

Para el G una lista de adyacencia

Implementación BFS

```
visitado = []
cola = []

def BFS(G,s):
    visitado.append(s)
    cola.append(s)

    while cola != []:
        u = cola.pop(0)

        for v in G[u]:
            if v not in visitado:
                visitado.append(v)
                cola.append(v)
```

Implementación DFS

Recursivo básico

```
visitado = []

def dfs(G,s):
    visitado.append(s)

    for v in G[s]:
        if v not in visitado:
            dfs(G,v)
```

Iterativo básico

```
from queue import LifoQueue

visitado = []
pila = LifoQueue()

def dfs(G,s):
    pila.put(s)

    while not pila.empty():
        u = pila.get()

        if u not in visitado:
            visitado.append(u)

            for v in G[u]:
                if v not in visitado:
                    pila.put(v)
```

Topological Sort

#habría que modificar DFS para que devuelva el postorder, alcanza con un postorder.append(v) al final en el recursivo

```
visitado = []
```

```
postorder = []

def toposort(G)
    for u in G:
        if u not in visitado:
            dfs(G,u, postorder)
    return postorder[::-1] # dada vuelta
```