

Análisis de Algoritmos y Estructura de Datos

Ejercicios: TDA grafo y recorrido BFS

Prof. Violeta Chang C

Semestre 2 – 2023



TDA grafo

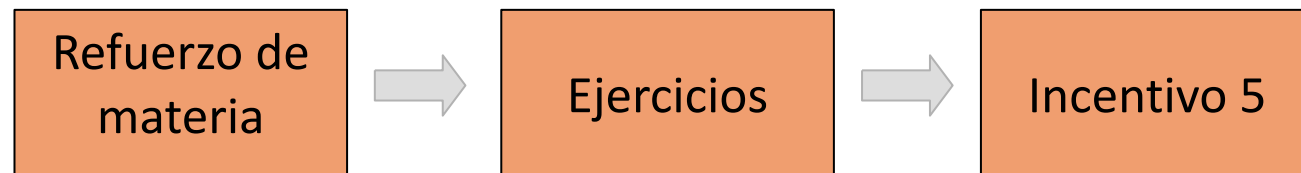
- **Contenidos:**

- Terminología de grafos
- Representación de estructura de datos grafo
- Definición formal de TDA grafo
- Algoritmo de recorrido BFS en grafos

- **Objetivos:**

- Dominar nomenclatura relacionada a grafos para sentar base de conocimiento posterior
- Modelar problemas usando grafos
- Construir algoritmos en pseudocódigo para el manejo de grafos
- Aplicar algoritmo de recorrido BFS para resolución de problemas

Ruta de la sesión

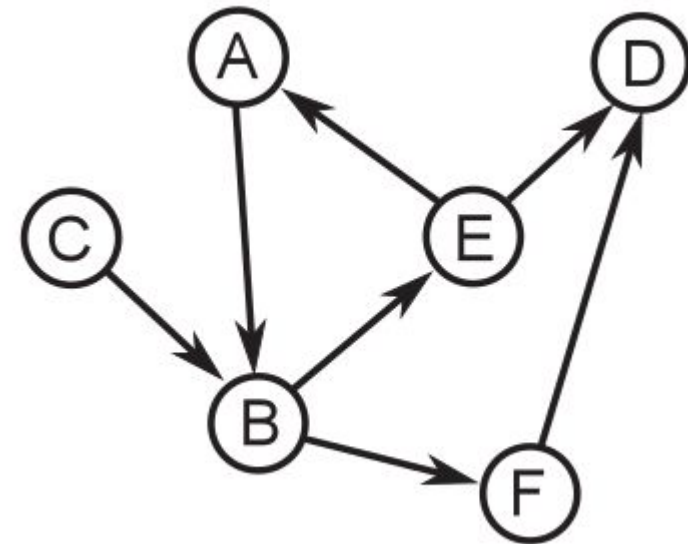


Refuerzo de materia



Terminología de grafos

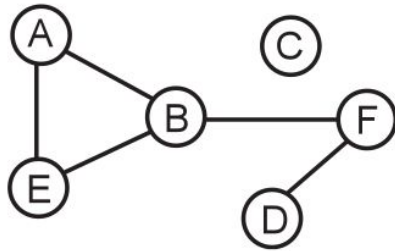
- Un grafo G se define con:
 - **Conjunto de vértices**
 $V = \{A, B, C, D, E, F\}$
 - **Conjunto de aristas**
 $A = \{(A, B), (B, E), (B, F), (C, B), (E, D), (F, D)\}$



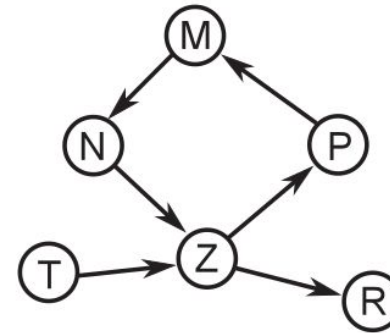


Tipos de grafos

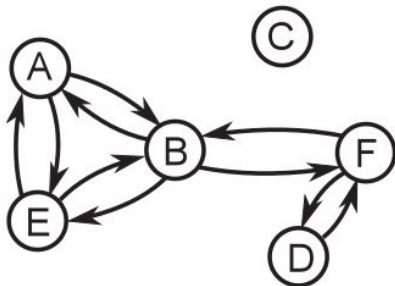
- Ponderado/dirigido



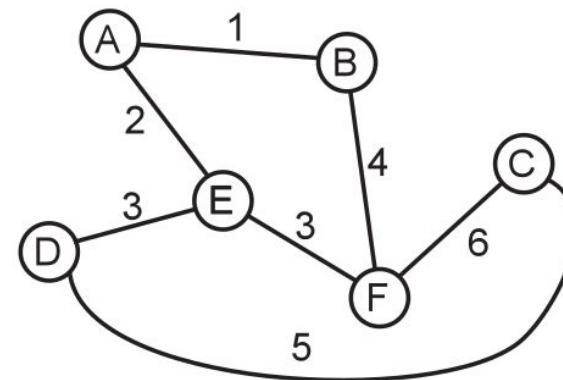
Grafo no dirigido G1



Grafo dirigido G2



Versión dirigida de grafo G1

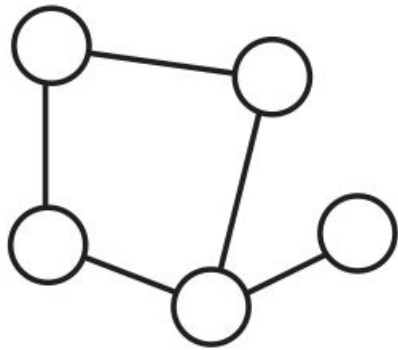


Grafo no dirigido con pesos en las aristas

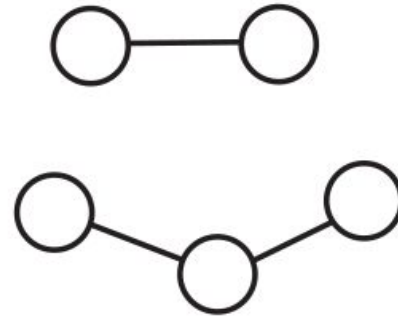


Tipos de grafos

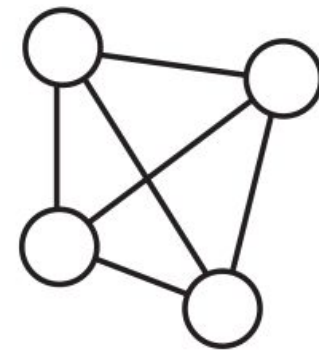
- Conexión en grafos



Grafo conexo



Grafo desconexo

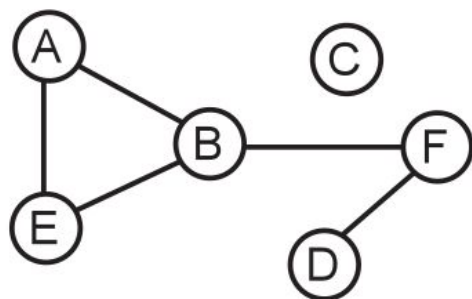


Grafo completo



Representación de grafos

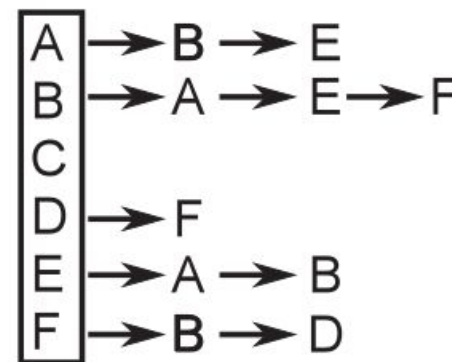
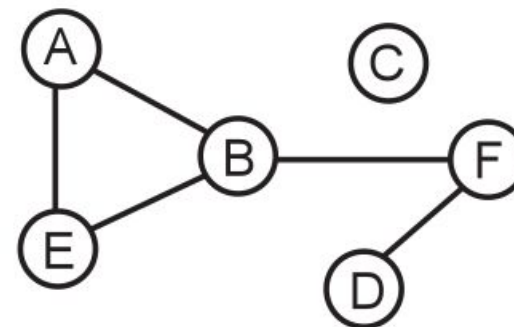
Matriz de adyacencia



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 1 | 0 |
| B | 1 | 0 | 0 | 0 | 1 | 1 |
| C | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 0 | 0 | 1 |
| E | 1 | 1 | 0 | 0 | 0 | 0 |
| F | 0 | 1 | 0 | 1 | 0 | 0 |

Matriz de adyacencia de un
grafo no dirigido

Lista de adyacencia



Lista de adyacencia de un
grafo no dirigido

Definición de TDA grafo

- **Estructura de datos**

- Un grafo es una colección no lineal de elementos homogéneos que se entiende como un conjunto de vértices y un conjunto de aristas que conectan dichos vértices. Las aristas pueden tener un peso asociado.
- La estructura de datos que representa un grafo consiste de una de las siguientes alternativas:
 - Número de vértices + listas de adyacencia
 - Número de vértices + matriz de adyacencia



Definición de TDA grafo

- **Operaciones**
 - ***crearGrafo(V,A)***
 - ***agregarVertice(v)***
 - ***agregarArista(v1,v2)***
 - ***eliminarVertice(v)***
 - ***eliminarArista(v1,v2)***
 - ***obtenerAdyacentes(v)***
 - ***recorrerGrafo(g)***
 - ***obtenerCaminoMinimo(g,v1,v2)***
 - ***obtenerArbolCoberturaMinimo(g)***
 - ***calcularMaximoFlujo(g)***

Actividades



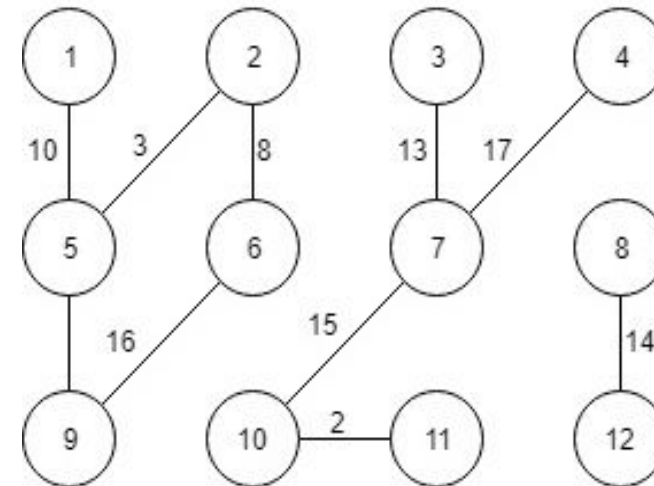
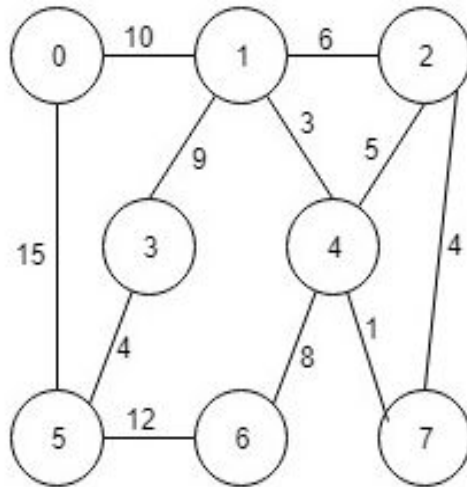
Actividad en parejas - 20 minutos

- **Ejercicio 1**: Escribir un algoritmo en pseudocódigo que reciba un grafo no dirigido $G(V,A)$ y una secuencia de nodos S , y que indique si S es un camino del grafo G . Calcular y justificar la complejidad del algoritmo propuesto.
 - a) Asumir que G se representa usando matriz de adyacencia
 - b) Asumir que G se representa usando listas de adyacencia



Actividad individual - asíncrona

- **Ejercicio 2**: Para cada uno de los grafos, mostrar el recorrido en amplitud. Considerar el vértice 2 como el vértice inicial.





Actividad en conjunto/parejas

20 minutos

- **Ejercicio 3**: el encargado/a de mantener la red de datos de una determinada empresa necesita modelar su red como un grafo y escribir un algoritmo para cada uno de los siguientes problemas:
 - Saber si la red está activa para todos los lugares importantes de la organización.
 - Conocer cuántos grupos aislados de la red existen.
 - Suponiendo que se propaga un virus desde un equipo A, determinar en cuántos pasos podría llegar al equipo B.



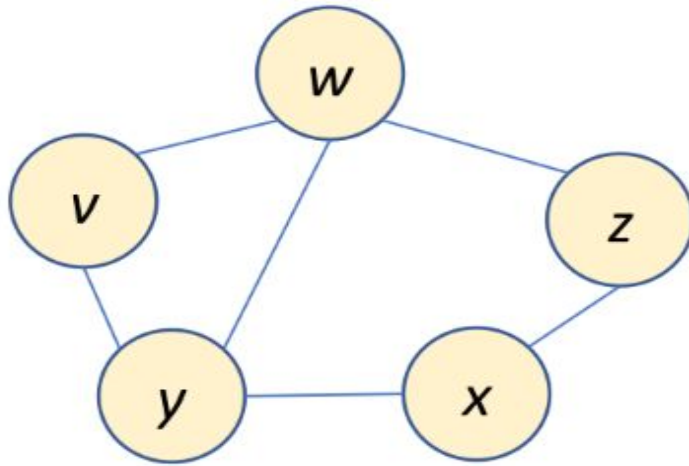
Actividad individual - asíncrona

- **Ejercicio 4**: Escribir un algoritmo en pseudocódigo para verificar si un grafo $G(V,A)$ es o no un grafo completo. Calcular y justificar la complejidad del algoritmo propuesto.
 - a) Asumir que G se representa usando matriz de adyacencia
 - b) Asumir que G se representa usando listas de adyacencia

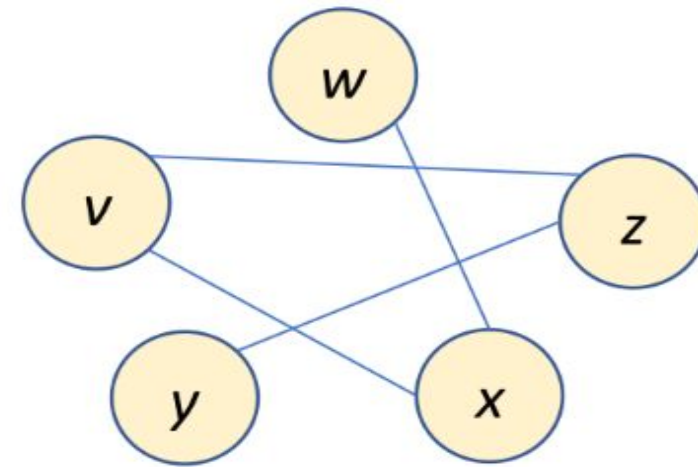


Incentivo 5 - 15+5 minutos

- Dado un grafo $G(V,A)$, representado por lista de adyacencia, construir un algoritmo que encuentre el grafo complemento G' . ¿Cuál es el orden de complejidad del algoritmo propuesto?



Grafo G



Grafo complemento G'



Incentivo 5 - 15+5 minutos

Puntaje

| | | |
|-------|---|--|
| I. | El algoritmo propuesto apunta a resolver el problema planteado. (SI 1 punto / NO 0 punto) | |
| II. | El algoritmo resuelve correctamente el problema planteado. (SI 1 punto / NO 0 punto) | |
| III. | El algoritmo está escrito en pseudocódigo ordenado (SI 1 punto / NO 0 punto) | |
| IV. | El algoritmo está escrito en el formato establecido (SI 1 punto / NO 0 punto) | |
| V. | El algoritmo identifica entradas correctamente (SI 1 punto / NO 0 punto) | |
| VI. | El algoritmo identifica y declara salidas de manera correcta. (SI 1 punto / NO 0 punto) | |
| VII. | Calcula correctamente la complejidad (SI 1 punto / NO 0 punto) | |
| VIII. | Justifica la complejidad del algoritmo propuesto. (SI 1 punto / NO 0 punto) | |
| | PUNTOS | |

Actividad de cierre



- Ir a [menti.com](https://www.menti.com) e ingresar código 3972 4341



Próximas fechas...

- Resumen de la semana:
 - Definiciones y tipos de grafos
 - TDA grafo
 - Modelamiento usando grafos
 - Algoritmo de recorrido BFS

~~cátedra~~ – ~~refuerzo~~ – laboratorio

U3 - S8

- Sub-siguiente semana:
 - Algoritmo de recorrido DFS
 - Algoritmo de camino mínimo Dijkstra
 - Algoritmo de flujo máximo Ford-Fulkerson

| Octubre 2023 | | | | | | |
|--------------|-------|--------|-----------|--------|---------|--------|
| Domingo | Lunes | Martes | Miércoles | Jueves | Viernes | Sábado |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | | | | | | |

| Noviembre 2023 | | | | | | |
|----------------|-------|--------|-----------|--------|---------|--------|
| Domingo | Lunes | Martes | Miércoles | Jueves | Viernes | Sábado |
| Receso | | | | | | |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | | |