Task-5

Adjacency Matroix: Assume that, there are 'V' vertex in our graph. So, to implement this in our graph, we have to create a 'VxV' matrix. Then, if we traverse the matrix to implement BFS and DFS then it will take time 'vxv'. If we compare this with 'n' then it will be n^2. So, time complexity for both BFS and DFS will be $O(n^2)$ if we use adjacency matrix.

Adjacency list: Suppose there is a graph where 'V' vertexes and 'E' edges. To implement this in ~~adjacenc~~ adjacency list at first we have to create a 'V' size array. Then in every index we store all the values of v. To connect the nodes, which are connecte with one vertexe we have to create nodes and connect them. So, the sum of all size of the adjacency list will be 'E'. So, if we want to traverse this by BFS / DFS. It will take $O(V+E)$ time. $

So, in adjacency matrix the time complexity will be $O(V^2)$ and for adjacency list it will be $O(V+E)$. So, it is always better to use adjacency list instead of adjacency matrix.

In the given scenario, the output of

BFS: 1 2 3 4 5 7 11 6 12

And for DFS: 1 2 3 4 5 6 7 8 9 10 11 12.

It is clearly visible that by using DFS it takes the longest path. And By using BFS we find the shortest path.