

# Assignment # 1

## (CS-2009 Design and Analysis of Algorithms – Spring-2024)

Due Date and Time: Tuesday, 13<sup>th</sup> February, 2024 (2:25 pm)

Marks: 20

Instructions:

- Late assignment will not be accepted.
- Only handwritten attempt will be graded, i.e., printed attempts will not be graded.
- Only the attempts submitted to Mr. Amir (or Mr. Aadil in case of Mr. Amir's unavailability) in the Academic office (till the due date & time) will be considered, i.e., the submissions that will be slided beneath instructors' office doors or submitted elsewhere will not be graded.
- There will be no credit if the given requirements are changed.
- Your solution will be evaluated in comparison with the best solution (having minimum number of steps).
- Please mention the question number and its part (if any) before writing down its solution. Use the same conventions used in the assignment's document.
- Plagiarism may result in zero marks in the whole assignments category (all assignments) regardless of the percentage plagiarized.
- Whenever a calculation is involved, your solution should show complete steps and a final answer. There will be significant marks for the correct final answer (as far as the assignments are concerned).
- In case of unavailability to submit the assignment in-person on the submission day, you must submit it either in-person (in the academic office) before the submission day or (as an emergency measure) email the scanned (using, e.g., CamScanner) copy of the handwritten attempt to your course instructor **before the deadline**.
- You must write your roll number, name, and section (Algo. Course section) on your submitted attempt.
- All algorithms must be written in the pseudocode form. For reference, please consult pages # 25, 30, 36, 39, 51, and 83 of the text book (4<sup>th</sup> Edition).

Q.1 For each of the following code snippets, calculate how many times the cout statement is executed. Also, provide the asymptotic time complexity function in terms of 'n'. Assume that there is no error in the codes. [12 Marks]

S. #	Code
1	<pre>for(int i=1;i&lt;=n;i++){     for(int j=2;j&lt;=n;j=j*j){         cout&lt;&lt;i&lt;&lt;j&lt;&lt;endl;     } }</pre>

2	<pre>for (i=n/2; i&lt;=n; i++)     for (j=1; j+n/2&lt;=n; j++)         for (k=1; k&lt;=n; k = k * 2){cout&lt;&lt;"hello ";c++; }</pre>
3	<pre>s=1; While(s&lt;=n) {     for(int i=1; i&lt;=s; s++)         cout&lt;&lt;" hello";     s*=2; }</pre>
4	<pre>for (j=1; j&lt;=n; j++)     for (k=1; k&lt;=j*3; k++)     {         cout&lt;&lt;"hello ";     }</pre>
5	<pre>for(int i=n/2;i&lt;=n;i++){     for(int j=1;j&lt;=n;j=j*j){         cout&lt;&lt;i&lt;&lt;j&lt;&lt;endl;     } }</pre>
6	<pre>for (j=1; j&lt;=n; j*=2)     for (k=n; k&gt;=1; k--) {for (i=1; i&lt;=n; i*=3)cout&lt;&lt;"hello ";     }</pre>
7	<pre>int m = (int)((15 + Math.round(3.2 / 2)) * (Math.floor(10 / 5.5) / 2.5) * Math.pow(2, 5)); for (int i = 0; i &lt; m; i++) {     cout&lt;&lt;"hello"; }</pre>
8	<pre>for (int i = 1; i &lt;= N * N; i *= 2) {for (int j = 0; j &lt; i; j++) {     cout&lt;&lt;"hello"; } }</pre>

**Q.2 Magic Square.** Analyze and find the asymptotic time complexity (in terms of N) of each segment of the following C++ code and then find the overall asymptotic time complexity of the code using the sum rule of complexity analysis, where N is the input size (Positive and Odd): Magic Square is a Square Matrix in which the sum of each row, column and diagonal is same. [8 Marks]

```
void MagicSquare(int N)
{ const int MaxSize = 51;
  //Max Square Size.
```

```

int square[MaxSize][MaxSize], k, l;

for(int i=0; i<N; i++) //Initialize the Square in a Loop.
    for(int j=0; j<N; j++)
        square[i][j] = 0;

square[0][(N-1)/2] = 1;      //Middle of First Row.
int key = 2;
i = 0; //int i and j are the current positions.
int j = (N-1)/2;

while(key <= N*N){ //Move Up and Left in a Circular Fashion.
    if(i <= 0)
        k = N-1;
    else
        k = i-1;
    if(j <= 0)
        l = N-1;
    else
        l = j-1;      //Moved Up and Left in a Circular Fashion.
    if(square[k][l])
        i = (i+1)%N;
    else{
        i = k;
        j = l;
    }
    square[i][j] = key; //Allocating the location with Key
    key++;
}

//Display the Magic Square by Accessing the Array.
cout<<"\n\n\nMagic Square is of Size: "<<N<<endl<<endl;

for(i=0; i<N; i++){
    for(j=0; j<N; j++)
        cout<<square[i][j]<<"\t";
    cout<<endl<<endl;
}

int Sum = 0;
for(i=0; i<N; i++){
    Sum += square[0][i];
}

cout<<"The Sum of each Row, Column and Diagonal is: "<<Sum<<endl;
}

```