22IT023 DHONI R

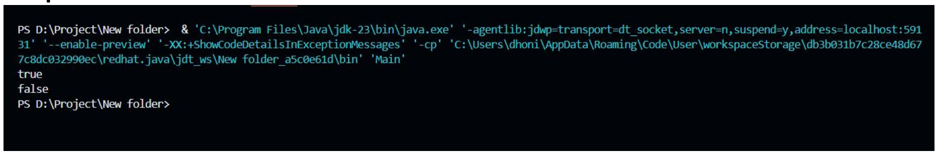
PRACTICE SET 3

1) Anagram problem

Code:

```
class Main {
public static boolean areAnagrams(String s1, String s2) {

// Your code here
int[] freq=new int[26];
for(char c:s1.toCharArray()){
freq[c-'a']++;
}
for(char c:s2.toCharArray()){
freq[c-'a']--;
}
for(int i=0;i<26;i++){
if(freq[i]!=0){
return false;
}
}
return true;
}
public static void main(String[] args) {
String s1 = "listen";
String s2 = "silent";
System.out.println(areAnagrams(s1, s2));

s1 = "hello";
s2 = "world";
System.out.println(areAnagrams(s1, s2));
}

}
```

Output:



```
PS D:\Project\New folder>  & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:591
31' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dhoni\AppData\Roaming\Code\User\workspaceStorage\db3b031b7c28ce48d67
7c8dc032990ec\redhat.java\jdt_ws\New folder_a5c0e61d\bin' 'Main'
true
false
PS D:\Project\New folder>
```

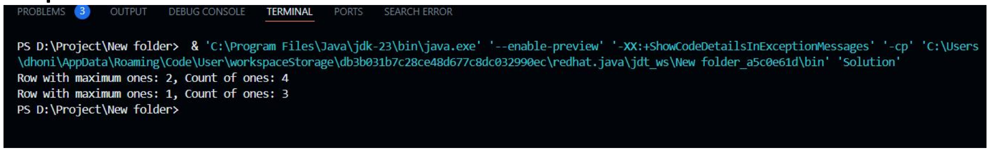
Time Complexity: O(N)

2) Row with maximum ones

Code:

```
class Solution {
public int[] rowAndMaximumOnes(int[][] mat) {
int m = mat.length;
int n = mat[0].length;
int max = 0;
int ind = 0;
for (int i = 0; i < m; i++) {
int crr = 0;
for (int j = 0; j < n; j++) {
if (mat[i][j] == 1) {
crr++;
}
}
if (crr > max) {
ind = i;
max = crr;
}
}
return new int[] { ind, max };
}
public static void main(String[] args) {
Solution solution = new Solution();

int[][] mat1 = {
{1, 0, 1, 1},

{0, 1, 1, 0},
{1, 1, 1, 1}
};
int[] result1 = solution.rowAndMaximumOnes(mat1);
System.out.println("Row with maximum ones: " + result1[0] + ", Count of ones: " + result1[1]);

int[][] mat2 = {
{0, 0, 0},
{1, 1, 1},
{0, 1, 0}
};
int[] result2 = solution.rowAndMaximumOnes(mat2);
System.out.println("Row with maximum ones: " + result2[0] + ", Count of ones: " + result2[1]);
}
}
```

Output:



```
PS D:\Project\New folder>  & 'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users
\dhoni\AppData\Roaming\Code\User\workspaceStorage\db3b031b7c28ce48d677c8dc032990ec\redhat.java\jdt_ws\New folder_a5c0e61d\bin' 'Solution'
Row with maximum ones: 2, Count of ones: 4
Row with maximum ones: 1, Count of ones: 3
PS D:\Project\New folder>
```

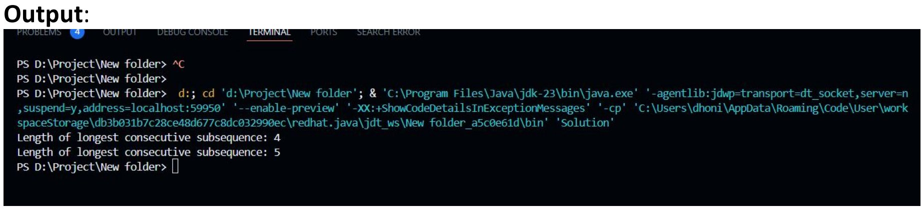
Time Complexity: O(N+M)

3) Longest consequtive subsequence

Code:

```
import java.util.*;
class Solution {
public int findLongestConseqSubseq(int[] arr) {
// code here
Arrays.sort(arr);
int crr=1;
int max=1;
int n=arr.length;
for(int i=1;i<n;i++){
if(arr[i]==arr[i-1]+1 ){

crr++;
}else if(arr[i]!=arr[i-1]){
crr=1;
}
max=Math.max(max,crr);
}

return max;
}
public static void main(String[] args) {
Solution solution = new Solution();

int[] arr1 = {1, 9, 3, 10, 4, 20, 2};
System.out.println("Length of longest consecutive subsequence: " +
solution.findLongestConseqSubseq(arr1));

int[] arr2 = {36, 41, 56, 35, 37, 34, 33, 42};
System.out.println("Length of longest consecutive subsequence: " +
solution.findLongestConseqSubseq(arr2));
}
}
```

Output:



```
PS D:\Project\New folder> ^C
PS D:\Project\New folder>
PS D:\Project\New folder>  d:; cd 'd:\Project\New folder'; & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n
,suspend=y,address=localhost:59950' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dhoni\AppData\Roaming\Code\User\work
spaceStorage\db3b031b7c28ce48d677c8dc032990ec\redhat.java\jdt_ws\New folder_a5c0e61d\bin' 'Solution'
Length of longest consecutive subsequence: 4
Length of longest consecutive subsequence: 5
PS D:\Project\New folder> 
```

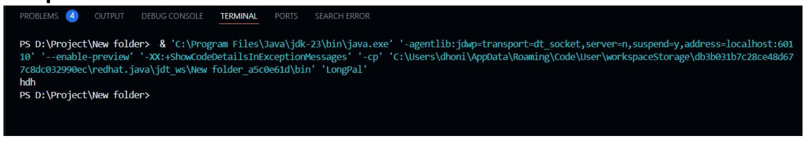
Time Complexity: O(nlogn)

4) Longest palindrome in a string

Code:

```
public class LongPal {

static boolean checkPal(String s, int low, int high) {
while (low < high) {
if (s.charAt(low) != s.charAt(high))
return false;
low++;
high--;
}
return true;
}

static String longestPalSubstr(String s) {

int n = s.length();
int maxLen = 1, start = 0;

for (int i = 0; i < n; i++) {
for (int j = i; j < n; j++) {
if (checkPal(s, i, j) && (j - i + 1) > maxLen) {
start = i;
maxLen = j - i + 1;
}
}
}

return s.substring(start, start + maxLen);
}

public static void main(String[] args) {
String s = "whdhaiueaknfsdnaijdjcjk";
System.out.println(longestPalSubstr(s));
}
}
```

Output:



```
PS D:\Project\New folder>  & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:601
10' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dhoni\AppData\Roaming\Code\User\workspaceStorage\db3b031b7c28ce48d67
7c8dc032990ec\redhat.java\jdt_ws\New folder_a5c0e61d\bin' 'LongPal'
hdh
PS D:\Project\New folder>
```

Time Complexity: O(n3)

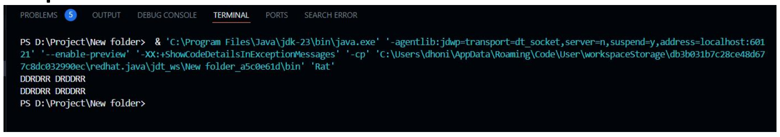
5) Rat in a maze problem

Code:

```
import java.util.ArrayList;
import java.util.List;

public class Rat{

static String direction = "DLRU";
static int[] dr = { 1, 0, 0, -1 };
static int[] dc = { 0, -1, 1, 0 };

static boolean isValid(int row, int col, int n, int[][] maze) {
return row >= 0 && col >= 0 && row < n && col < n && maze[row][col] == 1;
}

static void findPath(int row, int col, int[][] maze, int n, ArrayList<String> ans,
StringBuilder currentPath) {
if (row == n - 1 && col == n - 1) {
ans.add(currentPath.toString());
return;
}
maze[row][col] = 0;
for (int i = 0; i < 4; i++) {
int nextrow = row + dr[i];
int nextcol = col + dc[i];
if (isValid(nextrow, nextcol, n, maze)) {
currentPath.append(direction.charAt(i));
findPath(nextrow, nextcol, maze, n, ans, currentPath);
currentPath.deleteCharAt(currentPath.length() - 1);
}
}
maze[row][col] = 1;
}

public static void main(String[] args) {
int[][] maze = {
```

```
{ 1, 0, 0, 0 },
{ 1, 1, 0, 1 },
{ 1, 1, 0, 0 },
{ 0, 1, 1, 1 }
};

int n = maze.length;
ArrayList<String> result = new ArrayList<>();
StringBuilder currentPath = new StringBuilder();

if (maze[0][0] != 0 && maze[n - 1][n - 1] != 0) {
findPath(0, 0, maze, n, result, currentPath);
}

if (result.size() == 0)
System.out.println(-1);
else
for (String path : result)
System.out.print(path + " ");
System.out.println();

int[][] testMaze = {
{ 1, 0, 0, 0 },
{ 1, 1, 0, 1 },
{ 1, 1, 0, 0 },
{ 0, 1, 1, 1 }
};

int testN = testMaze.length;
ArrayList<String> testResult = new ArrayList<>();
StringBuilder testCurrentPath = new StringBuilder();

if (testMaze[0][0] != 0 && testMaze[testN - 1][testN - 1] != 0) {
findPath(0, 0, testMaze, testN, testResult, testCurrentPath);
}

if (testResult.size() == 0)
System.out.println(-1);
else
for (String path : testResult)

System.out.print(path + " ");
System.out.println();
}
}
```

Output:



```
PS D:\Project\New folder>  & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:601
21' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dhoni\AppData\Roaming\Code\User\workspaceStorage\db3b031b7c28ce48d67
7c8dc032990ec\redhat.java\jdt_ws\New folder_a5c0e61d\bin' 'Rat'
DDRDRR DRDDRR
DDRDRR DRDDRR
PS D:\Project\New folder>
```

Time Complexity: O(4^(n^2))