**Data Structures Lab (CSC204)** **Session: Monsoon 2023-24**
**III BTech (CSE)** **Date: 16.10.2023**

**Assignment-8 (Tree-III and Graph)**

---

1. B-tree is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree generalizes the binary search tree, allowing for nodes with more than two children. Unlike other self-balancing binary search trees, the B-tree is well suited for storage systems that read and write relatively large blocks of data, such as disks. It is commonly used in databases and file systems. A B-tree of order $m$ is a tree which satisfies the following properties:

- Every node has at most $m$ children.
- The root has at least two children and at most $m$ children.
- All internal nodes other than the root node have at least $\left\lceil \frac{m}{2} \right\rceil$child nodes and at most $m$ child nodes.
- The number of keys in each internal node is one less than the number of child nodes and these keys partitions the keys in the sub trees of the node in a manner similar to that of $m$-way search tree.
- All leaf nodes are on the same level. It means each path from the root to a leaf has the same length.

a) Write a program to create a B-tree of order $m$ by inserting the keys given by the user at runtime. Create the tree in right biased manner (i.e., during splitting if the nodes have even number of elements, then the upper median will be considered as the middle element). Then print the key values which are present in the root node of the created B-tree.

**Sample Input:**
Enter the order ($m$) of the B-tree: 4
Enter the number of keys to be inserted: 9
Enter the keys: 10 20 40 50 60 70 80 30 35

**Sample Output:**
Keys present in root nodes of the created B-tree are: 30 40 70

b) Write a function $\boldsymbol{key\_search}(\,)$ which will search for the presence of a user given key-value $k$ in the created B-tree. If $k$ is present in the tree then your function should print all the keys located in the same nodes in which $k$ is present.

**Sample Input:**
Enter the key to be searched: 20

**Sample Output:**
Key value 20 is present.
The node contains the keys: 10 20

2. An undirected graph $G$ is represented as $G = (V, E)$, where $V$ is the set of vertices and $E$ is the set of edges. Your graph can be represented using either an adjacency list or an adjacency matrix. Suppose $|V| = m$ and $|E| = n$, then perform the following tasks as described below.

   a) Write a program to generate the graph based on the user inputs, then display the graph using adjacency list and adjacency matrix separately.

      **Sample Input:**
      Enter the value of $m$: 5
      Enter the value of $n$: 5
      Enter the $m$ pair of vertices representing the edges: $AB, AC, AD, CD, CE$

      **Sample Output:**
      Adjacency List is:
      $A \rightarrow B, C, D$
      $B \rightarrow A$
      $C \rightarrow A, D, E$
      $D \rightarrow A, C$
      $E \rightarrow C$

      Adjacency Matrix is:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 1 | 1 | 1 | 0 |
| B | 1 | 0 | 0 | 0 | 0 |
| C | 1 | 0 | 0 | 1 | 1 |
| D | 1 | 0 | 1 | 0 | 0 |
| E | 0 | 0 | 1 | 0 | 0 |

   b) The *eccentricity* of a vertex is defined as the maximum of the shortest distances from that vertex to all other vertices in the graph. The *central point* of the graph is the vertex $V$ with minimum eccentricity and the set of all central points is called the *centre of the graph*. Write a function $\boldsymbol{find\_eccentricity}()$ to find the *eccentricity* of all the vertices as well as the *centre of the given graph*.

      **Sample Output:**
      The eccentricities of the vertices are:
      $A = 2$
      $B = 3$
      $C = 2$
      $D = 2$
      $E = 3$

      The centre of the graph is: $A, C, D$