



# Tecnológico de Monterrey

---

Casos de prueba

---

## **Programming of data structures**

Inés Alejandro Garcia Mosqueda A00834571

Profesores: Luis Ricardo Peña Llamas  
Jorge Gonzalez

09/11/2022

## Casos de prueba

Se proponen 4 casos de prueba diferentes con el fin de demostrar las distintas funciones implementadas

### Caso de prueba 1

Lectura del archivo Prueba.txt

```
Graph2 > Prueba.txt
1 0 1
2 1 2
3 2 3
4 5 4
5 4 6
```

.....Matriz Adyacencia.....	.....Prueba BFS.....	.....Prueba DFS.....
0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0	BFS start vertex (0) : 0 1 2 3  BFS start vertex (1) : 1 2 3  BFS start vertex (2) : 2 3  BFS start vertex (3) : 3  BFS start vertex (4) : 4 6  BFS start vertex (5) : 5 4 6  Es bipartita? 1	DFS start vertex (0) : 0 1 2 3  DFS start vertex (1) : 1 2 3  DFS start vertex (2) : 2 3  DFS start vertex (3) : 3  DFS start vertex (4) : 4 6  DFS start vertex (5) : 5 4 6

Obteniendo como el resultado de la búsqueda topologica, bipartita y detección de árboles, el siguiente resultado

```
Es bipartita? 1
topological sort: 3 2 1 0 6 4 5
Es arbol: 0
```

Siendo estos los esperados

## Caso de prueba 2

Lectura del archivo Prueba2.txt

```
Graph2 > Prueba2.txt
1 2
2 3 6
3 10 6
4 2 3
5 4 5
6 0 5
7 9 6
8 6 7
9 3 2
10 8 7
11 9 1
12 1 4
13 1 9
14 1 8
15 3 4
16 6 3
17 7 2
18 3 7
19 10 8
20 5 10
21 3 10
22 7 1
```

```
.....Matriz Adyacencia.....
0 0 0 0 0 1 0 0 0 0 0
0 0 1 0 1 0 0 0 1 1 0
0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 0 1
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1
0 0 0 1 0 0 0 1 0 0 0
0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0
0 1 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0

Vertex 0: -> 5
Vertex 1: -> 2 -> 4 -> 9 -> 8
Vertex 2: -> 3
Vertex 3: -> 6 -> 2 -> 4 -> 7 -> 10
Vertex 4: -> 5
Vertex 5: -> 10
Vertex 6: -> 7 -> 3
Vertex 7: -> 2 -> 1
Vertex 8: -> 7
Vertex 9: -> 6 -> 1
Vertex 10: -> 6 -> 8
```

```
.....Prueba BFS.....

BFS start vertex (0) :
0 5 10 6 8 7 3 2 1 4 9

BFS start vertex (1) :
1 2 4 9 8 3 5 6 7 10

BFS start vertex (2) :
2 3 6 4 7 10 5 1 8 9

BFS start vertex (3) :
3 6 2 4 7 10 5 1 8 9

BFS start vertex (4) :
4 5 10 6 8 7 3 2 1 9

BFS start vertex (5) :
5 10 6 8 7 3 2 1 4 9
```

```
.....Prueba DFS.....

DFS start vertex (0) :
0 5 10 6 7 2 3 4 1 9 8

DFS start vertex (1) :
1 2 3 6 7 4 5 10 8 9

DFS start vertex (2) :
2 3 6 7 1 4 5 10 8 9

DFS start vertex (3) :
3 6 7 2 1 4 5 10 8 9

DFS start vertex (4) :
4 5 10 6 7 2 3 1 9 8

DFS start vertex (5) :
5 10 6 7 2 3 4 1 9 8
```

Obteniendo como el resultado de la búsqueda topológica, bipartita y detección de árboles, el siguiente resultado

```
Es bipartita? 0
topological sort: 4 3 2 9 8 1 7 6 10 5 0
Es arbol: 0
```

## Caso de prueba 2

Lectura del archivo Prueba3.txt

```
Graph2 > Prueba3.txt
1 0 3
2 3 7
3 3 6
4 7 8
5 6 8
6 8 11
7 2 0
8 2 1
9 1 3
10 4 0
11 4 5
12 5 10
13 5 9
14 10 9
15 9 12
```

```
.....Matriz Adyacencia.....
0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0

Vertex 0: -> 3
Vertex 1: -> 3
Vertex 2: -> 0 -> 1
Vertex 3: -> 7 -> 6
Vertex 4: -> 0 -> 5
Vertex 5: -> 10 -> 9
Vertex 6: -> 8
Vertex 7: -> 8
Vertex 8: -> 11
Vertex 9: -> 12
Vertex 10: -> 9
Vertex 11:
Vertex 12:
```

```
.....Prueba BFS.....
BFS start vertex (0) :
0 3 7 6 8 11

BFS start vertex (1) :
1 3 7 6 8 11

BFS start vertex (2) :
2 0 1 3 7 6 8 11

BFS start vertex (3) :
3 7 6 8 11

BFS start vertex (4) :
4 0 5 3 10 9 7 6 12 8 11

BFS start vertex (5) :
5 10 9 12
```

```
.....Prueba DFS.....
DFS start vertex (0) :
0 3 7 8 11 6

DFS start vertex (1) :
1 3 7 8 11 6

DFS start vertex (2) :
2 0 3 7 8 11 6 1

DFS start vertex (3) :
3 7 8 11 6

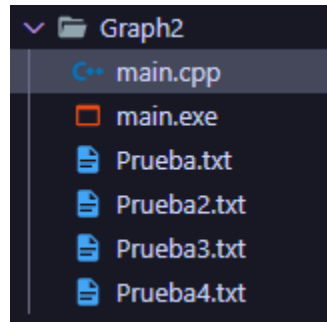
DFS start vertex (4) :
4 0 3 7 8 11 6 5 10 9 12

DFS start vertex (5) :
5 10 9 12
```

Obteniendo como el resultado de la búsqueda topológica, bipartita y detección de árboles, el siguiente resultado

```
Es bipartita? 1
topological sort: 11 8 7 6 3 0 1 2 12 9 10 5 4
Es arbol: 0
```

Adicionalmente se agregaron dos documentos txt de prueba



```
int main(){  
    //Caso de prueba  
    string casos [4];  
    casos[0] = "Prueba.txt";  
    casos[1] = "Prueba2.txt";  
    casos[2] = "Prueba3.txt";  
    casos[3] = "Prueba4.txt";  
  
    Graph G;  
    G.loadGraph(casos[2]);  
    cout << "\n.....Matriz Adyacencia....." <<endl;  
    G.printMatrix();  
    G.printGraph();  
    cout << "\n.....Prueba BFS....." <<endl;  
    G.BFS(0);  
    G.BFS(1);  
    G.BFS(2);  
    G.BFS(3);  
    G.BFS(4);  
    G.BFS(5);  
    cout << "\n.....Prueba DFS....." <<endl;  
    G.DFS(0);  
    G.DFS(1);  
    G.DFS(2);  
    G.DFS(3);  
    G.DFS(4);  
    G.DFS(5);  
    cout<< "\nEs bipartita? " << G.isBipartite(0)<<endl;  
    cout<<"topological sort: ";  
    G.topologicalSort();  
    cout<<"Es arbol: " <<G.isTree();  
    return 0;  
}
```