TALLER PROGCOMP: TRACK GRAFOS

ALGORITMO DE FLOYD-WARSHALL

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

September 29, 2024

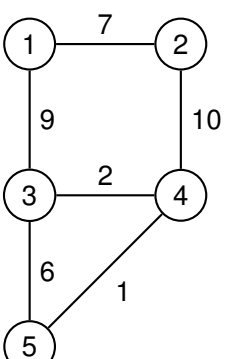
Part I

ALGORITMO DE DIJSKTRA

Algoritmo de Dijsktra

Problemas

- Camino más corto de un solo nodo a todo el resto.
- Si hay aristas negativas, muere.

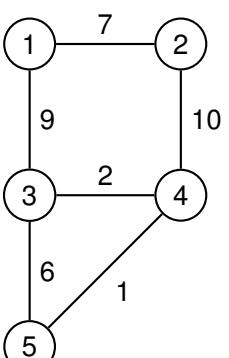


Part II

ALGORITMO DE FLOYD-WARSHALL

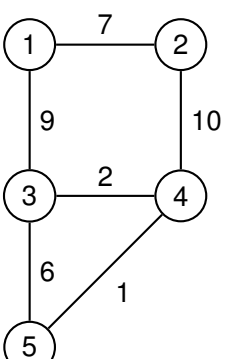
Ventajas

- Camino más corto de todos los nodos a todos los nodos
- Si hay aristas negativas, ¡¡no muere!!.



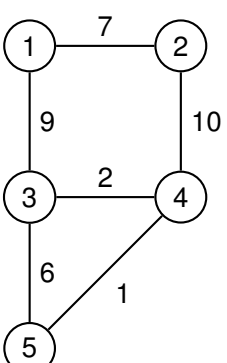
¿Cómo funciona?

- Inicializa matriz de distancias con todos los pesos directos.
- Intenta usar todos los nodos como nodo intermediario entre dos nodos.



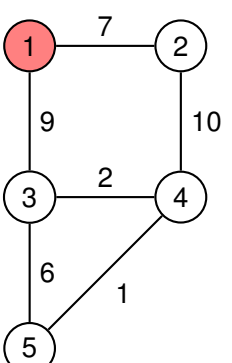
Algoritmo de Floyd-Warshall

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	∞	10	∞
3	9	∞	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



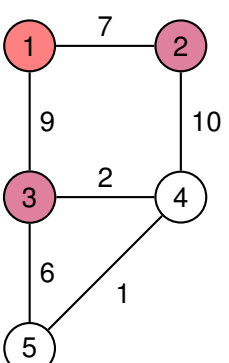
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	∞	10	∞
3	9	∞	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



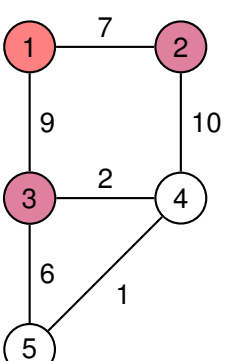
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	∞	10	∞
3	9	∞	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



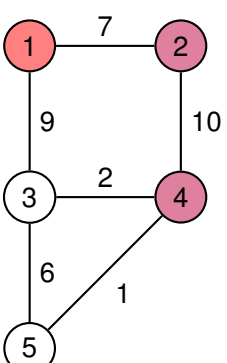
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	9 + 7 = 16	10	∞
3	9	9 + 7 = 16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



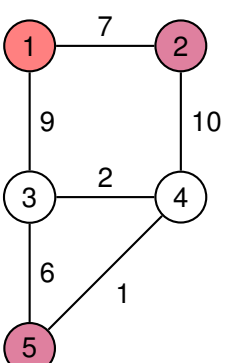
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



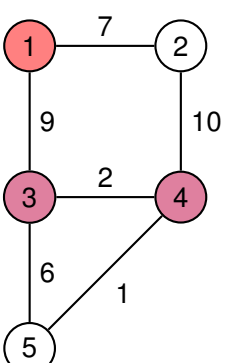
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



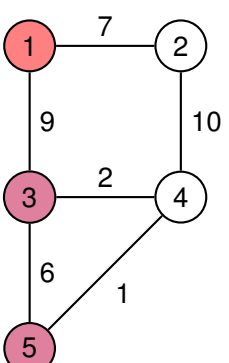
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



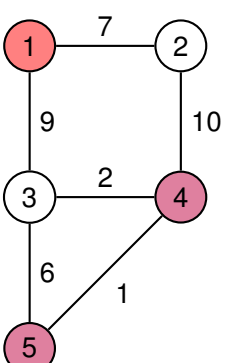
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



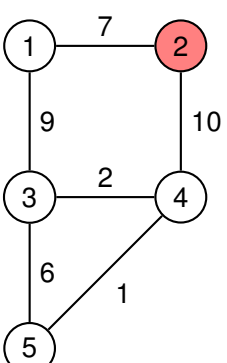
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



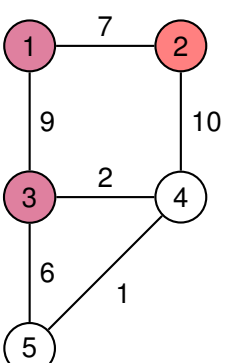
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



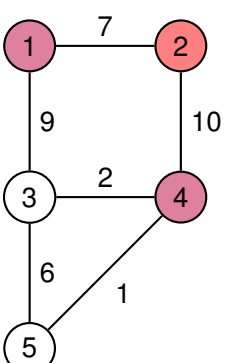
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



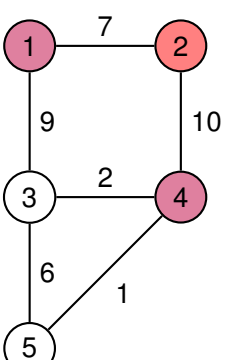
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	∞	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	∞	10	2	0	1
5	∞	∞	6	1	0



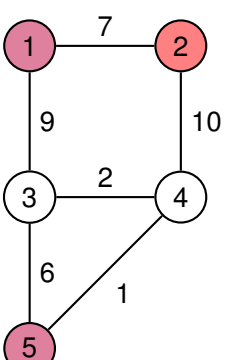
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	7 + 10 = 17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	7 + 10 = 17	10	2	0	1
5	∞	∞	6	1	0



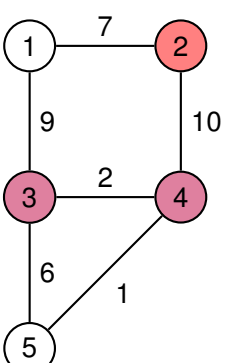
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



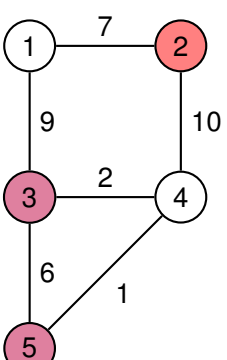
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



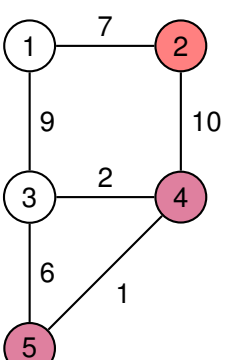
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



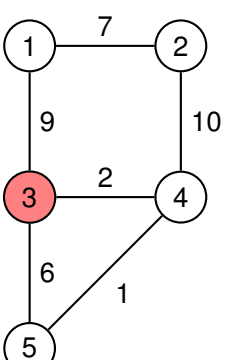
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



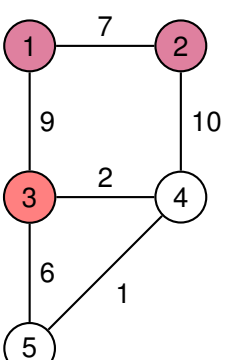
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



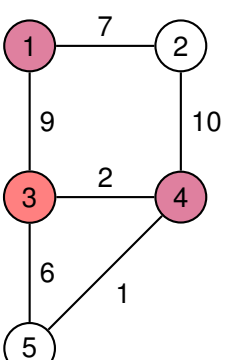
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



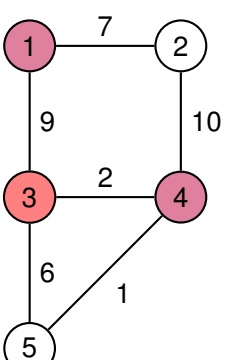
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	17	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	17	10	2	0	1
5	∞	∞	6	1	0



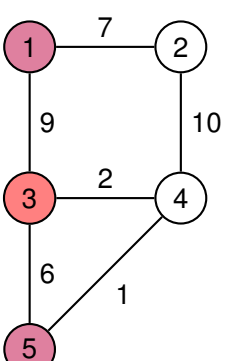
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	9 + 2 = 11	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	9 + 2 = 11	10	2	0	1
5	∞	∞	6	1	0



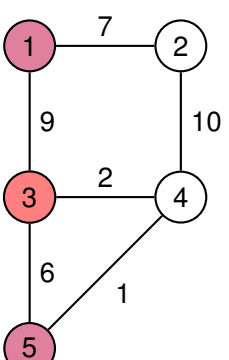
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	∞
2	7	0	16	10	∞
3	9	16	0	2	6
4	11	10	2	0	1
5	∞	∞	6	1	0



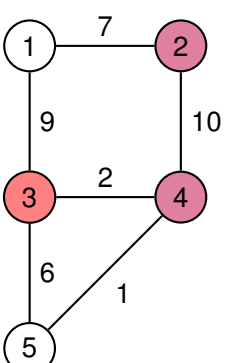
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	9 + 6 = 15
2	7	0	16	10	∞
3	9	16	0	2	6
4	11	10	2	0	1
5	9 + 6 = 15	∞	6	1	0



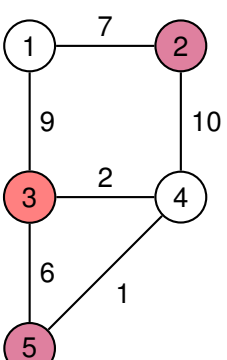
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	∞
3	9	16	0	2	6
4	11	10	2	0	1
5	15	∞	6	1	0



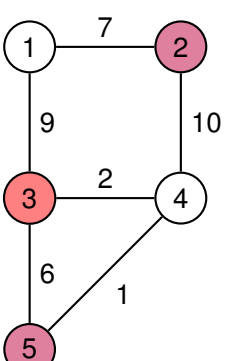
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	∞
3	9	16	0	2	6
4	11	10	2	0	1
5	15	∞	6	1	0



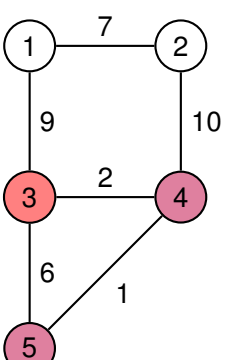
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	16 + 6 = 22
3	9	16	0	2	6
4	11	10	2	0	1
5	15	16 + 6 = 22	6	1	0



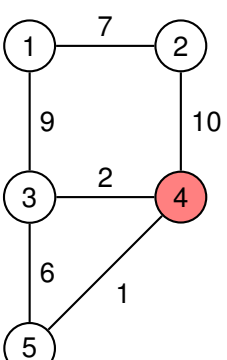
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	15	22	6	1	0



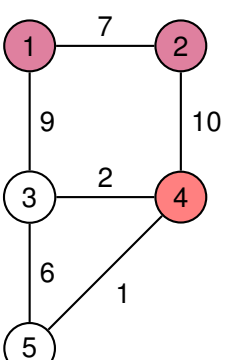
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	15	22	6	1	0



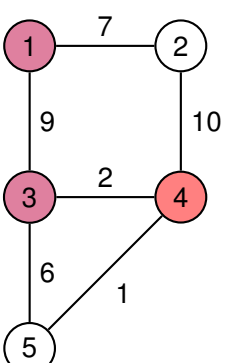
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	15	22	6	1	0



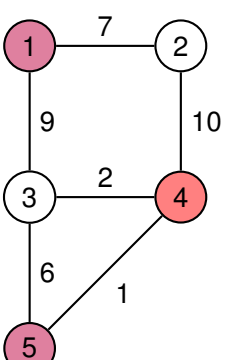
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	15	22	6	1	0



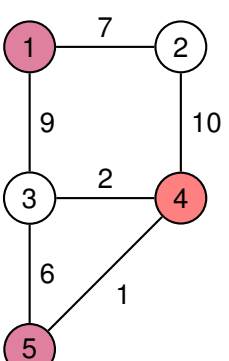
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	15
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	15	22	6	1	0



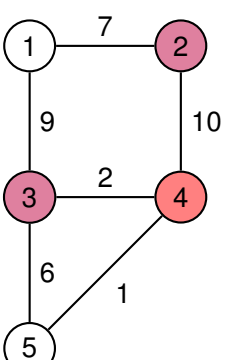
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	11 + 1 = 12
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	11 + 1 = 12	22	6	1	0



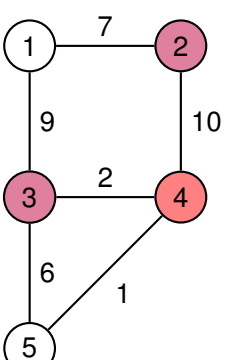
Algoritmo de Floyd-Warshall

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	16	10	22
3	9	16	0	2	6
4	11	10	2	0	1
5	12	22	6	1	0



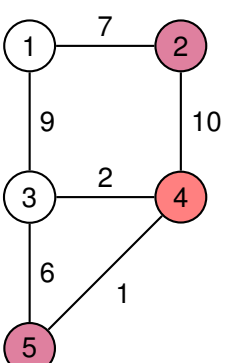
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	10 + 2 = 12	10	22
3	9	10 + 2 = 12	0	2	6
4	11	10	2	0	1
5	12	22	6	1	0



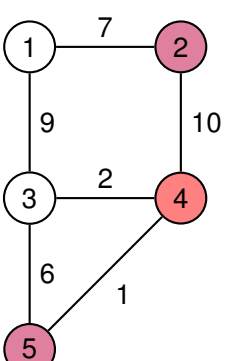
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	22
3	9	12	0	2	6
4	11	10	2	0	1
5	12	22	6	1	0



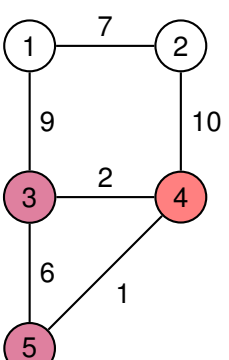
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	10 + 1 = 11
3	9	12	0	2	6
4	11	10	2	0	1
5	12	10 + 1 = 11	6	1	0



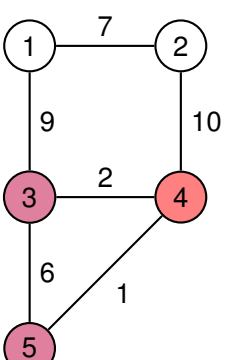
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	6
4	11	10	2	0	1
5	12	11	6	1	0



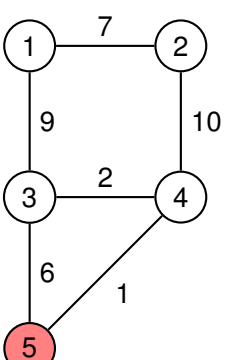
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	2 + 1 = 3
4	11	10	2	0	1
5	12	11	2 + 1 = 3	1	0



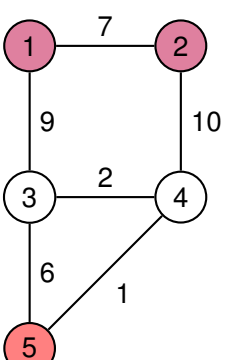
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



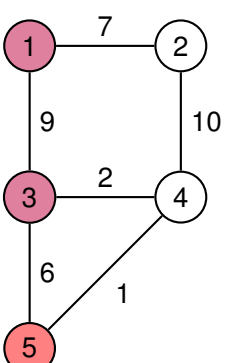
Algoritmo de Floyd-Warshall

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



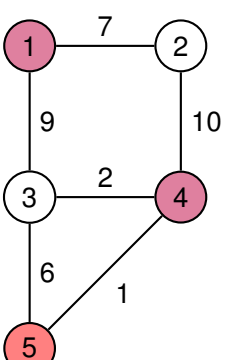
Algoritmo de Floyd-Warshall

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



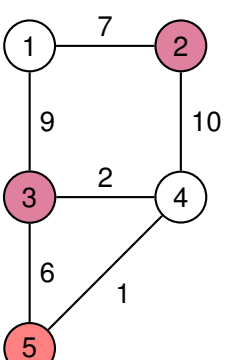
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



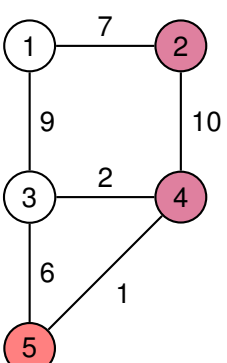
Algoritmo de Floyd-Warshall

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



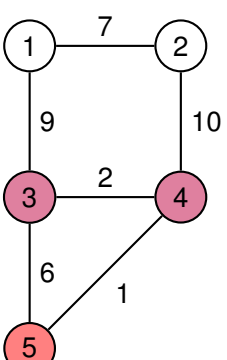
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



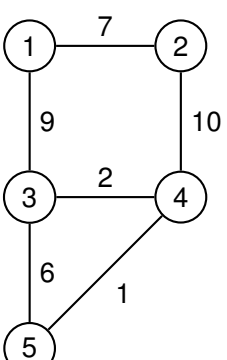
ALGORITMO DE FLOYD-WARSHALL

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



Algoritmo de Floyd-Warshall

	1	2	3	4	5
1	0	7	9	11	12
2	7	0	12	10	11
3	9	12	0	2	3
4	11	10	2	0	1
5	12	11	3	1	0



Algoritmo de Floyd-Warshall

Detalles

- Complejidad algoritmica $O(n^3)$.

Algoritmo de Floyd-Warshall

Detalles

- Complejidad algoritmica $O(n^3)$.
- Complejidad espacial $O(n^2)$.

Detalles

- Complejidad algoritmica $O(n^3)$.
- Complejidad espacial $O(n^2)$.
- **OJO**: en caso de grafo dirigido $a \to b$ no es lo mismo que $b \to a$.

Código

```
int n;
// matriz de adyacencia
vector< vector< int > > matriz;
// matriz de distancias
vector< vector< int > > d;

void floyd-warshall() {
  for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
      d[i][j] = matriz[i][j];
    }
  }

  for(int k = 0; k < n; k++) {
    for(int i = 0; i < n; i++) {
      for(int j = 0; j < n; j++) {
        d[i][j] = min(d[i][j], d[i][k] + d[k][j]);
      }
    }
  }
}
```

Código main

```
int inf = 1000000000;
int main() {
  cin >> n;
  int e;
  cin >> e;
  matriz.resize(n, vector< int >(n, inf));

  for(int i = 0; i < n; i++) matriz[i][i] = 0;

  for(int i = 0; i < e; i++) {
    int a, b, w;
    cin >> a >> b >> w;
    matriz[a][b] = w; // asumiendo que es dirigido
    // matriz[b][a] = w;
  }

  d.resize(n, vector< int >(n, inf));
  floyd-warshall();

  // hacer lo que quieras
}
```

REFERENCES I