```python
import numpy as np
class Graph:
    def __init__(self,vertices):
        self._adjMat = np.zeros((vertices, vertices))
        self._vertices = vertices
        self._visited = [0] * vertices

    def insert_edge(self,u,v,w=1):
        self._adjMat[u][v] = w

    def delete_edge(self,u,v):
        self._adjMat[u][v] = 0

    def get_edge(self,u,v):
        return self._adjMat[u][v]

    def vertices_count(self):
        return self._vertices

    def edge_count(self):
        count = 0
        for i in range(self._vertices):
            for j in range(self._vertices):
                if not self._adjMat[i][j] == 0:
                    count += 1
        return count

    def indegree(self,u):
        count = 0;
        for i in range(self._vertices):
            if not self._adjMat[i][u] == 0:
                count += 1
        return count

    def outdegree(self,u):
        count = 0;
        for i in range(self._vertices):
            if not self._adjMat[u][i] == 0:
                count += 1
        return count

    def display(self):
        print(self._adjMat)

    def DFS(self, source):
        if self._visited[source] == 0:
            print(source, end=' - ')
```

```
48              self._visited[source] = 1
49              for j in range(self._vertices):
50                  if self._adjMat[source][j] == 1 and self.
   _visited[j] == 0:
51                      self.DFS(j)
52
53 G = Graph(7)
54 G.insert_edge(0,1)
55 G.insert_edge(0,5)
56 G.insert_edge(0,6)
57 G.insert_edge(1,0)
58 G.insert_edge(1,2)
59 G.insert_edge(1,5)
60 G.insert_edge(1,6)
61 G.insert_edge(2,3)
62 G.insert_edge(2,4)
63 G.insert_edge(2,6)
64 G.insert_edge(3,4)
65 G.insert_edge(4,2)
66 G.insert_edge(4,5)
67 G.insert_edge(5,2)
68 G.insert_edge(5,3)
69 G.insert_edge(6,3)
70 G.DFS(0)
71
72
73
```