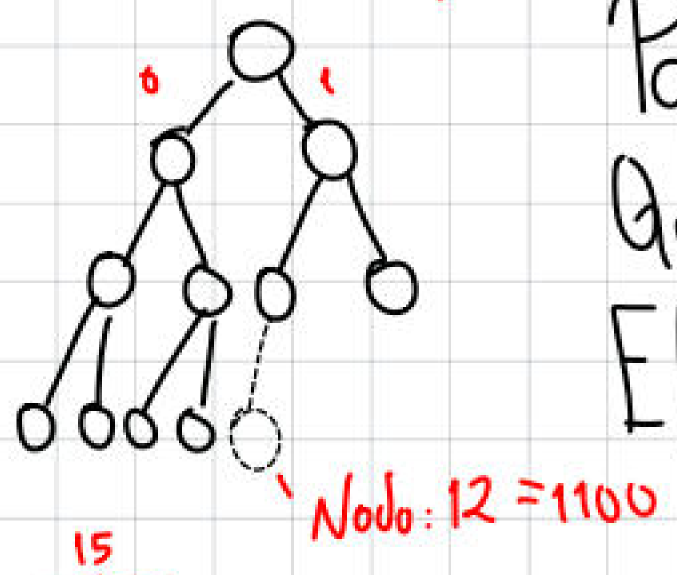
Arbol binario completo.



Para movernos: 0 izq, 1 der.

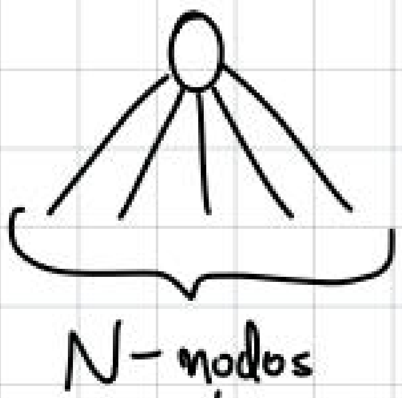
Queremos llegar al 12 = 1100, 11 => der, 110 => izq, 1100 => izq

El binario codifica nuestros movimientos, iniciando en el primer bit que nunca es cero.

Así: $\overbrace{1111}^{15}$: Der, der, der y allí se coloca el nodo.

$\underbrace{10001}_{17}$: Izq, izq, izq, der y se coloca el nodo.

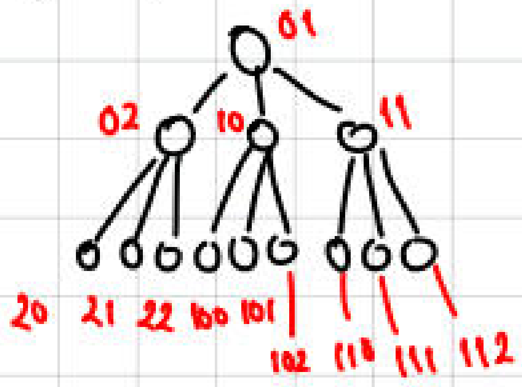
Árbol n-ario:



struct_nodo_ *minodo[N], otra opción: struct_nodo_ **minodo.

Podemos usar base N cada número para representar cada Nodo.

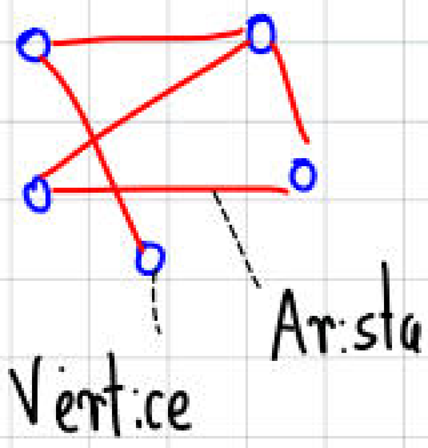
Ejemplo: base 3:



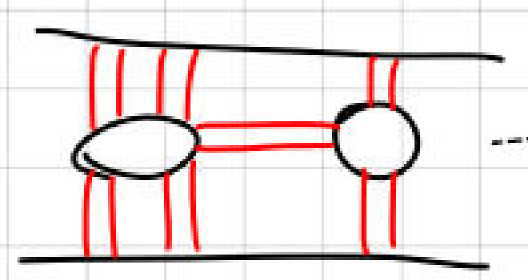
02: izq inicial

0: izq, 1: cent, 2 der al pasar nivel 1.

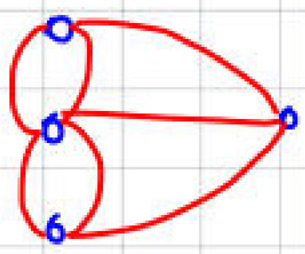
Grafos.



Puentes de Köninsberg.

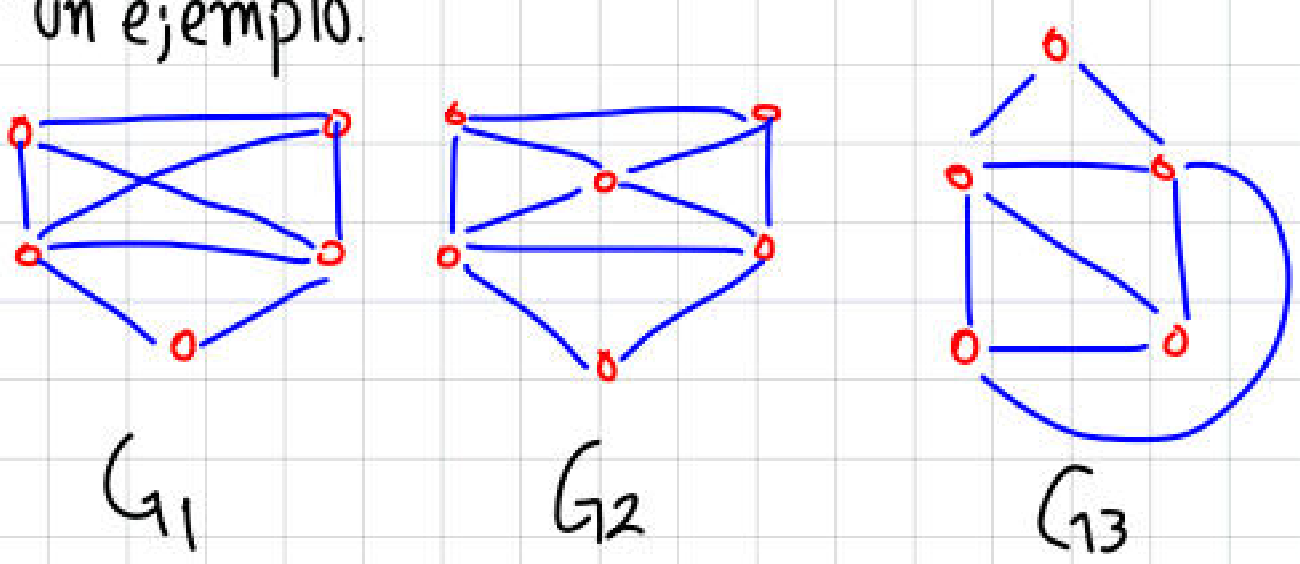


Grafo:



Grafos sin pesos no dirigidos.

Sea V el conjunto de vértices, $A = \{\{a,b\} \in P(V) \mid a \neq b\}$ es el conjunto de aristas, este es un ejemplo.



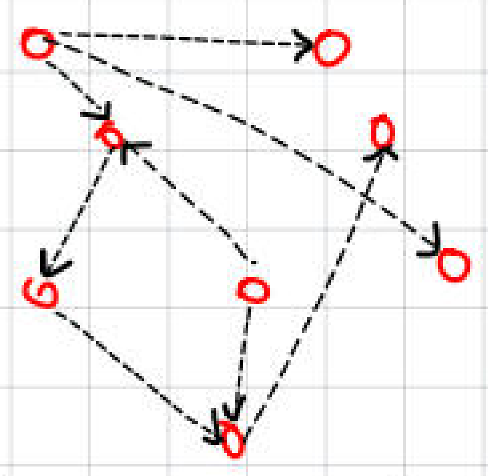
Se cumple que $G_1 = G_2 = G_3$. No, G_2 tiene 6 vértices, pero se cumple que $G_1 = G_3$ tienen los mismos vértices y las aristas unen de esta forma 2 vértices.

No dirigidos: Es lo mismo ir de Va a Vb, solo se va en una dirección.

Sin pesos: las aristas no tienen prioridad, esta prioridad es el peso.

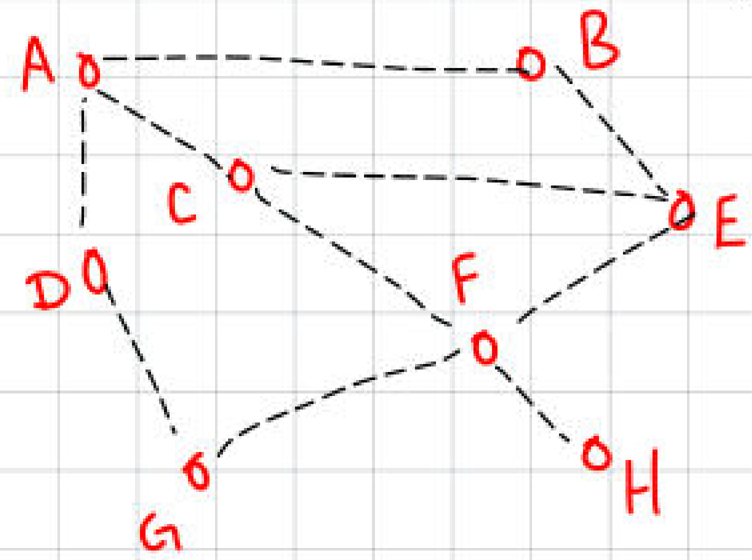
Grafos sin pesos dirigidos.

V conjunto de vértices, $A = \{(a,b) \in V \times V \mid a \neq b\}$



Ejemplo grafo dirigido.

En un grafo, los nodos o vértices no tienen información del peso. Un grafo lo representaremos como una matriz. Ejemplo:



HACIA ↓

DE →	A	B	C	D	E	F	G	H
A	•	1	1	1	0	0	0	0
B	1	•	0	0	1	0	0	0
C	1	0	•	0	1	1	0	0
D	1	0	0	•	0	0	1	0
E	0	1	1	0	•	1	0	0
F	0	0	1	0	1	•	1	1
G	0	0	0	1	0	1	•	0
H	0	0	0	0	0	1	0	•

Matriz de adyacencia.

Si el grafo es dirigido, la matriz resultante puede no ser simétrica. La longitud de camino máxima entre 2 vértices, es de n-1 aristas, n: número de vértices.

Veamos esta matriz (camino de al menos, 2 aristas).

	A	B	C	D	E	F	G	H
A	•	1	1	1	2	2	2	0
B	1	•	2	2	1	2	0	0
C	1	2	•	2	1	1	2	2
D	1	2	2	•	0	2	1	0
E	2	1	1	0	•	1	2	2
F	2	2	1	2	1	•	1	1
G	2	0	2	1	2	1	•	2
H	0	0	2	0	2	1	2	•

Aquí se cuentan el número de pasos para ir de un vértice a otro.

Nos muestra si existe un camino de, máximo 2 pasos de un nodo a otro.

Implementar estas matrices, y poner el nodo del que viene (enrutar)

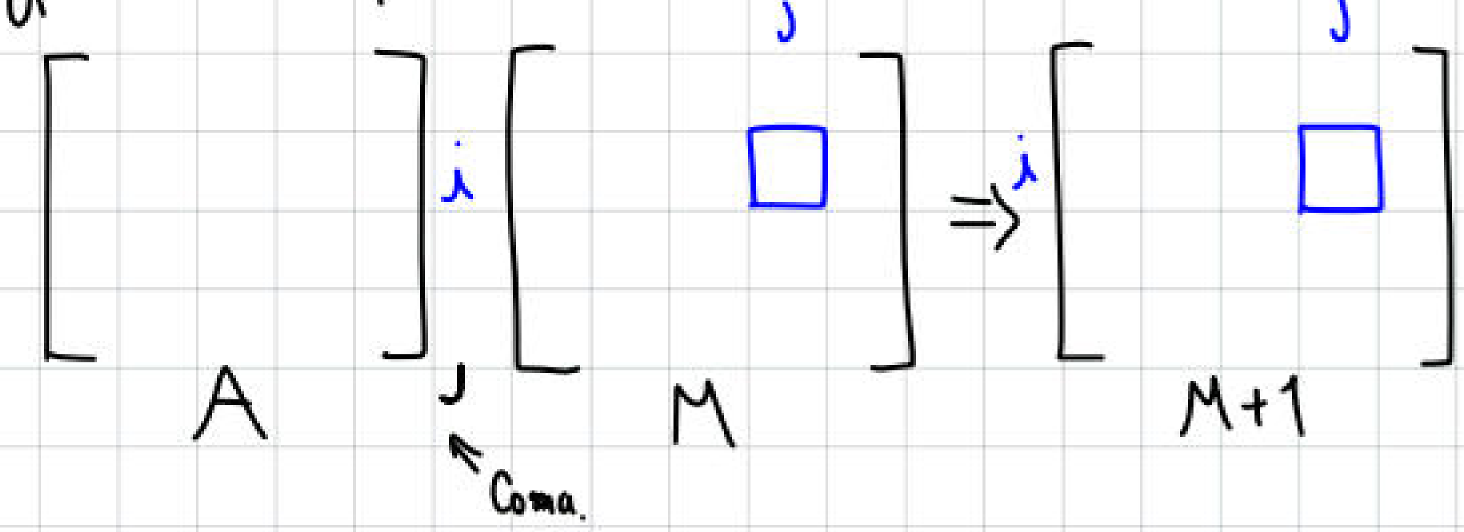
25/04/22

Para el código: nombre nodo (entero), peso w, info pasos.

Entradas:

26/04/22

Programa a implementar: con la matriz de adyacencia y otra, llamada: M y M+1:



En M+1 se plasma si hay un camino de el nodo i al j, sin pesos da las aristas y con pesos, el de menor peso.

Luego, debemos recordar el camino para ir de un nodo a otro. Queremos ver si se produce cambio de la matriz M a la matriz M-1. Nos fijamos en $K \in N_n$, $K \neq i, j$ y vemos si existe un camino de i a K y de K a j, en caso de que no haya uno directo de

i a j. Se recorre la K en los valores, y en caso de que se encuentre la K, vemos si: $V_{ij} > V_{iK} + V_{Kj}$, si sucede, el valor se cambia a $V_{iK} + V_{Kj}$ y en caso contrario, se deja igual. Si no existe el camino, se pone el valor $V_{iK} + V_{Kj}$.
M es la matriz de M-caminos. Para $M=1$, $M=A$, $M=2=1+1=A+1$. Luego, la matriz siguiente es con 2 o menos caminos.