Facultad de Ingeniería
Ingeniería de Sistemas y Computación

Estructura de Datos
Ejercicios de grafos

Luis Bertel
Jairo Vélez
Jorge Meza

1 Grafos no conexo

Dado el grafo de la figura 1, elaborar un programa que indique si el grafo es conexo o no conexo. De no ser conexo indique el/los vértice(s). Desarrolle los programas utilizando lista y matriz de adyacencia.

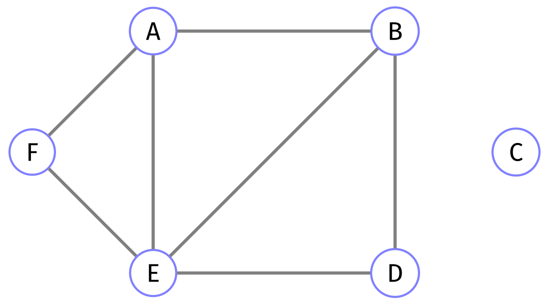

Figura 1: Grafo no conexo

2 Grafos no dirigidos y ponderados

Con el grafo de la figura 2, elabore los siguientes programas (elaborar los programas utilizando lista y matriz de adyacencia):

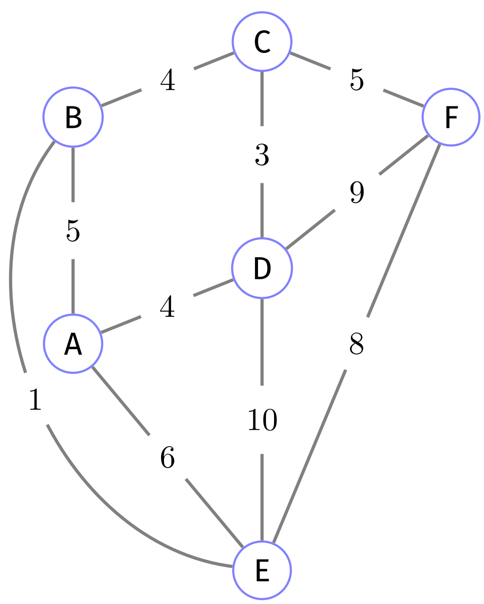

Figura 2: Grafo no dirigido y ponderado

1. Calcule el grado de cada vértice. Recuerde que para un grafo no dirigido el grado de un vértice es el número de aristas que inciden en él.
2. Solicitando un vértice de inicio se pide hallar el recorrido por anchura a partir del vértice suministrado en orden ascendente.
3. Solicitando un vértice de inicio se pide hallar el recorrido por profundidad a partir del vértice suministrado en orden ascendente.
4. Indicar si existe un camino simple en el grafo. Un camino simple es aquel que partiendo un vértice se puede recorrer todo el grafo sin repetir arista o vértice.
5. Indicar si el grafo es fuertemente conectado. Un grafo es fuertemente conectado cuando si desde cualquier vértice se puede llegar a todos los demás.
6. Indicar si el grafo es Euleriano. Un grafo es Euleriano cuando se parte de un vértice y se recorre todas las aristas sin repetirlas y llegando al vértice de partida (se puede repetir vértices).
7. Indicar si es Hamiltoniano. Un grafo es Hamiltoniano si se parte de un vértice, se puede recorrer todos los vértices sin repetir ninguno y llegando al vértice de origen (se puede repetir aristas).
8. Indicar si el grafo es completo. Un grafo es completo cuando existe para todos los n vértices, existen $n - 1$ aristas que conecten a los $n - 1$ vértices.
9. Dado dos vértices, indicar los n posibles caminos que existen entre dichos vértices.
10. Dado dos vértices, indicar el/los vértice(s) que más se utilizan en el/los camino(s) hallado(s).
11. Hallar el árbol de expansión mínima utilizando PRIM.
12. Hallar el árbol de expansión mínima utilizando KRUSKAL.
13. Dado un vértice y utilizando el recorrido por profundidad, indique el costo de cada una de las posibles rutas del vértice dado a los demás vértices.
14. Dado un vértice, calcule el costo mínimo del vértice dado a los demás vértices utilizando el algoritmo de Dijkstra.

3 Digrafo ponderado

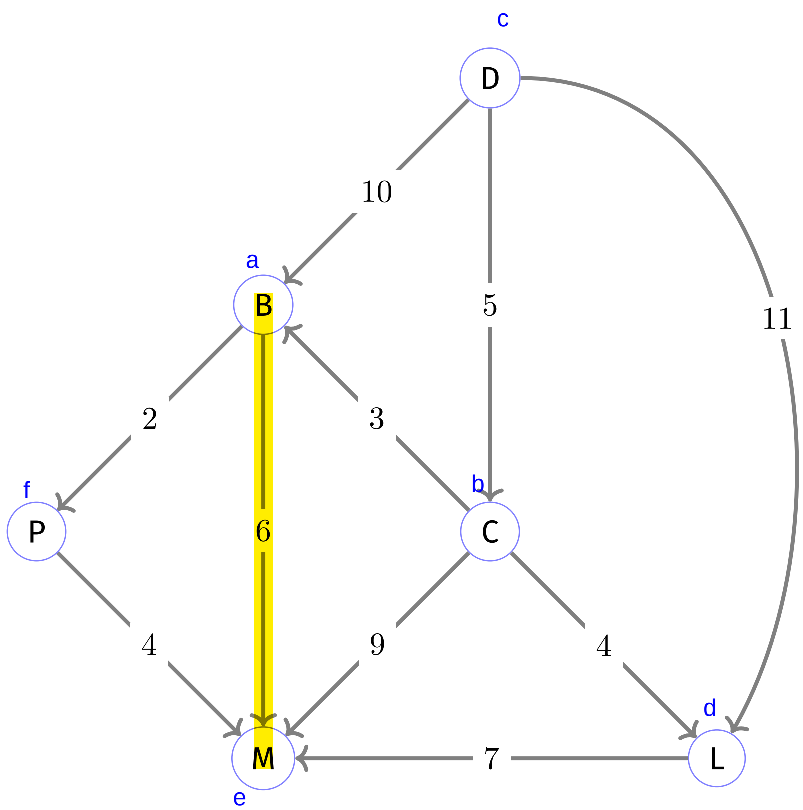

Figura 3: Digrafo ponderado

1. Calcule el grado de cada vértice. Recuerde que para un digrafo se tiene el grado entrada de un vértice es el número de aristas que llegan en él, y el grado de salida que indica el número de aristas que salen de él.
2. Solicitando un vértice de inicio se pide hallar el recorrido por anchura a partir del vértice suministrado en orden ascendente.
3. Solicitando un vértice de inicio se pide hallar el recorrido por profundidad a partir del vértice suministrado en orden ascendente.
4. Indicar si existe un camino simple en el grafo. Un camino simple es aquel que partiendo un vértice se puede recorrer todo el grafo sin repetir arista o vértice.
5. Indicar si el grafo es fuertemente conectado. Un grafo es fuertemente conectado cuando si desde cualquier vértice se puede llegar a todos los demás.
6. Indicar si el grafo es completo. Un grafo es completo cuando existe para todos los n vértices, existen $n-1$ aristas que conecten a los $n-1$ vértices.

7. Dado dos vértices, indicar los n posibles caminos que existen entre dichos vértices.
8. Dado dos vértices, indicar el/los vértice(s) que más se utilizan en el/los camino(s) hallado(s).
9. Dado un vértice y utilizando el recorrido por profundidad, indique el costo de cada una de las posibles rutas del vértice dado a los demás vértices.
10. Indique si en el digrafo existen fuentes, y de haberlas indique cuántas hay y cuáles son.
11. Indique si en el digrafo existen sumideros, y de haberlas indique cuántos hay y cuáles son.
12. Dado un vértice, calcule el costo mínimo del vértice dado a los demás vértices utilizando el algoritmo de Dijkstra.
13. Utilizando el algoritmo de Floyd, encuentre la matriz de costo mínimo.
14. Utilizando el algoritmo de Warshall, encuentre la matriz de clausura transitiva.
15. Utilizando el algoritmo de Floyd-Warshall, halle la matriz de costo mínimo así como sus caminos.
16. Utilice el algoritmo de Ford-Fulkerson para hallar el flujo máximo posible desde la fuente al sumidero.