.pop();

}

RS[i] = stack.length > 0 ? stack[stack.length - 1] : 0;

stack.push(arr[i]);

}

let maxDiff = 0;

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

const diff = Math.abs(LS[i] - RS[i]);

maxDiff = Math.max(maxDiff, diff);

}

return maxDiff;

}

Example:

const arr1 = [2, 1, 8];

console.log(maxAbsoluteDiff(arr1));

Output: 1

const arr2 = [2, 4, 8, 7, 7, 9, 3];

console.log(maxAbsoluteDiff(arr2));

Output: 4

const arr3 = [5, 1, 9, 2, 5, 1, 7];

console.log(maxAbsoluteDiff(arr3));

Output: 1