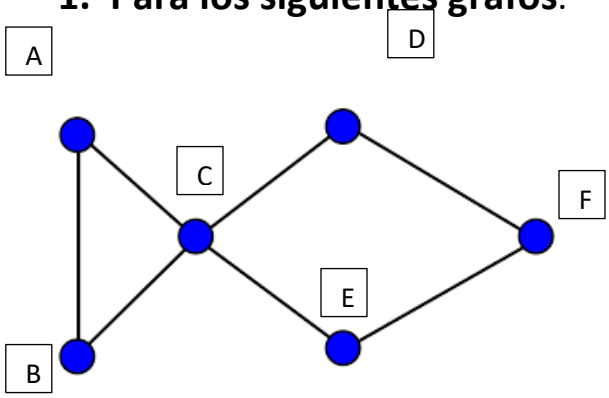
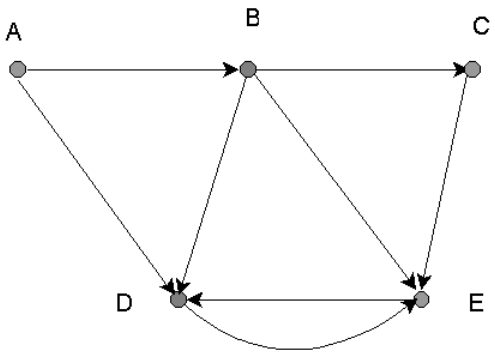
CORPORACION UNIVERSITARIA REMINGTON

TALLER DE ESTRUCTURAS DE DATOS

1. **Para los siguientes grafos**:





Conteste las siguientes preguntas:

a. Explique la diferencia entre los dos grafos anteriores

b. En el grafo dirigido hay una trayectoria para ir de D hasta A?. sí la hay describa la trayectoria y si no que le colocaría al grafo para que se de esa trayectoria y escríbala.

c. Diga cual es el número máximo de lados que pueden tener los dos grafos

d. Represente el grafo dirigido con matriz de adyacencia

e. Represente el grafo no dirigido con lista ligada de adyacencia

f. Represente el grafo dirigido con matriz de incidencia

g. Represente el grafo dirigido con lista ligada de adyacencia

h. Cuantos ciclos se pueden dar en ambos gafos y escriba cada uno de ellos

i. Hallar el grado para cada uno de los vértices de cada grafo

2. Construir un algoritmo que permita crear la matriz de adyacencia en un grafo no dirigido.

3. Construir un algoritmo que permita crear la matriz de incidencia en un grafo no dirigido.

4. Defina con sus palabras:

a) Adyacencia

b) Incidencia

c) Grado de un grafo

d) Trayectoria

e) Trayectoria simple

f) Ciclo

g) Grafo conectado

h) Grafo fuertemente conectado

5. Investigar los Recorridos sobre grafos:

5.1 Investigar que el Recorrido DFS sobre grafos y un ejemplo

5.2 Investigar que el Recorrido BFS sobre grafos y un ejemplo

SOLUCIÓN

1)

A) Explique la diferencia entre los dos grafos anteriores

- La diferencia entre los grafos como podemos observar uno tiene flechas y el otro no, es decir que uno es un grafo dirigido y el otro no.

B) **¿En el grafo dirigido hay una trayectoria para ir de D hasta A? sí la hay describa la trayectoria y si no que le colocaría al grafo para que se de esa trayectoria y escríbala**

- No tiene trayectoria ya que las flechas provocan un bucle en el recorrido de D hasta E y viceversa.

C) Diga cuál es el número máximo de lados que pueden tener los dos grafos.

Grafo 1:

- N=6
  6*(6-1) /2
  6*(5) /2
  30/2
  =15
  El número máximo de lados que tiene el grafo 1 es de 15.

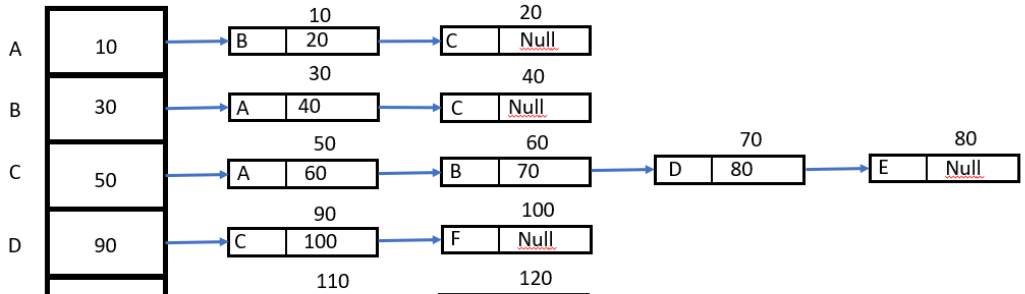
Grafo 2:

- N= 5
  5*(5-1)
  5*(4)
  =20
  El número máximo de lados en el grafo 2 es de 20.

D) Represente el grafo dirigido con matriz de adyacencia.

	A	B	C	D	E
A		1		1	
B			1	1	1
C				1	
D					1
E				1	

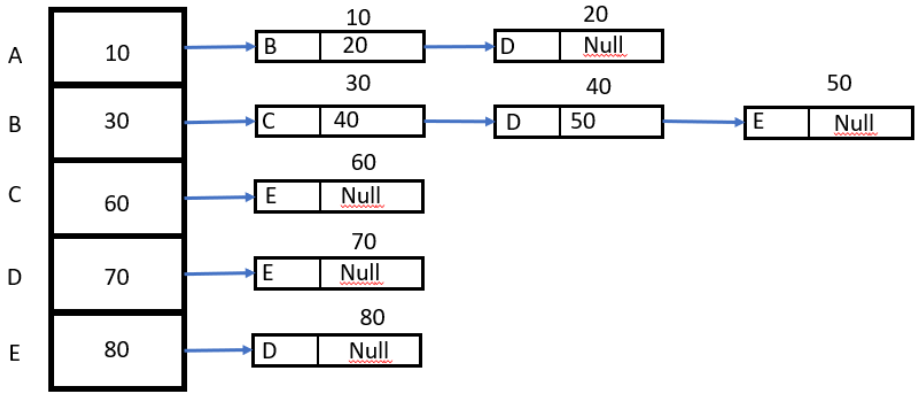
E) Represente el grafo no dirigido con lista ligada de adyacencia.



F) Represente el grafo dirigido con matriz de incidencia.

In	1	2	3	4	5	6	7	8
A	1							1
B		1				1	1	
C			1					
D					1			
E				1				

G) Represente el grafo dirigido con lista ligada de adyacencia.



H) Cuantos ciclos se pueden dar en ambos gafos y escriba cada uno de ellos.

Grafo 1:

- A, B, C, A
- C, D, F, E, C

Grafo 2:

- D, E, D

I) Hallar el grado para cada uno de los vértices de cada grafo.

Grafo 1:

- A es de grado 2
- B es de grado 2
- C es de grado 4
- D es de grado 2
- F es de grado 2
- E es de grado 2

Grafo 2:

- A
  Salida: 3
  Entrada: 0
  Total: 4

- B
  Salida: 3
  Entrada: 1
  Total: 4

- C
  Salida: 1
  Entrada: 1
  Total:2

- D
  Salida: 1
  Entrada: 3
  4

- E
  Salida: 1
  Entrada: 3
  Total: 4

2) **Construir un algoritmo que permita crear la matriz de adyacencia en un grafo no dirigido.**

```
Función matriz_de_adyacencia(grafo):
    num_vertices = Longitud(grafo)
    matriz = CrearMatrizDeCeros(num_vertices, num_vertices)

    Para i de 0 hasta num_vertices - 1:
        Para cada vecino en grafo[i]:
            matriz[i][vecino] = 1
            matriz[vecino][i] = 1  // Debido a que es un grafo no dirigido
    retornar matriz
```

3) **Construir un algoritmo que permita crear la matriz de incidencia en un grafo no dirigido.**

```
Función matriz_de_incidencia(grafo):
    num_vertices = Longitud(grafo)
    num_aristas = 0
    Para cada vecinos en grafo:
        num_aristas += Longitud(vecinos)
    num_aristas = num_aristas / 2  // Dividido por 2 debido a que es un grafo no dirigido
    matriz = CrearMatrizDeCeros(num_vertices, num_aristas)

    arista_index = 0
    Para vertice de 0 hasta num_vertices - 1:
        Para cada vecino en grafo[vertice]:
            Si vertice < vecino:
                matriz[vertice][arista_index] = 1
                matriz[vecino][arista_index] = 1
                arista_index += 1

  retornar matriz
```

4) **Defina con tus palabras:**

a) Adyacencia: Dos vértices están adyacentes si están directamente conectados por una línea.

b) Incidencia: Es el lado que forma dos vértices.

c) Grado de un grafo: El grado de un vértice es la cantidad de flechas o líneas que se conectan a ese vértice.

d) Trayectoria: Una secuencia de vértices donde cada par consecutivo está conectado por aristas.

e) Trayectoria simple: Una trayectoria donde no se repiten vértices, excepto el primero y el último.

f) Ciclo: Una trayectoria cerrada que comienza y termina en el mismo vértice.

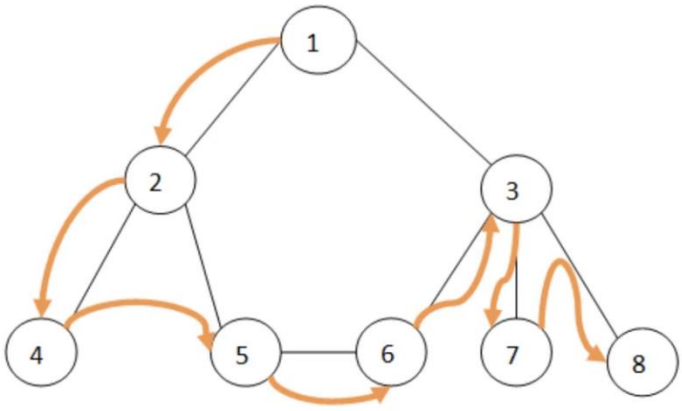
g) Grafo conectado: Hay una trayectoria entre cualquier par de vértices en el grafo.

h) Grafo fuertemente conectado: En un grafo dirigido, hay una trayectoria dirigida entre cualquier par de vértices.

5)

a) Investigar que el Recorrido DFS sobre grafos y un ejemplo

El recorrido DFS (Depth-First Search) es un algoritmo que se utiliza para explorar grafos en el que se explora tan profundamente como sea posible a lo largo de un camino antes de retroceder. Se utiliza para recorrer y buscar en profundidad en estructuras de datos como árboles y grafos.



b) Investigar que el Recorrido BFS sobre grafos y un ejemplo

El recorrido BFS (Breadth-First Search) es un algoritmo que se utiliza para explorar grafos en el que se visitan primero todos los vecinos del vértice actual antes de avanzar a los vecinos de los vecinos. Este enfoque garantiza que los vértices cercanos se visiten antes que los más alejados. Es eficiente y se utiliza para encontrar caminos cortos en grafos no ponderados y para determinar la conectividad de un grafo.