**Design and Analysis of Algorithm** A blue circle with white text

Description automatically generated

**SUBMISSION GUIDELINES:**

1. Assignment must be submitted within the due date.
2. Assignment must be on paper and in handwritten form.
3. Plagiarism is not allowed. In the case of pledge, you will score zero marks.
4. Do not submit your assignment after the deadline.
5. Do not copy from any other source.
6. It is an individual submission.
7. The deadline for this assignment is.
8. Question should be written with answer.

**Question no 1:**

The following are some pieces of code. Provide a Dry Run of all samples of 5 Passes. Also specify its worst-case time complexity.

Must Follow the Pattern in the attached File, Otherwise Marks will be Zero.

**A**

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

        for( j=0 ; j<n ; j++ )

{

             sum[i] += entry[i][j];

}

**B**

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

        for( j=0 ; j<i ; j++ )

            m += j;

**C**

i = 1;

while (i < n) {

  tot += i;

  i = i \* 2;

}

**D**

i = n;

while (i > 0) {

  tot += i;

  i = i / 2;

}

**E**

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

for( j=0 ; j<n ; j++ )

             sum[i] += entry[i][j][0];

**F**

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

   for( j=0 ; j<n ; j++ )

        for( k=0 ; k<n ; k++ )

            sum[i][j] += entry[i][j][k];

**G**

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

        for( k=0 ; k<n ; k++ )

             sum[i] += entry[i][0][k];

**H**

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

        for( j=0 ; j< sqrt(n) ; j++ )

            m += j;

**I**

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

{

     m += j;

     m += j;

     m += j;

     …

     m += j; // 31 times

}

**J**

int total(int n)

 for( i=0 ; i < n; i++)

  subtotal += i;

main()

 for ( i=0 ; i<n ; i++ )

  tot += total(i);

**K**

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

subtotal = 0;

for( j=0 ; j < i; j++)

  subtotal += j;

tot += subtotal;

}

**Question no 2:**

1. Determine the exact time complexity of this loop structure in terms of n.
2. Express the time complexity using Big O notation.
3. Simplify the time complexity expression.

**A**

int sum = 0;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

sum += 1;

}

}

**B**

int sum = 0;

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

for (int j = i; j <= 2\*i; ++j) {

sum += 1;

}

}

**C**

int sum = 0;

for (int i = 1; i <= n; i\*= 2) {

for (int j = 0; j < i; j++) {

sum += 1;

}

}

**Question no 3:**

1. Determine the exact time complexity of this loop structure in terms of n.
2. Express the time complexity using Big O notation.
3. Simplify the time complexity expression.

**A**

int sum = 0, i = 1;

while (i <= n) {

for (int j = 0; j < i; j++) {

sum += 1;

}

i \*= 2;

}

**B**

int sum = 0

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

for (int j = i; j < n; j++) {

for (int k = j; k < n; k += j) {

sum += 1;

}

}

}

**Question no 4:**

**A**

Imagine you’re a librarian, and you have a small number of books that are almost in order, but one book is out of place.

This is like the list **[9, 1, 2, 3, 4, 5, 6, 7, 8]**

**B**

Imagine you're working for a music streaming service, and you need to sort a user's recently played songs based on their last playback time. The list isn't very long (around 20-30 songs), but you want to quickly display the songs in chronological order with the most recently played song appearing first.

In the case **A** and **B** which Sorting Algorithm (Insertion sort, Selection sort, bubble sort) is best. Explain? And explain why other sorting algorithms are not.

**C**

Write time complexity of each Sorting Algorithms (All Cases).

**Question no 5:**

**A**

Apply Insertion Sort, Bubble Sort, Selection Sort on **Your Name** e.g. ( ahmad 🡪 aadhm )

**B**

Is these Sorting Algorithms hold properties which are given below :

1. In-place Sorting
2. Online Sorting
3. Stable Sorting