**DSA Practice Test – 7** 9th Nov 2024

**1. Minimum Path Sum:**

**Code:**

**Output**:

**Time Complexity:** O (n)

**Space Complexity:** O (1)

**2. Validate Binary Search Tree**

**Code:**

**Output:**

**Time Complexity:** O (n)

**Space Complexity:** O (1)

**3. Word Ladder**

**Code:**

**Output:**

**Time Complexity:** O (n)  
**Space Complexity:** O (1)

**4. Word Ladder - II**

**Code:**

**Output:**

**Time Complexity:** O (n log n)  
**Space Complexity:** O (1)

**5. Course Schedule**

**Code:**

**Output:**

**Time Complexity:** O (n)

**Space Complexity:** O (1)

**6. Design Tic Tac Toe**

**Code:**

**Output:**

**Time Complexity:** O (n log n)

**Space Complexity:** O (1)

**7. Next Permutation**

**Code:**

**Output:**

**Time Complexity:** O (n)

**Space Complexity:** O (1)

**8. Spiral Matrix**

**Code:**

**Output:**

**Time Complexity:** O (n)

**Space Complexity:** O (1)

**9. Longest Substring Without Repeating Characters**

**Code:**

**Output:**

**Time Complexity:** O(n2)

**Space Complexity:** O (1)

**10. Remove Linked List Elements**

**Code:**

public class SpiralMat {

static void spir(int[][] arr){

int m=arr.length;

int n=arr[0].length;

int top=0, bot=m-1, l=0, r=n-1;

while(top<=bot&&l<=r){

for(int i=l;i<=r;i++){

System.out.print(arr[top][i] + " ");

}

top++;

for(int i=top;i<=bot;i++){

System.out.print(arr[i][r] + " ");

}

r--;

if(top <= bot){

for(int i=r;i>=l;i--){

System.out.print(arr[bot][i] + " ");

}

bot--;

}

if(l<=r){

for(int i=bot;i>=top;i--){

System.out.print(arr[i][l] + " ");

}

l++;

}

}

}

public static void main(String ar[]){

int[][] arr={

{1,2,3,4},

{5,6,7,8},

{9,10,11,12},

{13,14,15,16}

};

spir(arr);

}

}

**Output:**

**Time Complexity:** O (n2)

**Space Complexity:** O (1)

**11. Palindrome Linked List**

**Code:**

**Output:**

**Time Complexity:** O (n)

**Space Complexity:** O (1)