GRAFOS

1. Definición
2. Representación gráfica
3. Terminología
4. Implementación
5. Algoritmos de búsqueda

SESIÓN **/13**

**INTRODUCCIÓN**

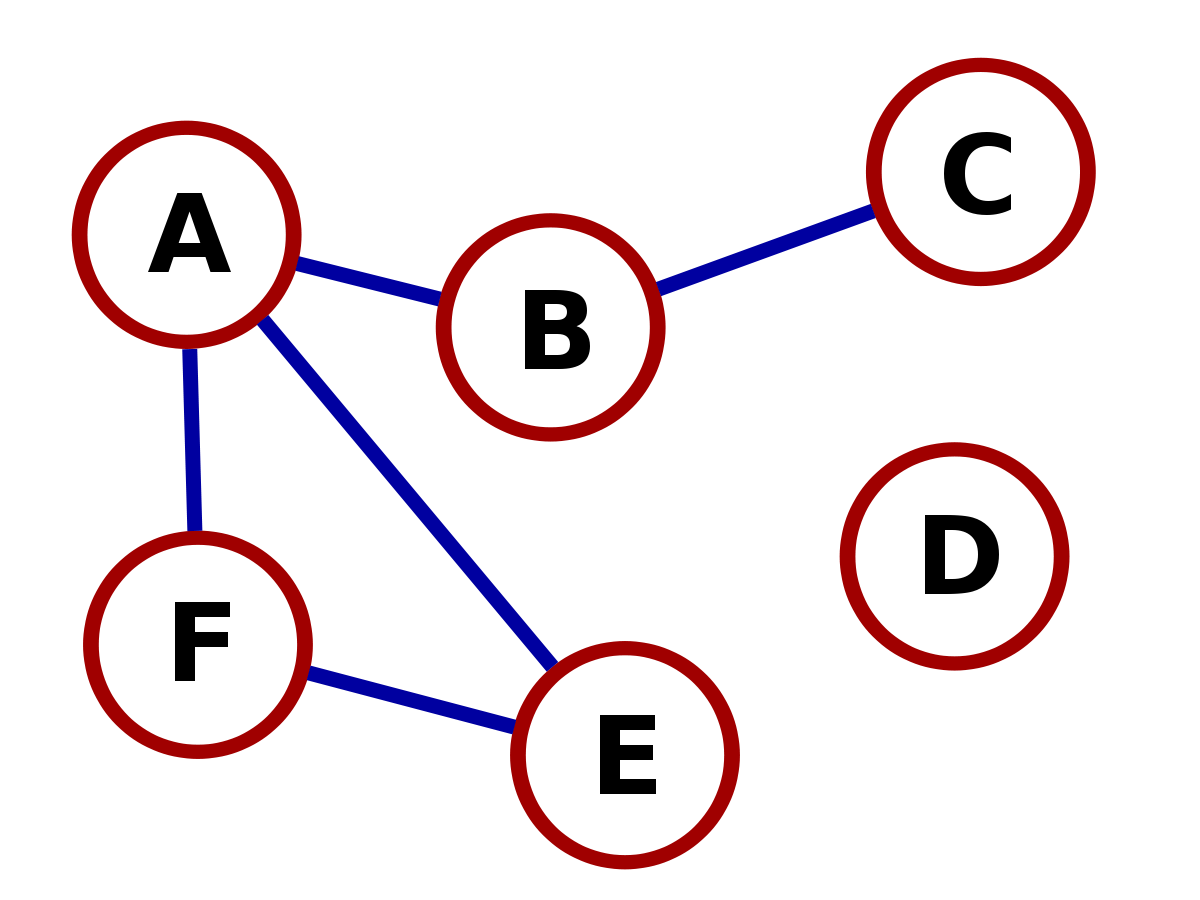
* La clase anterior conocimos los procedimientos recursivos. Describimos la forma cómo aplicarlos y analizamos su eficiencia en comparación con los procedimientos iterativos.
* En la presente sesión conoceremos la quinta Estructura de datos: Los Grafos. Revisaremos sus principales características, su representación gráfica y las operaciones que se pueden realizar con esta estructura.
* Culminaremos describiendo las situaciones del mundo real en las cuales se pueden aplicar la teoría de Grafos.

1. **DEFINICIÓN**

Es un conjunto de elementos llamados vértices (nodos) unidos por enlaces denominados aristas (arcos), que permiten representar las relaciones binarias entre elementos.

***EJEMPLO:***

***Dado el siguiente Grafo:***



***Los vértices ó nodos son : A, B, C, D, E, F***

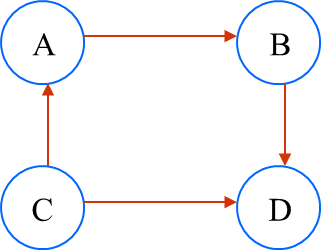
***Las aristas ó arcos son : (A, B), (B, C), (A, F), (A, E), (F, E)***

1. **REPRESENTACIÓN GRÁFICA**
2. Representación de un Grafo dirigido.

Es un grafo que tiene un par ordenado de vértices.

***EJEMPLO:***

***Dado el siguiente Grafo:***



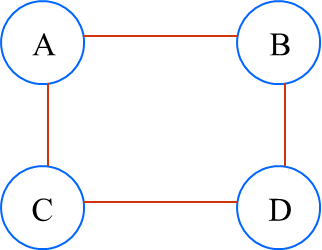
***El par de nodos (A, B) y el par de nodos (B, A) representan dos arcos diferentes.***

1. Representación de un Grafo no dirigido.

Es un grafo que tiene un par NO ordenado de vértices.

***EJEMPLO:***

***Dado el siguiente Grafo:***



***El par de nodos (A, B) y el par de nodos (B, A) representan el mismo arco.***

1. **TERMINOLOGÍA**

En los grafos se utilizan términos adicionales a los utilizados en otras Estructuras de Datos.

* **Vértices adyacentes**

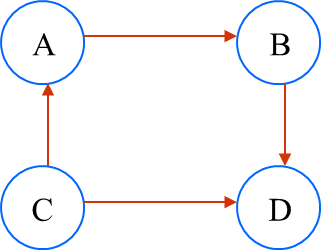
Se dice que dos nodos son adyacentes cuando comparten un mismo arco.

* **Aristas adyacentes**

Se dice que dos arcos son adyacentes cuando comparten un mismo nodo.

***EJEMPLO:***

***Dado el siguiente Grafo:***



***Vértices adyacentes : A y B.***

***Aristas adyacentes : (A, C) y (C, D)***

* **Ponderación**

Es el valor numérico asociado a un arco.

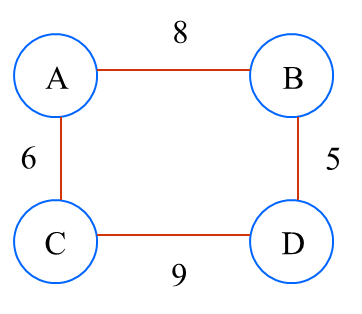
La ponderación puede representar distancia, costo, tiempo, etc.

* **Grafo ponderado**

Es un Grafo en el cual cada nodo tiene una ponderación.

***EJEMPLO:***

***Dado el siguiente Grafo:***



***El arco (A, B) tien una ponderación de 8 unidades.***

1. **IMPLEMENTACIÓN**

La implementación de un Grafo se puede realizar en una de las siguientes estructuras:

* En una Matriz.
* En dos Listas enlazadas simples.

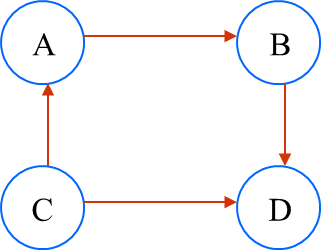
Por la naturaleza del curso, sólo describiremos la implementación de un Grafo en una Matriz.

**Procedimiento**

1. Se crea una Matriz cuadrada de tamaño n, donde n indica la cantidad de nodos.
2. Cada fila y cada columna de la Matriz represeente un nodo.
3. Cada par formado por una fila y una columna representa un arco.
4. Se asigna el valor de 1 en la intersección de la fila y columna que representa el arco.
5. Si el Grafo es ponderado, en lugar de 1 se asigna la ponderación.

***EJEMPLO 1:***

***Para un Grafo dirigido de 4 nodos***



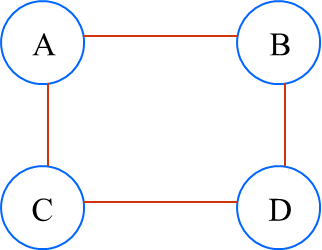
***Se coloca 1 si existe un arco del nodo i al nodo j.***

***Se coloca 0 si no existe un arco del nodo i al nodo j.***

***EJEMPLO 2:***

***Para un Grafo no dirigido de 4 nodos***





***Se coloca 1 si existe un arco entre ambos nodos.***

***Se coloca 0 si no existe un arco entre ambos nodos.***

1. **ALGORITMOS DE BÚSQUEDA**

Los algoritmos de búsqueda se caracterizan por el orden en el cual se expanden (visistan) los nodos. Los algoritmos de búsqueda básicos son los siguientes:

* 1. Búsqueda en profundidad (DFS)

En este algoritmo se expanden los nodos generados más recientemente; es decir, se explora cada camino posible hasta el final antes de intentar otro. La profundidad del nodo inicial es cero y la profundidad de un nodo que no es inicial es igual a uno más la profundidad de su padre.

* 1. Búsqueda en anchura (BFS)

En este algoritmo se expanden los nodos en el orden en que han sido generados; es decir, se explora todos los nodos del mismo nivel antes de generar y analizar un nivel más profundo. Trabaja con dos listas, la primera se utiliza para almacenar los nodos que se van a expander, mientras que la segunda almacenará los nodos que ya fueron expandidos.

**CONCLUSIÓN**

Independientemente de la forma de implementación, los grafos representan una Estructura de datos no lineal. Los valores almacenados en una Matriz de adyacencia representan la existencia de enlaces (arcos) y, de ser el caso, su ponderación. La eficiencia de un Grafo dependerá de la Estructura que se utilizó para su implementación (Matriz ó Listas enlazadas simples).

**REFERENCIAS BIBLIOGRÁFICAS**

* Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., Mexico: McGraw Hill.
* Instituto NIIT (2011). Data Structures and Algorithms. Student guide.