

DAY-13.

Graphs.

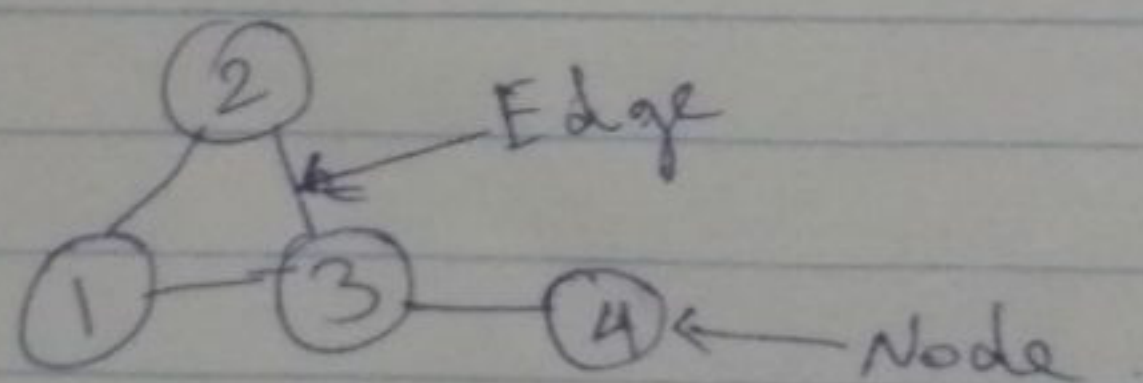
→ a non-linear data structure.

→ a pictorial representation

of a set of objects where some pairs of objects are connected by links.

→ Nodes are called here as "vertices", and links as "edges".

→ Implemented by "Dictionaries" and "Lists".



{
1 : [2, 3],
2 : [1, 3],
3 : [1, 2, 4],
4 : [3]
}

Implementation:-

A simple program:-

```
class Graph:
```

```
    graph_dict = {}
```

```
    def addEdge(self, node, neighbor):
```

```
        if node not in self.graph_dict:
```

```
            self.graph_dict[node] = [neighbor]
```

```
        else:
```

```
            self.graph_dict[node].append(neighbor)
```

```
    def show_edges(self):
```

```
        for node in self.graph_dict:
```

```
            for neighbor in self.graph_dict[node]:
```

```
                print("(" + node + ", " + neighbor + ")")
```

Application of graph:-

- to define the flow of computation.
- to represent networks of communication.
- to represent data organization.

→ can be used to model many types of relations and process dynamics.

* graph plays major role in social media such as LinkedIn and Facebook.