

## Tarea 3: Medición experimental de la complejidad asintótica

Gabriela Sánchez Y.

### Introducción

En esta práctica se modifica la clase **Grafo** creada en la práctica anterior [3] con el objetivo de incluir el algoritmo de Floyd-Warshall y el algoritmo de Ford-Fulkerson [2] con la finalidad de estudiar la complejidad asintótica de ambos.

Primero se presentan algunos ejemplos para ver cómo funciona cada uno de estos algoritmos. Después se crean grafos de distintos tamaños y se revisan los tiempos de ejecución para el análisis de la complejidad asintótica.

### Algoritmo de Floyd-Warshall

El algoritmo de Floyd-Warshall determina el camino más corto entre cualquier par de nodos, en un tiempo de ejecución de  $O(n^3)$ , donde  $n$  es el número de nodos en el grafo [1]. El algoritmo funciona tanto para grafos dirigidos como no dirigidos.

En la figura 1 vemos un ejemplo de un grafo con tres nodos, en este caso la ponderación de la arista representa una distancia, éstas se muestran en el cuadro 1. Los resultados obtenidos se presentan en el cuadro 2 y se interpretan de la siguiente manera: por ejemplo, la distancia mínima del nodo  $a$  al nodo  $c$  es la intersección del primer renglón con la tercera columna, en este caso tres.

Cuadro 1: Cuadro de distancias correspondientes al grafo ejemplo.

	a	b	c
a	-	1	2
b	1	-	1
c	4	2	-

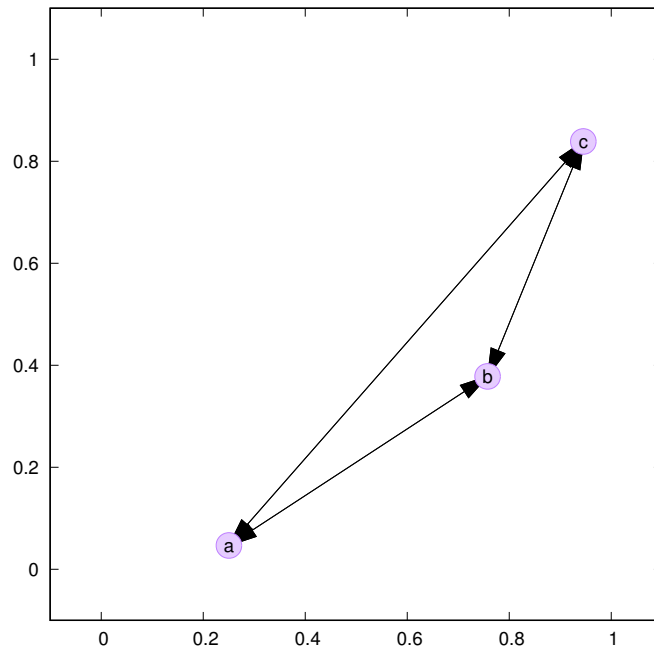


Figura 1: Grafo ejemplo para el algoritmo de Floyd-Warshall.

Cuadro 2: Resultados ejemplo Floyd-Warshall.

	a	b	c
a	0	1	2
b	1	0	1
c	3	2	0

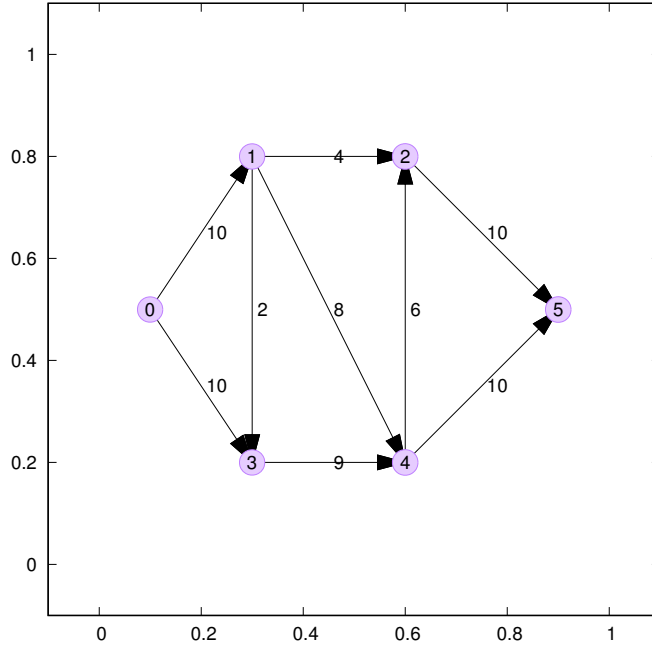


Figura 2: Grafo ejemplo para el algoritmo de Ford-Fulkerson.

## Algoritmo de Ford-Fulkerson

El algoritmo de Ford-Fulkerson determina el flujo máximo entre el nodo fuente  $s$  y el nodo destino  $t$ . Cuando las capacidades son reales, el tiempo de ejecución del algoritmo es  $O(nmU)$ , donde  $n$  es el número de nodos en el grafo,  $m$  el número de aristas y  $U$  es una cota para todas las capacidades [1].

Como ejemplo, se resolvió el problema de encontrar el flujo máximo en grafo en la figura 2, partiendo del nodo cero al nodo cinco. El resultado en este caso es  $f = 19$ .

## Complejidad computacional

Para determinar la complejidad computacional del algoritmo de Floyd-Warshall se realizó un experimento en el que se varía el número de nodos en el grafo  $n = 10, 20, \dots, 100$ , realizando diez réplicas para cada prueba guardando el tiempo de ejecución en cada caso, los resultados se muestran en la figura 3.

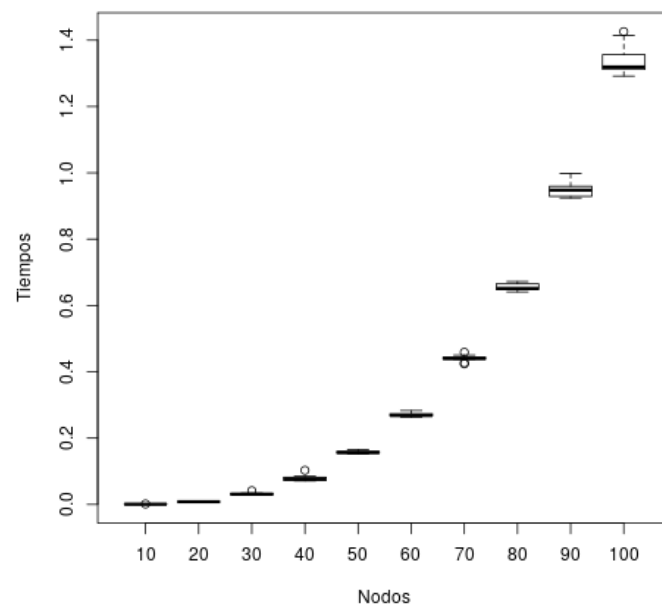


Figura 3: Tiempos de ejecución del algoritmo de Floyd-Warshall para grafos dirigidos.

## Referencias

- [1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- [2] Elisa Schaeffer. Matemáticas discretas, curso en línea. <https://elisa.dyndns-web.com/teaching/mat/discretas/md.html>.
- [3] Gabriel Sánchez Y. Visualización de grafos simples, ponderados y dirigidos. [https://github.com/Saphira3000/Flujo\\_Redes/tree/master/tarea2](https://github.com/Saphira3000/Flujo_Redes/tree/master/tarea2).