



Algoritmos y Estructura de Datos

Unidad 2 – Semana 13



Logro de sesión

Al finalizar la sesión, el estudiante construye aplicaciones basada en grafos para el procesamiento de grandes cantidades de datos.



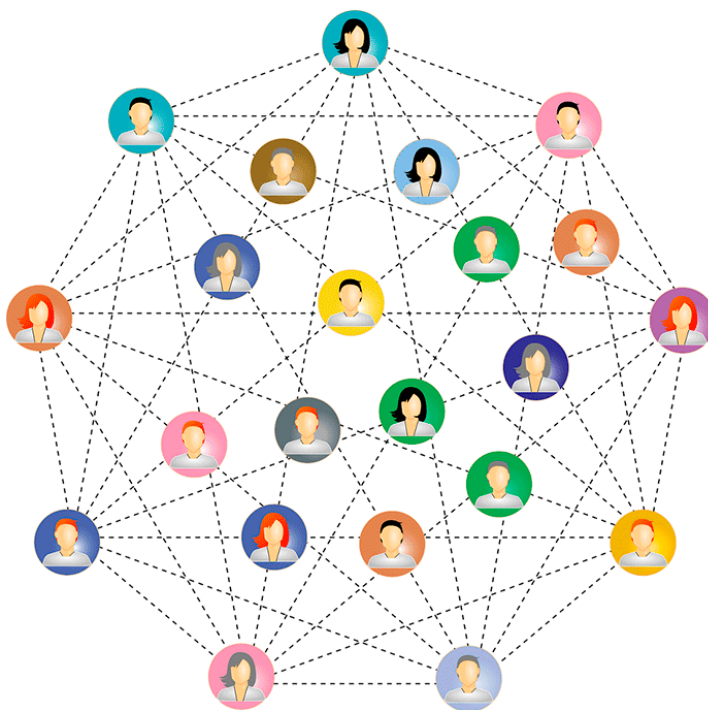
Tema : Grafos

Contenido:

- Introducción
- Grafos
- Ejercicios



Grafos



Introducción



Consideraciones:

- Los grafos son una estructura de datos generalizada en ciencias de la computación.
- Hay cientos de interesantes problemas de computación definidos en términos de grafos.

Grafos

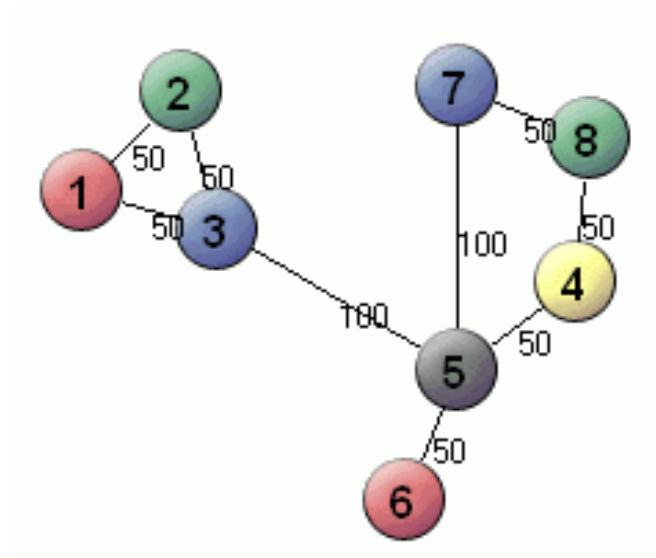
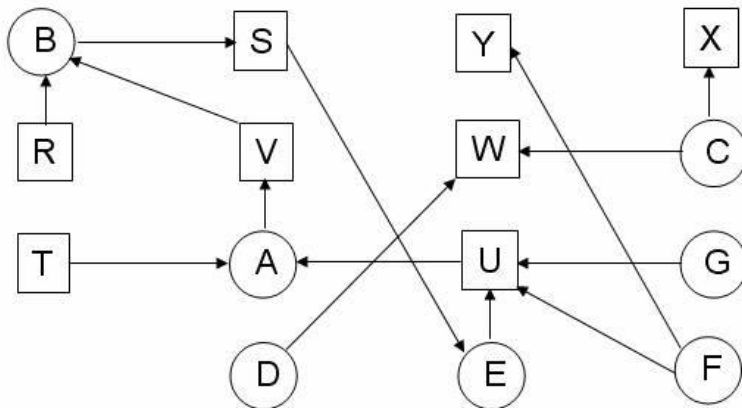
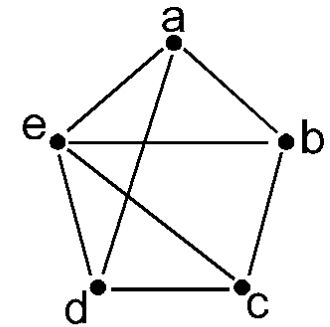
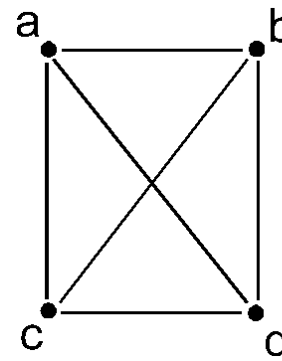


Definición:

- Un grafo es una estructura de datos que permite almacenar una colección de datos en los que cada elemento puede estar relacionado mediante conexiones con otros elementos de la misma colección.
- Estas conexiones pueden ser dirigidas ó no. En el caso de que sean orientadas entonces la relación entre los elementos se puede ver con predecesores y sucesores.
- Un grafo $G = (V, A)$ está formado por (V) értices (ó Nodos) y (A) ristas (ó Arcos) que conectan los vértices. Cada nodo es un par (v, w) donde v, w pertenecen a V .

```

graph TD
    A ---|1| D
    D ---|6| F
    D ---|7| G
    F ---|9| I
    G ---|7| H
    H ---|8| E
    H ---|10| J
    E ---|2| B
    E ---|3| C
  
```



Grafos



Ejemplo:



Grafos



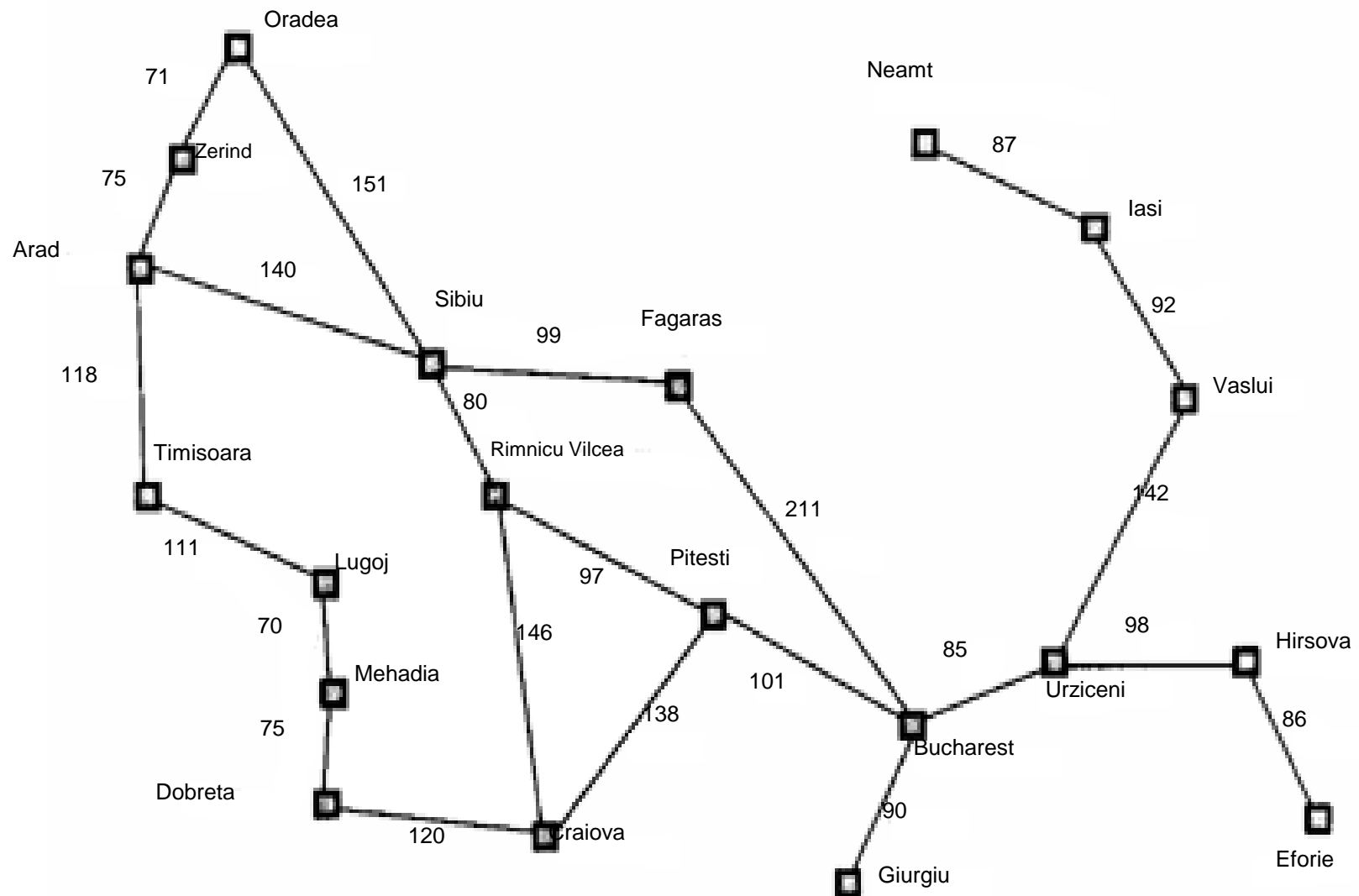
Ejemplo:



Grafos



Ejemplo:

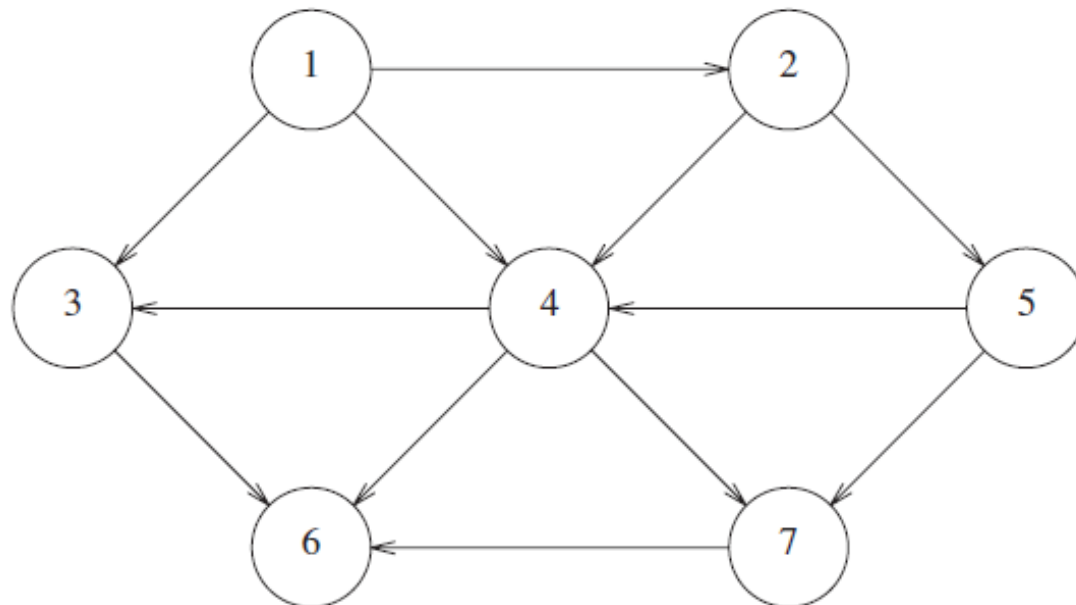


Grafos



Representación:

- Consideraremos grafos dirigidos (los grafos no dirigidos tienen una representación similar).



Grafos



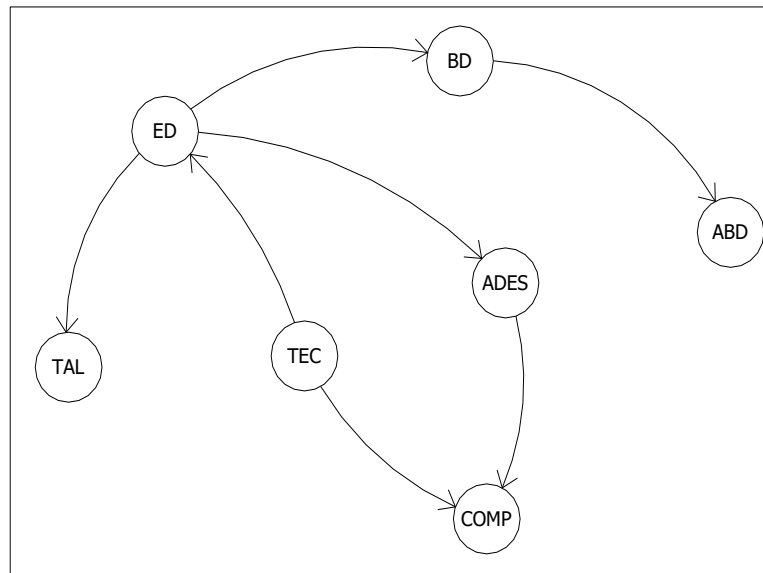
Representación:

- Una forma simple de representar un grafo es usando un arreglo bidimensionales. Eso es conocido como una matriz de adyacencia.

Grafos



Matriz de
adyacencia:



	ED	BD	ADB	TAL	TEC	ADES	COMP
ED	0	1	0	1	0	1	0
BD	0	0	1	0	0	0	0
ADB	0	0	0	0	0	0	0
TAL	0	0	0	0	0	0	0
TEC	1	0	0	0	0	0	1
ADES	0	0	0	0	0	0	1
COMP	0	0	0	0	0	0	0

Grafos



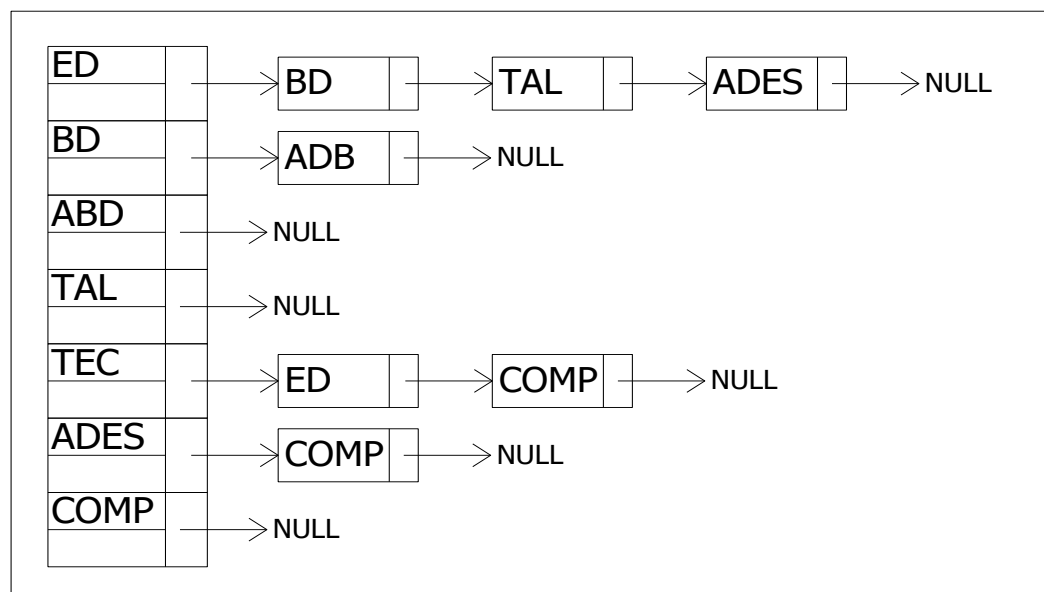
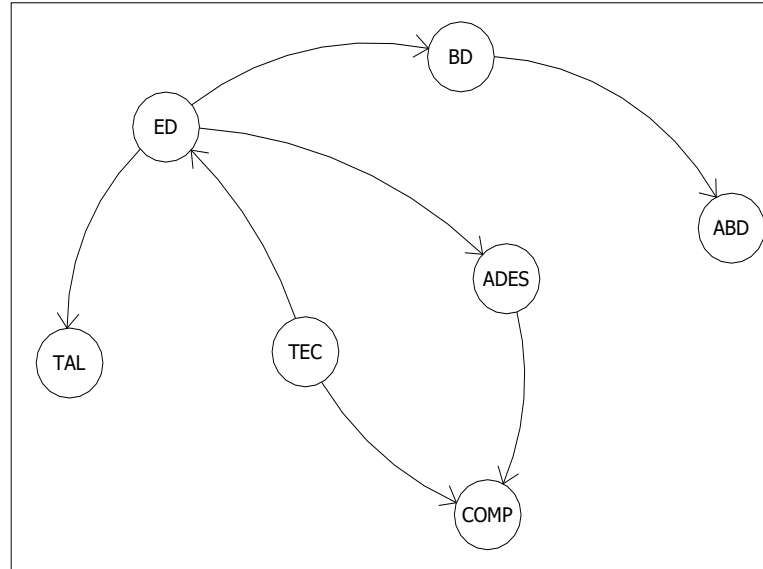
Representación:

- Si el grafo no es muy denso, en otras palabras el grafo es espaciado, la mejor solución es una lista de adyacencia. Por cada vértice, mantenemos una lista de todos los vértices adyacentes.

Grafos



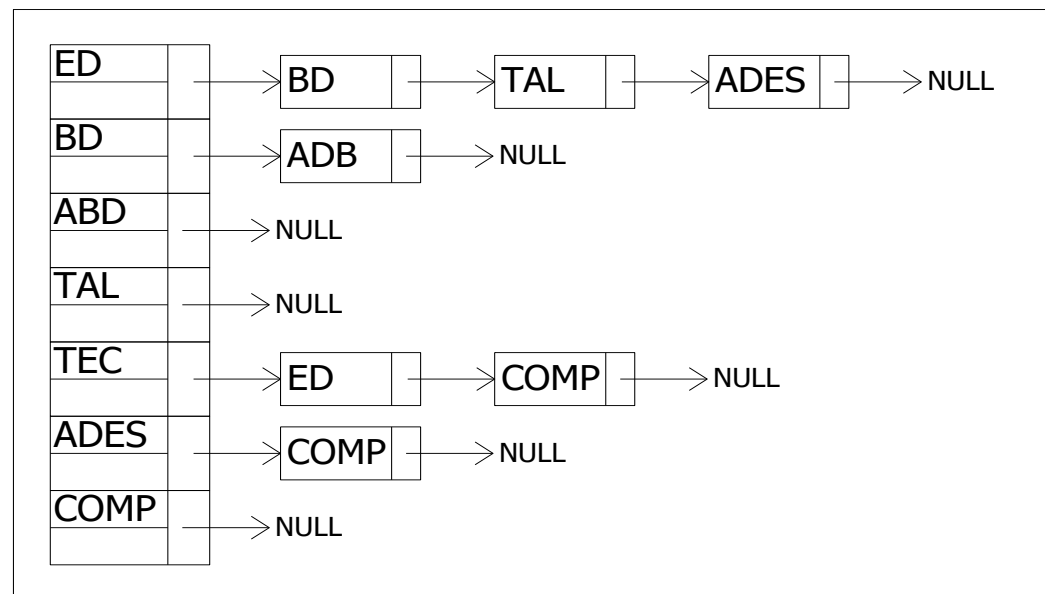
Lista de
adyacencia:



Grafos



Implementación (Lista de adyacencia):



Se necesita la lista de vértices

De cada vértice la información y la lista de adyacencia (de arcos)

De cada arco se necesita la información y el vértice de llegada

Ejercicios



- Ordenamiento topológico
 - Es un ordenamiento de vértices en un grafo acíclico dirigido, tal que si este es un camino de v_i , v_j , entonces v_j aparece después de v_i en el ordenamiento.
 - La figura siguiente representa una estructura de prerrequisitos de cursos de una universidad estatal de Miami. Un arco dirigido (v, w) indica que el curso v debe ser completado antes que el curso w pueda ser aceptado.
 - Un ordenamiento topológico de estos cursos es cualquier secuencia de cursos que no debe violar el requerimiento de prerrequisito.

Ejercicios



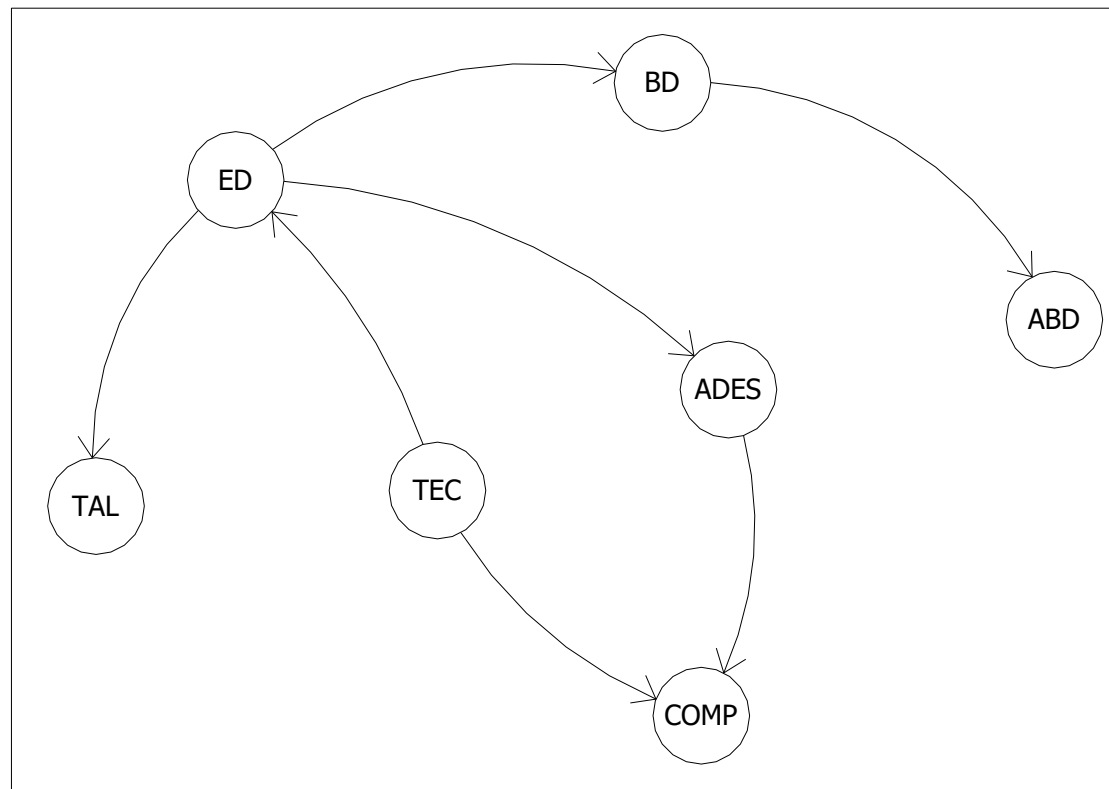
- Ordenamiento topológico
 - Asignaturas y relaciones

Asignatura	Prerrequisito de
TEC	ED
ED	BD
ED	TAL
ED	ADES
BD	ABD
TEC	COMP
ADES	COMP

Ejercicios



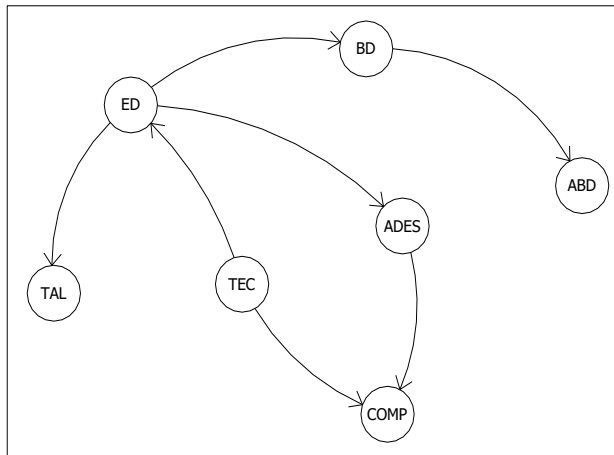
- Ordenamiento topológico
 - Grafo



Ejercicios

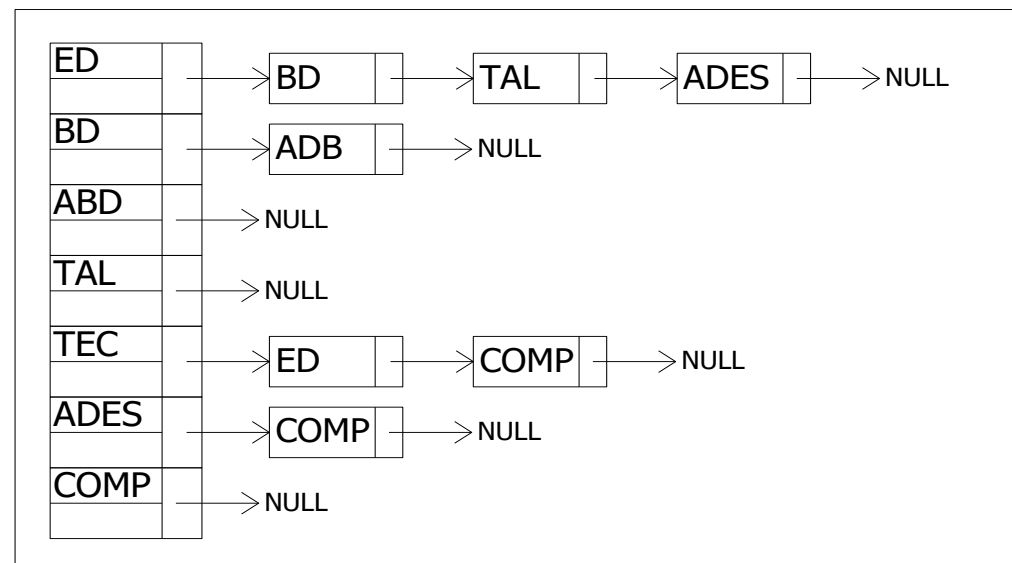


Matriz de Adyacencia



Listas de Adyacencia

	ED	BD	ABD	TAL	TEC	ADES	COMP
ED		1		1		1	
BD			1				
ABD							
TAL							
TEC	1						1
ADES							1
COMP							

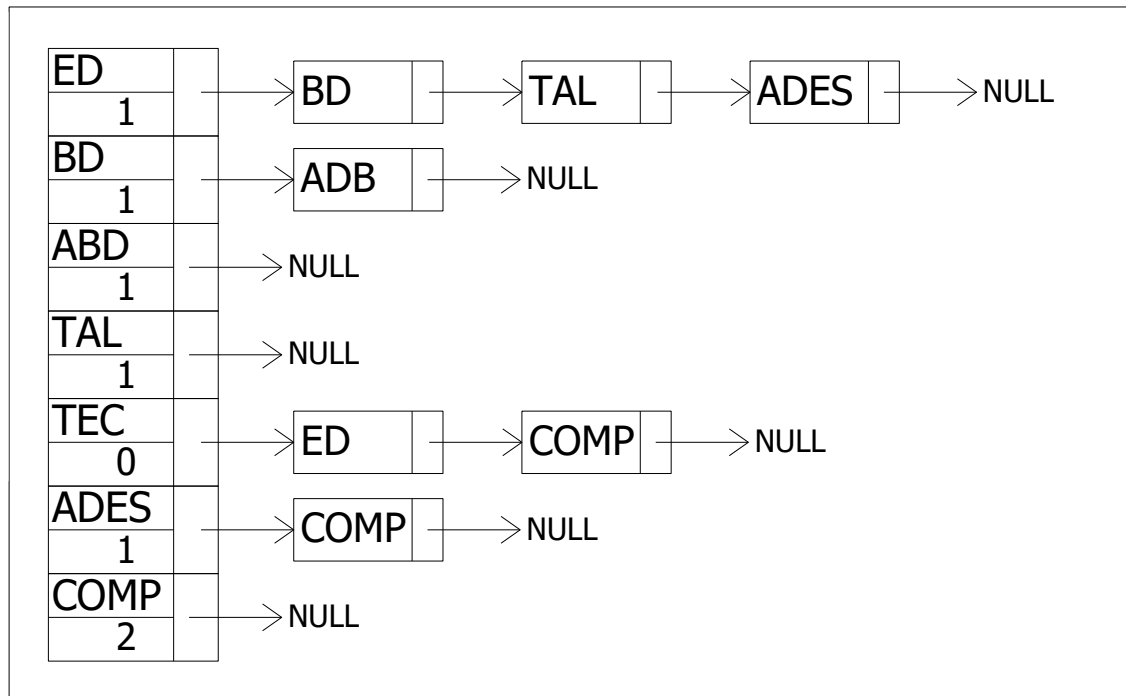
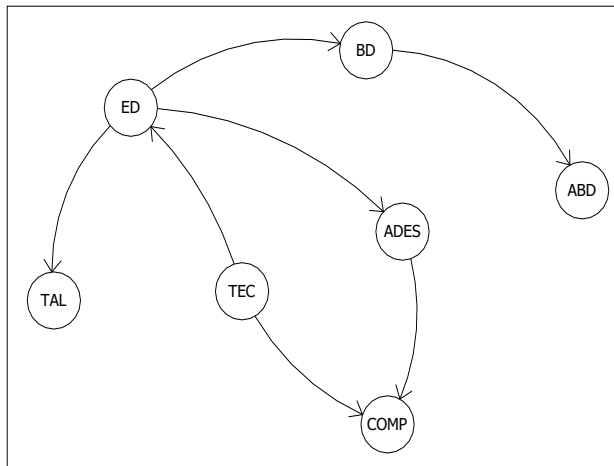


Ejercicios



Método general:

Calcular para cada Vértice, su grado de entrada
(cantidad de aristas que llegan al Vértice)

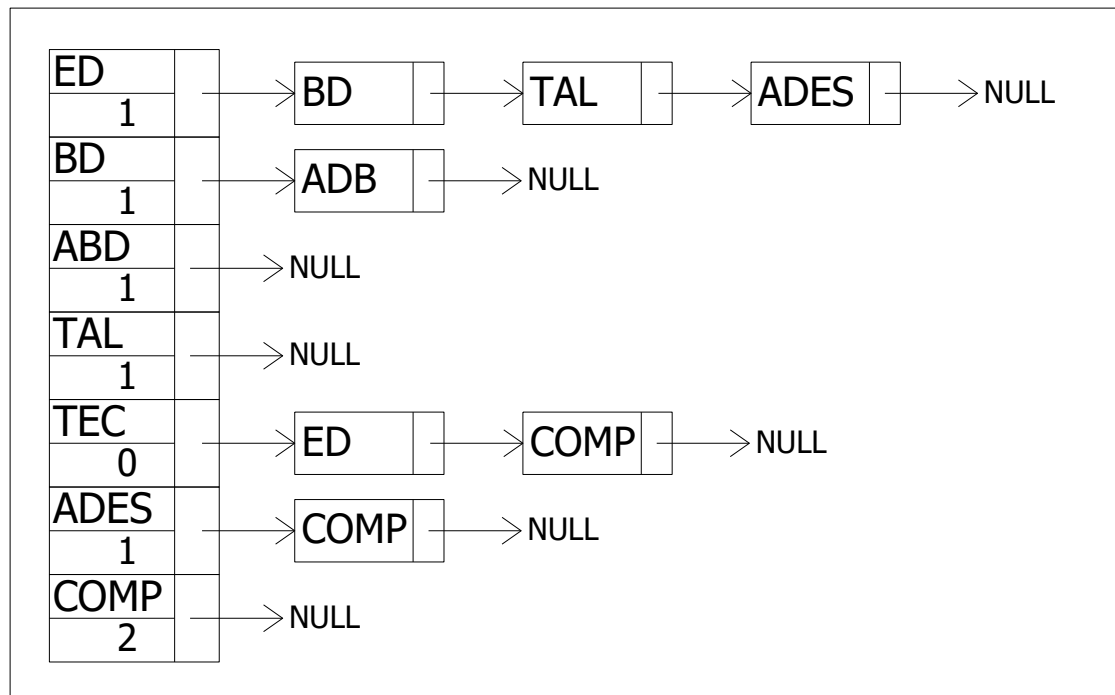


Ejercicios



Colocar en una Cola (ó Pila) aquellos vértices con grado de entrada igual a cero

COLA:	TEC						
-------	------------	--	--	--	--	--	--



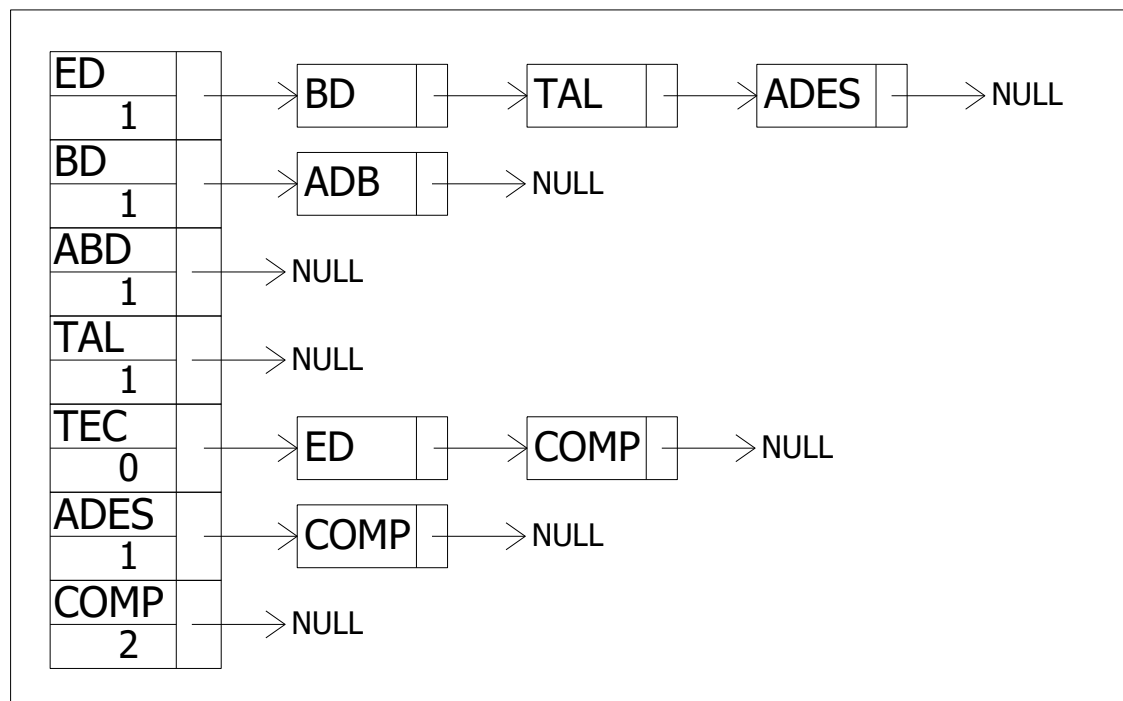
Ejercicios



Mientras la Cola no sea vacía:

- Sacar de la cola un Vértice
- Colocarlo en el ordenamiento de salida
- Decrementar en Uno el grado de entrada de sus vértices adyacentes. Colocar en la Cola aquellos cuyos grados de entrada tomen valor Cero.

COLA:	TEC						
-------	-----	--	--	--	--	--	--



A la salida:
TEC

Decrementar en Uno
los grados de Entrada de:
ED
COMP

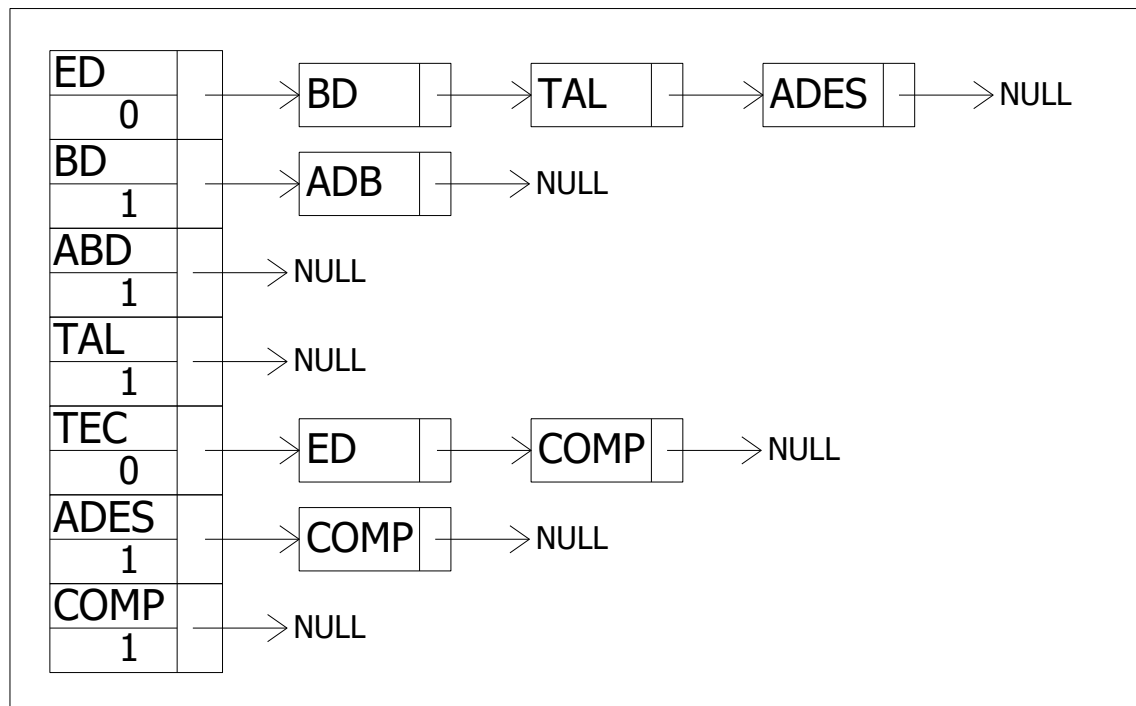
Ejercicios



Mientras la Cola no sea vacía:

- Sacar de la cola un Vértice
- Colocar en el ordenamiento de salida
- Decrementar en Uno el grado de entrada de sus vértices adyacentes. Colocar en la Cola aquellos cuyos grados de entrada tomen valor Cero.

COLA:	<i>TEC</i>	ED					
-------	------------	-----------	--	--	--	--	--



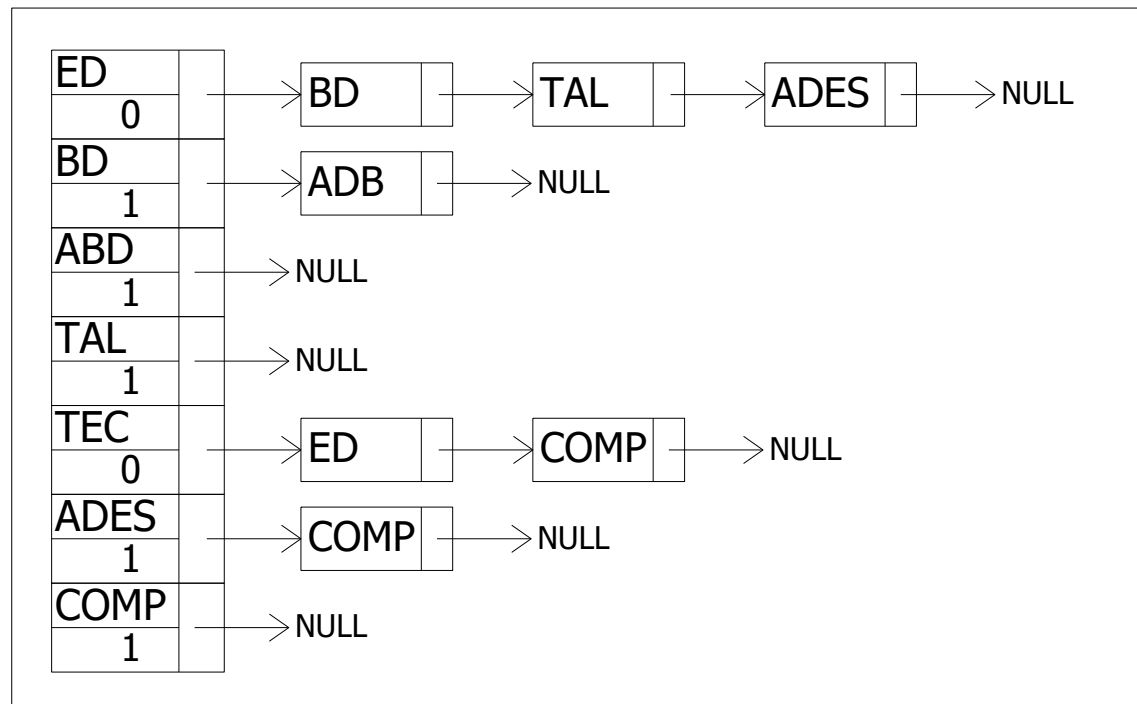
Ejercicios



Mientras la Cola no sea vacía:

- Sacar de la cola un Vértice
- Colocarlo en el ordenamiento de salida
- Decrementar en Uno el grado de entrada de sus vértices adyacentes. Colocar en la Cola aquellos cuyos grados de entrada tomen valor Cero.

COLA:	TEC	ED					
-------	-----	----	--	--	--	--	--



A la salida:

ED

Decrementar en Uno
los grados de Entrada de:

BD

TAL

ADES

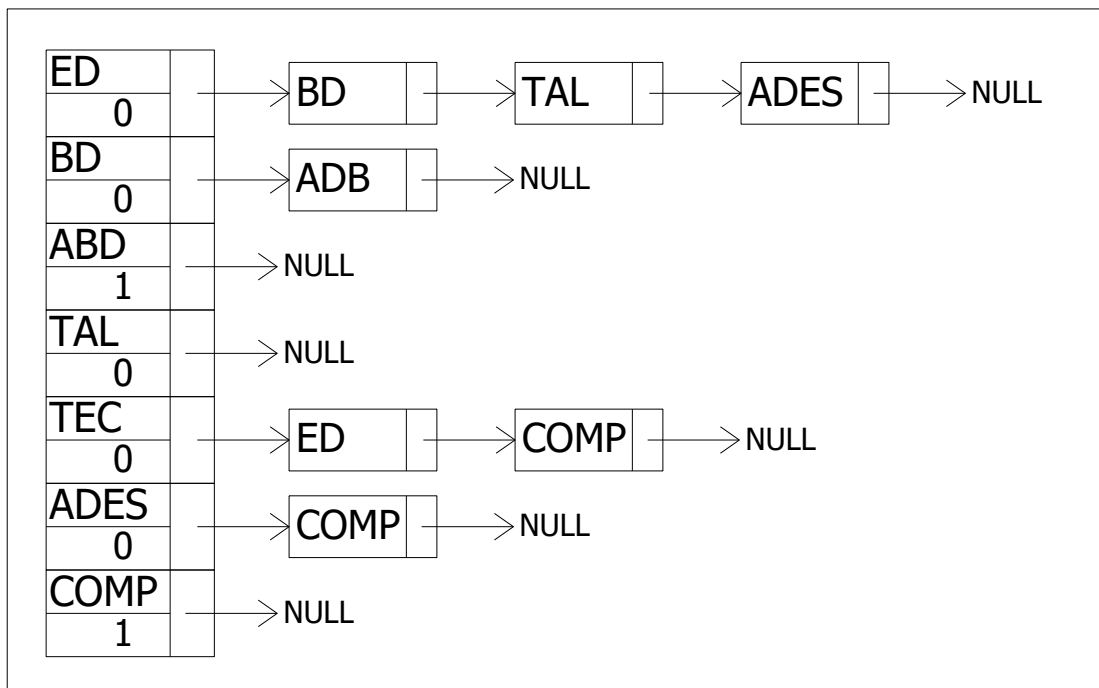
Ejercicios



Mientras la Cola no sea vacía:

- Sacar de la cola un Vértice
- Colocar en el ordenamiento de salida
- Decrementar en Uno el grado de entrada de sus vértices adyacentes. Colocar en la Cola aquellos cuyos grados de entrada tomen valor Cero.

COLA:	<i>TEC</i>	<i>ED</i>	BD	TAL	ADES		
-------	------------	-----------	-----------	-----	------	--	--

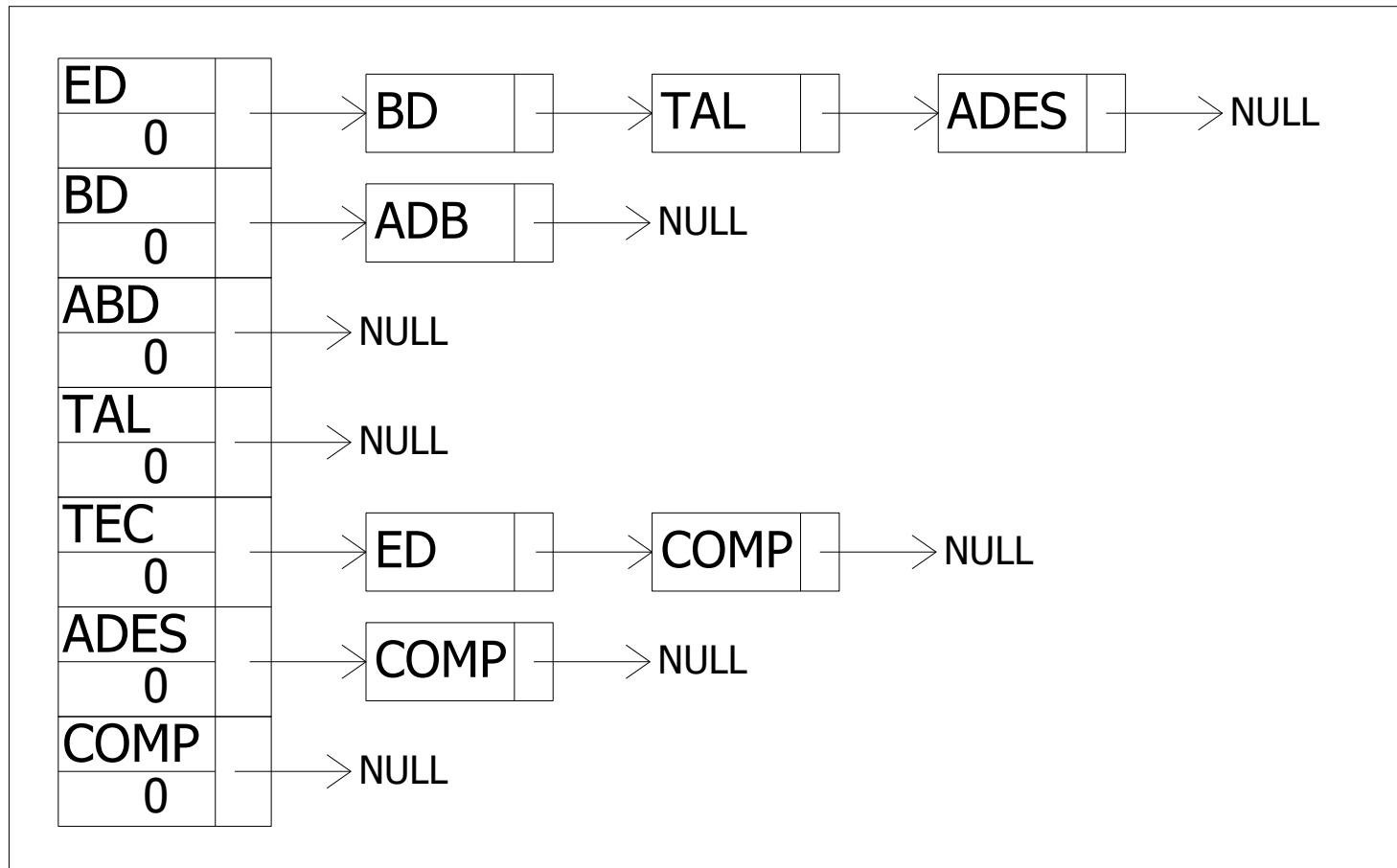


Ejercicios



Configuración Final

COLA(vacía)	<i>TEC</i>	<i>ED</i>	<i>BD</i>	<i>TAL</i>	<i>ADES</i>	<i>ABD</i>	<i>COMP</i>
-------------	------------	-----------	-----------	------------	-------------	------------	-------------



Ejercicios



- Implementar el algoritmo de Dijkstra

Referencias



- ❑ Sedgewick, R., et. al. (2011) Algorithms, Fourth Edition. Pearson.
- ❑ Cormen, H., et. al. (2009) Introduction to Algorithms, MIT Press.
- ❑ Allen, Mark (2014) Data Structures and Algorithms Analysis in C++, Fourth Edition. Pearson.



EXIGETE INNOVA