

**Metropolitan University, Sylhet**  
Department of Computer Science and Engineering  
Spring Term Examination Spring – 2021  
**Program: B.Sc. in CSE    Batch: 49<sup>th</sup>, 50<sup>th</sup>(A+B)**  
**Course: CSE-231:: Algorithm**

---

- ★ Submit a **PDF** file for the assignment containing your answer.
    - Answers can be either handwritten or typed.
  - ★ **Assignment File Name: 201-115-ZZZ\_Algo\_Assign\_Mid**
    - Replace **ZZZ** with your roll.
    - If you are a retake student, replace the first part of the assignment file corresponding to your roll number.
  - ★ If you have any queries, comment, or inform your CR.
- 

1. The following code adds two matrices and prints the first row of the sum matrix. Analyze the following code. 10

1. Find **T(n)** for the code.
2. Then analyze it with Big Theta.
  - a. Find the corresponding **C<sub>1</sub>**, **C<sub>2</sub>**, and **n<sub>0</sub>** and validate your logic for that.

```
1 void addMatrices(int a[][n], int b[][n], int sum[][n])
2 {
3
4     for (int i = 0; i < n; ++i) {
5         for (int j = 0; j < n; ++j) {
6             sum[i][j] = a[i][j] + b[i][j];
7         }
8     }
9
10    for (int j = 0; j < n; ++j) {
11        cout << sum[0][j] << " ";
12    }
13    cout << endl;
14
15    return;
16 }
17
```

2. Given the graph below,

10

1. Run BFS in the graph from the source nodes.
2. Show the explored nodes at each level.
3. Also, write how many edges are required to reach each node from the source.

Edges of the graphs are:

(a, b), (a, c), (f, i), (b, d), (e, f), (i, e), (i, j)

**Source node : a**

The values will of { a, b, c, d, e, f, i, j } will be found in here:

<https://ideone.com/TJ0Unn>

Enter your roll in there in the stdin and it will generate the output for you.

- 3a Sheldon Cooper is a brilliant Physicist, String theory scientist, and also a huge nerd who have recently learned how to program. He has recently learned about stack and the queue. So he made a weird algorithm which he calls “Sheldorithm”. Given a set of numbers in an array or vector as an input called **vecInput** (total number of elements will always be even), the algorithm work as following. Write the output (where you were told to print) as mentioned in the code accordingly. **sz = 10** in the code. 10



```
1 int sz = vecInput.size();
2
3 queue <int> que;
4 stack <int> stk;
5
6
7 for i=0; i<sz; i++
8 {
9     if i%2 == 1
10    {
11        stk.push( vecInput[i] )
12    }
13    else {
14        que.push( vecInput[i] )
15    }
16 }
17
18 print the stk;
19 print the que;
20
21 vector <int> vecOutPut;
22
23 curIdx = 0;
24
25 while curIdx < sz
26 {
27     if stk is not empty
28     {
29         vecOutPut[ curIdx ] = stk.top();
30         stk.pop();
31
32         curIdx++;
33     }
34
35     if que is not empty
36     {
37         vecOutPut[ curIdx ] = que.front();
38         que.pop();
39
40         curIdx++;
41     }
42 }
43
44 print the vecOutPut
45
46
```

Just change the value of the **roll** in the code (Python) and “**ideone it**”. Or you also can uncomment the input line and then give the last 3 digits of your roll as a input.  
Link to generate values for your roll: <https://ideone.com/WLgxHs>

3b Also, show analysis of your algorithm and express it in order of O (big Oh).