

**Name: Haroon Wajid**

**Roll Number: 21i-1763**

**Section: BS-CS-9A**

**Assignment#1**

**Question # 1:**

1. O(N+M)
2. O(n^3)
3. O(logn)
4. O(logn)

**Question # 2: Part (A)**

function searchInMatrix(matrix, N, M, K):

# Start from the top-right corner of the matrix

row = 0

col = M - 1

while row < N and col >= 0:

if matrix[row][col] == K:

return (row, col) # Return the position of K

elif matrix[row][col] > K:

col -= 1 # Move left

else:

row += 1 # Move down

return (-1, -1) # If K is not found in the matrix

**Part B: Time Complexity**

The time complexity of the above algorithm is O(N+M).

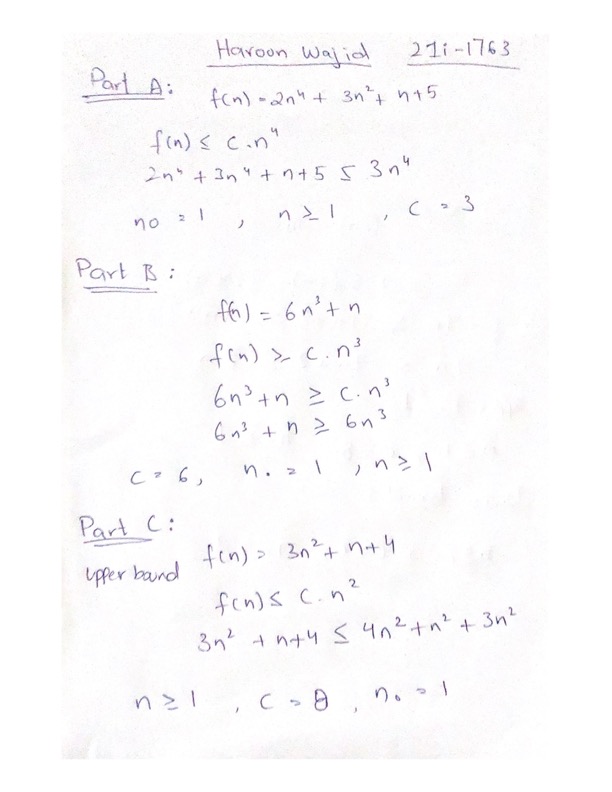
Explanation:

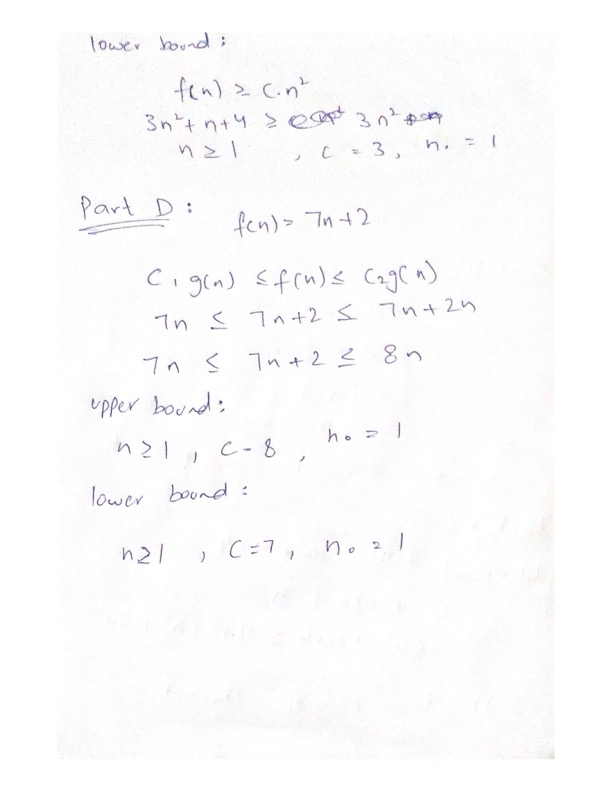
In the worst case, the algorithm processes each row and column exactly once.

Starting from the top-right corner, it can move left up to M times and down up to N times.

Therefore, the total number of steps is at most N+M.

**Question # 3:**

****

****

**Question # 4:**

A real-world example that requires sorting is the process of organizing a list of students by their grades in descending order to determine class rankings. This sorting task is crucial for various purposes, such as:

Awarding Scholarships: Many scholarships are awarded based on academic performance. Sorting students by their grades helps to quickly identify the top performers who are eligible for scholarships.

Class Valedictorian Selection: The student with the highest grades is often selected as the valedictorian. Sorting the list of students by grades ensures that the correct student is identified.

Example Scenario:

Suppose a school has a list of students with their respective grades in a final exam. The list looks like this:

Ali: 85

Usman: 92

Faisal: 78

Dawood: 95

Haroon: 88

To determine the class ranking, the school administration needs to sort this list in descending order of grades. After sorting, the list would look like this:

Dawood: 95

Usman: 92

Haroon: 88

Ali: 85

Faisal: 78

By sorting the students' grades, the school can efficiently perform tasks such as awarding the top student, determining eligibility for scholarships, and generating accurate class rankings.

**Question # 5:**

INSERTION-SORT-DESCENDING(A)

1 for j = 2 to A.length

2 key = A[j]

3 // Insert A[j] into the sorted sequence A[1..j-1]

4 i = j - 1

5 while i > 0 and A[i] < key

6 A [i + 1] = A[i]

7 i = i - 1

8 A [i + 1] = key