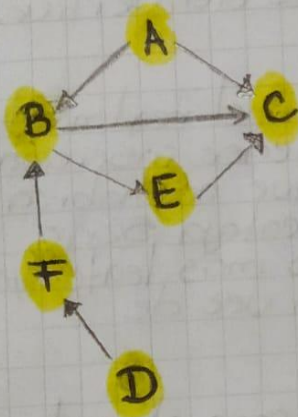


Taller # 5 - Gratos

2- Dibujo o diagrama que represente el grato dirigido



A → B → C

B → C → E

C

D → F

E → C

F → B

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

A → B → C

B → A → C → D

C → A → B → E

D → B → E

E → C → D → F

F → E

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

Preguntas de reflexión

1- Cada que se agregue una arista $i \rightarrow j$, también agregaría automáticamente la arista $j \rightarrow i$.

Y en la lista de adyacencia, se añade el vértice en ambos sentidos, y en la matriz de adyacencia se marca matriz $[i][j] = 1$ y matriz $[j][i] = 1$.

2- Si un vértice deja de existir en la lista de adyacencia se elimina la lista que está en ese vértice y quita las apariciones de ese vértice en las demás listas.
En la matriz eliminaría la fila asociada al vértice y elimina la columna correspondiente del vértice.

3- Si cambiara la lista por arreglos el rendimiento tendría una ventaja la cual es de tener inserciones y eliminaciones más eficientes, ya que no habría que realizar desplazamientos, la desventaja sería que acceder a un nodo específico es más lento, ya que tendría que recorrer la lista en vez de acceder directo al índice.

4- Las ventajas que hay al dibujarlo manualmente genera mejor comprensión visual, me permite resaltar detalles y hacer énfasis en los nodos y sus formas de organizar, pero, al organizarlo en código su ventaja es que es más rápido y más exacto.

Sus desventajas es que en dibujo si te equivocas hay que repetir desde cero el dibujo al igual que si deseas añadir nodos, además de que si el gráfico es grande ya sería más confuso y complicado de mantener, desventaja en código es que si deseas editarlo pueda que el programa no deje en lista de adyacencia el código tiene búsquedas más lentas en gráficos densos y en matriz ocupa más memoria y es menos eficiente en gráficos dispersos.