**Ejercicio 1. Algoritmos de grafos dirigidos**

Dado el siguiente grafo dirigido:

1. Aplique el algoritmo DFS, para encontrar todos los vértices conectados con el vértice **3**.
2. Aplique el algoritmo BFS, para encontrar todos los vértices conectados con el vértice **3**.

0

1

2

3

4

5



2

1

2

0

5

4

2

1

2

4

3

0

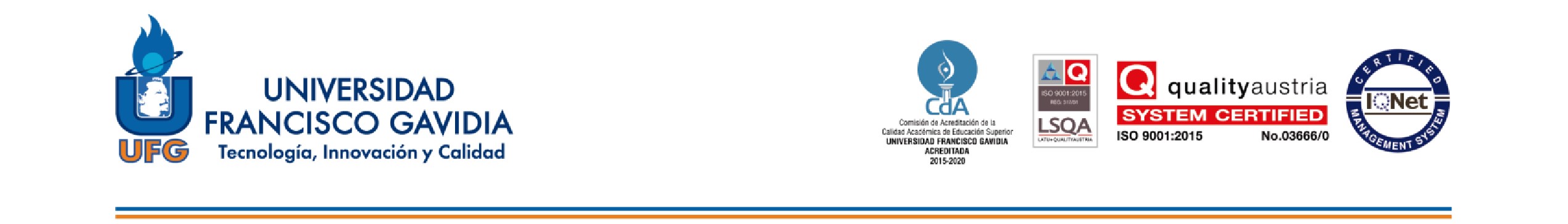
1

2

3

4

5



1. Aplique el algoritmo DFS, para encontrar todos los vértices conectados con el vértice **3**.

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | T | 1 |
| 1 | T | 3 |
| 2 | T | 0 |
| 3 |  | - |
| 4 |  |  |
| 5 |  |  |

0

1

2

4

3

5

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | T | 1 |
| 1 | T | 3 |
| 2 | T | 0 |
| 3 |  | - |
| 4 | T | 3 |
| 5 |  |  |

0

1

2

4

3

5

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | T | 1 |
| 1 | T | 3 |
| 2 | T | 0 |
| 3 |  | - |
| 4 | T | 3 |
| 5 | T | 3 |

0

1

2

4

3

5

Dando respuesta a la pregunta de ¿qué vértices están conectados con el vértice 3?

* Son: en Pos-orden 2, 0, 1, 4, 5, 3

0

1

2

4

3

5

Aplique algoritmo BFS, para encontrar todos los vértices conectados con el vértice 3

0

0

1

2

4

3

5

|  |  |  |
| --- | --- | --- |
| V | EdgeTo[] | distTo[] |
| 0 |  |  |
| 1 | 3 | 1 |
| 2 | 3 | 1 |
| 3 | - | 0 |
| 4 | 3 | 1 |
| 5 | 3 | 1 |

|  |  |  |
| --- | --- | --- |
| V | EdgeTo[] | distTo[] |
| 0 | 1 | 2 |
| 1 | 3 | 1 |
| 2 | 3 | 1 |
| 3 | - | 0 |
| 4 | 3 | 1 |
| 5 | 3 | 1 |

0

1

2

4

3

5

**Según el algoritmo de búsqueda en anchura los vértices conectados con el 3 son:**

* **En orden: 3, 1, 2, 4, 5, 0**

0

1

2

4

3

5

# Ejercicio 2. Orden topológico

Encuentre el orden topológico de los vértices en el grafo siguiente, iniciando con el vértice **3**.

0

0

1

2

4

3

5

2

1



|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 0 |  | |
| 5 | 4 | 2 | 1 |
| 2 |  |  |  |
| 4 |  |  |  |

2

3



4

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | T | 1 |
| 1 | T | 3 |
| 2 | T | 0 |
| 3 | T | - |
| 4 |  |  |
| 5 |  |  |

0

1

2

4

3

5

Orden: 3, 1, 0, 2

Pos Orden: 2, 0, 1

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | T | 1 |
| 1 | T | 3 |
| 2 | T | 0 |
| 3 | T | - |
| 4 | T | 3 |
| 5 | T | 3 |

0

1

2

4

3

5

Orden: 3, 1, 0, 2, 4, 5

Pos Orden: 2, 0, 1, 4, 5, 3

**Según el orden topológico el grafo anterior tiene el siguiente ordenamiento topológico:**

**3, 5, 4, 1, 0, 2**

3

5

4

1

2

0

**Ejercicio 3. Árboles de expansión mínima**

Dado el siguiente grafo no dirigido

V 7 a) Dibuje su grafo asociado.

13 A  b) Encuentre el orden en que se agregan los vértices al árbol de expansión 0-1 7 mínima usando el algoritmo de Kruskal.

0-2 5 c) Encuentre el orden en que se agregan los vértices al árbol de expansión 0-6 12 mínima usando el algoritmo de Prim.

0-4 8

1-2 11

1-3 4

2-3 9

2-6 17

3-4 3

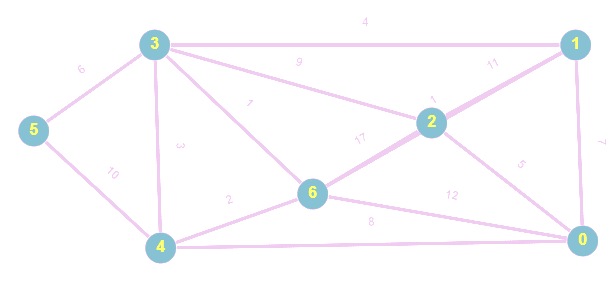
3-5 6

3-6 1

4-5 10

4-6 2

1. Dibuje su grafo asociado



1. Encuentre el orden en el que se agregan los vértices al árbol de expansión mínima usando el algoritmo de kruskal

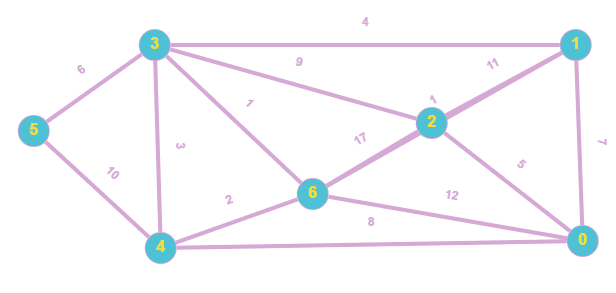
|  |  |  |
| --- | --- | --- |
| 7 |  |  |
| 13 | A |
| 3-6 | 1 |
| 4-6 | 2 |
| 3-4 | 3 |
| 1-3 | 4 |
| 0-2 | 5 |
| 3-5 | 6 |
| 0-1 | 7 |
| 0-4 | 8 |
| 2-3 | 9 |
| 4-5 | 10 |
| 1-2 | 11 |
| 0-6 | 12 |
| 2-6 | 17 |

**Por el algoritmo de kruskal los vértices se agregan en el siguiente orden:**

**3 – 6, 4 – 6, 1 – 3, 0 – 2, 3 – 5, 0 – 1**

**3, 6, 4, 1, 0, 2, 5**

1. Encuentre el orden en que se agregan los vértices al árbol de expansión minima usando el algoritmo de Prim



**Por el algoritmo de kruskal los vértices se agregan en el siguiente orden:**

**3 – 6, 4 – 6, 1 – 3, 3 -5, 0 – 1, 0 - 2**

**3, 6, 4, 1, 5, 0, 2**

**Ejercicio 4. Algoritmo de la ruta más corta**

Encuentre la ruta más corta desde el vértice 3 hacia cualquier otro vértice en el siguiente grafo.

0

1

4

2

3

5

7

1

5

8

3

4

10

17

4

11

|  |  |  |
| --- | --- | --- |
| V | distTo[] | EdgeTo[] |
| 0 | 10.0 | 3 🡪 0 |
| 1 |  |  |
| 2 | 4.0 | 3 🡪 2 |
| 🡪 3 | 0.0 | - |
| 4 | 11.0 | 3 🡪 4 |
| 5 | 17.0 | 3 🡪 5 |

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | 7.0 | 2 🡪 0 |
| 1 | 9.0 | 2 🡪1 |
| 🡪 2 | 4.0 | 3 🡪 2 |
| 3 | 0.0 | - |
| 4 | 11.0 | 3 🡪 4 |
| 5 | 17.0 | 3 🡪 5 |

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 🡪 0 | 7.0 | 2 🡪 0 |
| 1 | 9.0 | 2 🡪1 |
| 2 | 4.0 | 3 🡪 2 |
| 3 | 0.0 | - |
| 4 | 11.0 | 3 🡪 4 |
| 5 | 11.0 | 0 🡪 5 |

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | 7.0 | 2 🡪 0 |
| 🡪 1 | 9.0 | 2 🡪1 |
| 2 | 4.0 | 3 🡪 2 |
| 3 | 0.0 | - |
| 4 | 10.0 | 1 🡪 4 |
| 5 | 11.0 | 0 🡪 5 |

* **La ruta más corta que se puede tomar es:**

|  |  |  |
| --- | --- | --- |
| V | Marked[] | EdgeTo[] |
| 0 | 7.0 | 2 🡪 0 |
| 🡪 1 | 9.0 | 2 🡪1 |
| 2 | 4.0 | 3 🡪 2 |
| 3 | 0.0 | - |
| 4 | 10.0 | 1 🡪 4 |
| 5 | 11.0 | 0 🡪 5 |

1

0

4

2

5

3