Universidad de Valparaíso
CHILE
Escuela de Ingeniería Civil Informática
Facultad de Ingeniería

Estructuras de datos

Capítulo III: Grafos

Eduardo Godoy
eduardo.gl@gmail.com

2017-II

Index

Definición general

- Un grafo es un par $G = (V, E)$ donde V es un conjunto de nodos y E un conjunto de aristas.
- En general se clasifican en dirigidos y no dirigidos.

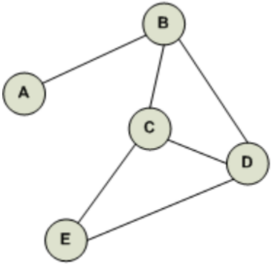


Fig 1. Grafo no dirigido

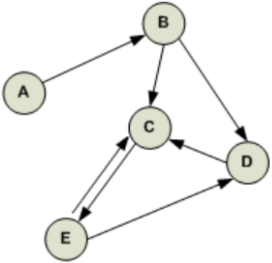
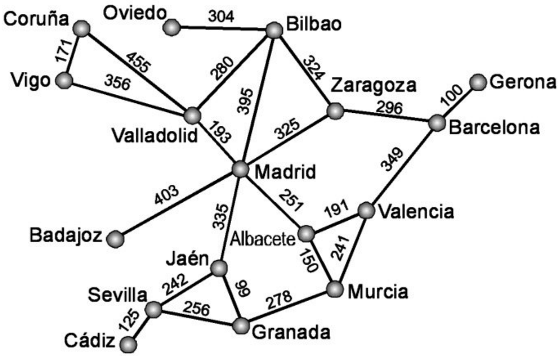


Fig 2. Grafo dirigido

Aplicación


La Coruña
Asturias
Santander
Bilbao
San Sebastián
Pamplona
Vigo
Valladolid
Zaragoza
BARCELONA
Tarragona
MADRID
Palma de Mallorca
Valencia
Alicante
Murcia
Córdoba
Sevilla
Granada
Cádiz
Málaga
ÁREAS METROPOLITANAS
más de 3.000.000
de 1.000.000 a 3.000.000
de 500.000 a 1.000.000
de 300.000 a 500.000
Santa Cruz de Tenerife
Las Palmas
Fuente: INE, Proyecto AUDES
Datos a 01-01-2005

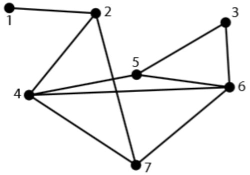
Aplicación


Coruña
Oviedo
304
Bilbao
171
455
280
324
Vigo
356
395
Zaragoza
Gerona
100
296
Valladolid
193
325
Barcelona
349
Madrid
403
251
191
Valencia
335
Badajoz
Albacete
241
150
Jaén
242
99
Sevilla
278
Murcia
256
125
Granada
Cádiz

Representación

- La forma común de representar un grafo en memoria es mediante un arreglo bidimencional llamado Matriz de Adyacencia.
- Matríz de Adyacencia: Es una matriz nxn donde n es la cantidad de nodos que forma al grafo y la posición i,j representa si hay o no arista entre los nodos siento este valor 1 o 0 si no es dirigido de un numero natural positivo si es un grafo dirigido.

Ej:



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

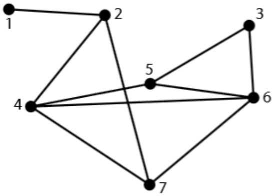
Representación - Matriz de Adyacencia

- Esta es una de las representaciones más utilizadas. Si bien el ejemplo es para un grafo no dirigido, también se puede utilizar la misma estructura para grafos dirigidos y grafos con pesos.
- Ventajas:
  - Permite saber si existe o no arista entre dos nodos cualesquiera en O(1).
  - Es muy fácil de implementar, matrizAdy[i][j] guarda toda la información sobre la arista.
- Desventajas:
  - La complejidad espacial: se necesitan n^2 casillas para representar un grafo de n nodos.

Representación - Lista de Adyacencia

- La lista de adyacencia es un vector de vectores de enteros, que en el i-ésimo vector tiene el número j si hay una arista entre los nodos i y j.
- También se le conoce como lista de vecinos ya que para cada nodo se guarda la lista de nodos para los que existe una arista que los conecta (o sea, los vecinos)

Ej:



$L_1 : 2$
$L_2 : 1 \rightarrow 4 \rightarrow 7$
$L_3 : 5 \rightarrow 6$
$L_4 : 2 \rightarrow 5 \rightarrow 6 \rightarrow 7$
$L_5 : 3 \rightarrow 4 \rightarrow 6$
$L_6 : 3 \rightarrow 4 \rightarrow 5 \rightarrow 7$
$L_7 : 2 \rightarrow 4 \rightarrow 6$

Representación - Lista de Adyacencia

- Nuevamente, con la misma idea también se pueden modelar grafos dirigidos y con pesos.
- La complejidad espacial de esta representación será posiblemente mucho menor. ¿Cuánta memoria necesitaremos para un grafo de n nodos y m aristas? O(m+n)

Recorrido - Camino más Corto

- Dado un grafo G con pesos en las aristas, el problema de camino mínimo entre dos nodos u y v consiste en encontrar un camino entre esos nodos cuyo peso sea menor o igual que el peso de cualquier otro camino entre u y v

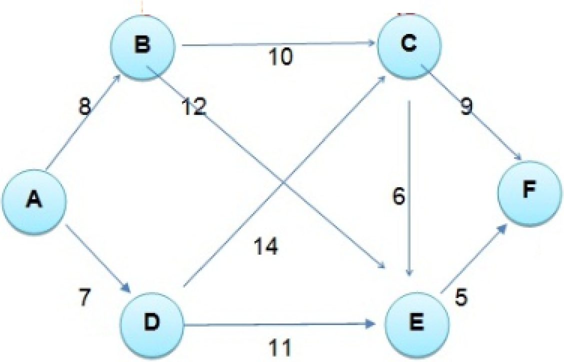
Recorrido - Algoritmo de Dijkstra

- Este algoritmo fue creado por uno de los padres de la computación, Edger W. Dijkstra, en 1956. Sirve para cualquier grafo con pesos (dirigido o no) siempre y cuando sus pesos no sean negativos.

Recorrido - Algoritmo de Dijkstra

- El algoritmo calcula las distancias mínimas desde un nodo inicial a todos los demás. Para hacerlo, en cada paso se toma el nodo más cercano al inicial que aún no fue visitado (le diremos v). Este nodo tiene calculada la menor distancia al nodo inicial (¿por qué?).
- Luego, recalculamos todas los caminos mínimos, teniendo en cuenta a v como camino intermedio.
- Así, en cada paso tendremos un subconjunto de nodos que ya tienen calculada su mínima distancia y los demás tienen calculada su mínima distancia si solo puedo usar los nodos del conjunto como nodos intermedios.
- Con cada iteración agregaremos un nodo más a nuestro conjunto, hasta resolver el problema en su totalidad.

Algoritmo de Dijkstra - Ejemplo



Algoritmo de Dijkstra - Ejemplo

Iteración	nodo fijo		A	B	C	D	E	F
1	A	Anterior	-	A	inf	A	inf	inf
		Distancia	0	8	-	7	-	-
2	D	Anterior	-	A	D	A	D	inf
		Distancia	0	8	21	7	18	-
3	B	Anterior	-	A	B	A	D	inf
		Distancia	0	8	18	7	18	-
3	C	Anterior	-	A	B	A	D	C
		Distancia	0	8	18	7	18	27
3	E	Anterior	-	A	B	A	D	E
		Distancia	0	8	18	7	18	23

- Respuesta: A $\rightarrow D \rightarrow E \rightarrow F$