

TALLER PROGCOMP: TRACK GRAFOS

INTRODUCCIÓN A GRAFOS

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

September 1, 2024

Part I

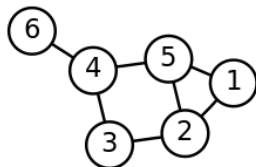
GRAFOS

GRAFOS DEFINICIÓN

Definición

Estructura que consiste en un conjunto de *objetos* y relaciones entre esos objetos.

- ▶ los objetos son representados utilizando *nodos*
- ▶ las relaciones entre ellos son representados utilizando *aristas*

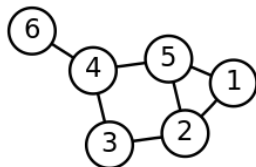


GRAFOS DEFINICIÓN

Definición

Estructura que consiste en un conjunto de *objetos* y relaciones entre esos objetos.

- ▶ los objetos son representados utilizando *nodos*
- ▶ las relaciones entre ellos son representados utilizando *aristas*



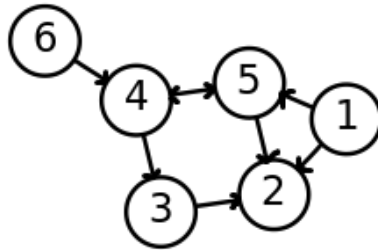
Casos de uso

- ▶ ciudades, conexiones entre ciudades
- ▶ posiciones en una grilla
- ▶ personas y relaciones entre personas
- ▶ artículos científicos
- ▶ bases de datos
- ▶ y mucho más!

OTROS TIPOS DE GRAFOS - GRAFO DIRIGIDO

Diferencia con el Grafo (Grafo no dirigido)

Grafo en el cuál si existe la relación de a con b , no necesariamente existe la relación b con a .

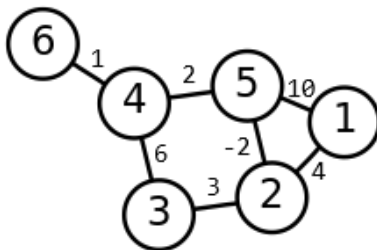


OTROS TIPOS DE GRAFOS - GRAFO CON PESO

Diferencia con el Grafo (Grafo no dirigido)

Grafo en el cuál, además de existir una relación, se dirá que la relación tiene un *peso*.

OJO: en un grafo dirigido con pesos, de a a b y de b a a pueden tener pesos distintos.



REPRESENTACIONES DE GRAFOS

¿Cómo nosotros podemos representar un grafo en un problema?

REPRESENTACIONES DE GRAFOS

¿Cómo nosotros podemos representar un grafo en un problema?

Historicamente existen dos formas:

- ▶ Matriz de adyacencia
- ▶ Lista de adyacencia

Part II

MATRIZ DE ADYACENCIA

MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

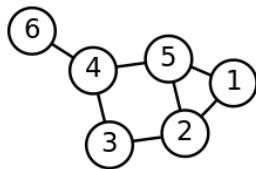
MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

Grafo no dirigido

En la posición i, j de la matriz se verá un 1 si existe una arista entre i y j , 0 en otro caso.



i/j	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

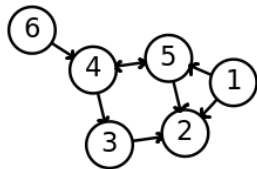
```
1  int main() {
2      int n, a; // n siendo cantidad de nodos
3              // a siendo cantidad de aristas
4      cin >> n >> a;
5
6      // ojo que los vectores en C++ parten de 0
7      vector< vector< int > > matriz(n, vector< int >(n, 0));
8      for(int i = 0; i < a; i++) {
9          int x, y;
10         cin >> x >> y;
11         // si los nodos parten de 1, recuerden restar 1
12
13         matriz[x][y] = 1;
14         matriz[y][x] = 1;
15     }
16
17     // hacer lo que quieran
18     return 0;
19 }
```

MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

Grafo dirigido



i/j	1	2	3	4	5	6
1	0	1	0	0	1	0
2	0	0	0	0	0	0
3	0	1	0	0	0	0
4	0	0	1	0	1	0
5	0	1	0	1	0	0
6	0	0	0	1	0	0

MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

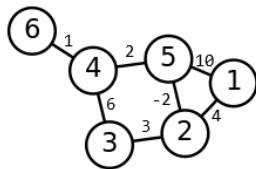
```
1  int main() {
2      int n, a; // n siendo cantidad de nodos
3              // a siendo cantidad de aristas
4      cin >> n >> a;
5
6      // ojo que los vectores en C++ parten de 0
7      vector< vector< int > > matriz(n, vector< int >(n, 0));
8      for(int i = 0; i < a; i++) {
9          int x, y;
10         cin >> x >> y;
11         // si los nodos parten de 1, recuerden restar 1
12
13         matriz[x][y] = 1;
14     }
15
16     // hacer lo que quieran
17     return 0;
18 }
```

MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

Grafo con peso



i/j	1	2	3	4	5	6
1	0	4	0	0	10	0
2	4	0	3	0	-2	0
3	0	3	0	6	0	0
4	0	0	6	0	2	1
5	10	-2	0	2	0	0
6	0	0	0	1	0	0

MATRIZ DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una matriz de tamaño $n \times n$. La posición i, j de la matriz representa la arista entre el nodo i y el nodo j .

```
1  int main() {
2      int n, a; // n siendo cantidad de nodos
3              // a siendo cantidad de aristas
4      cin >> n >> a;
5
6      // ojo que los vectores en C++ parten de 0
7      vector< vector< int > > matriz(n, vector< int >(n, 0));
8      for(int i = 0; i < a; i++) {
9          int x, y, w;
10         cin >> x >> y >> w;
11         // si los nodos parten de 1, recuerden restar 1
12
13         matriz[x][y] = w;
14         matriz[y][x] = w; // si es dirigido con peso borrar esta linea
15     }
16
17     // hacer lo que quieran
18     return 0;
19 }
```


MATRIZ DE ADYACENCIA

A tener en cuenta:

- ▶ Espacio utilizado es n^2
- ▶ Si quiero determinar si la arista entre x y y existe es $O(1)$
- ▶ Determinar todos los vecinos de un nodo es $O(n)$

Part III

LISTA DE ADYACENCIA

LISTA DE ADYACENCIA

Definición

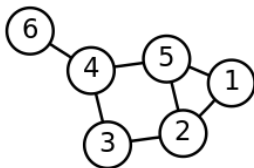
Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenamos un arreglo que contiene los vecinos del nodo i .

LISTA DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenas un arreglo que contiene los vecinos del nodo i .

Grafo no dirigido



i	
1	→ 2 → 5
2	→ 1 → 5 → 3
3	→ 2 → 4
4	→ 3 → 5 → 6
5	→ 1 → 2 → 4
6	→ 4

LISTA DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenas un arreglo que contiene los vecinos del nodo i .

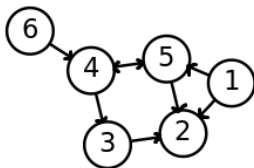
```
1  int main() {
2      int n, a; // n siendo cantidad de nodos
3              // a siendo cantidad de aristas
4      cin >> n >> a;
5
6      // ojo que los vectores en C++ parten de 0
7      vector< vector< int > > lista(n);
8      for(int i = 0; i < a; i++) {
9          int x, y;
10         cin >> x >> y;
11         // si los nodos parten de 1, recuerden restar 1
12
13         lista[x].push_back(y);
14         lista[y].push_back(x);
15     }
16
17     // hacer lo que quieran
18     return 0;
19 }
```

LISTA DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenas un arreglo que contiene los vecinos del nodo i .

Grafo dirigido



i	
1	→ 2 → 5
2	
3	→ 2
4	→ 3 → 5
5	→ 2 → 4
6	→ 4

LISTA DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenas un arreglo que contiene los vecinos del nodo i .

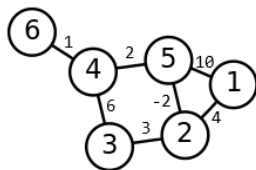
```
1  int main() {
2      int n, a; // n siendo cantidad de nodos
3              // a siendo cantidad de aristas
4      cin >> n >> a;
5
6      // ojo que los vectores en C++ parten de 0
7      vector< vector< int > > lista(n);
8      for(int i = 0; i < a; i++) {
9          int x, y;
10         cin >> x >> y;
11         // si los nodos parten de 1, recuerden restar 1
12
13         lista[x].push_back(y);
14     }
15
16     // hacer lo que quieran
17     return 0;
18 }
```

LISTA DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenas un arreglo que contiene los vecinos del nodo i .

Grafo con peso



i	
1	→ 2,4 → 5,10
2	→ 1,4 → 5,-2 → 3,3
3	→ 2,3 → 4,6
4	→ 3,6 → 5,2 → 6,1
5	→ 1,10 → 2,-2 → 4,2
6	→ 4,1

LISTA DE ADYACENCIA

Definición

Si el grafo posee n nodos, puede ser representado utilizando una lista de los vecinos de cada nodo. En términos de implementación, utilizando un arreglo, donde en la posición i almacenamos un arreglo que contiene los vecinos del nodo i .

```
1  int main() {
2      int n, a; // n siendo cantidad de nodos
3              // a siendo cantidad de aristas
4      cin >> n >> a;
5
6      // ojo que los vectores en C++ parten de 0
7      vector< vector< pair< int, int > > > lista(n);
8      for(int i = 0; i < a; i++) {
9          int x, y, w;
10         cin >> x >> y >> w;
11         // si los nodos parten de 1, recuerden restar 1
12
13         lista[x].push_back({y, w});
14         lista[y].push_back({x, w}); // si es dirigido con peso borrar esta linea
15     }
16
17     // hacer lo que quieran
18     return 0;
19 }
```

LISTA DE ADYACENCIA

A tener en cuenta:

- ▶ Espacio utilizado es $n + a$
- ▶ Si quiero determinar si la arista entre x e y existe es $O(n)$
- ▶ Determinar todos los vecinos de un nodo es $O(1)$

REFERENCES I