PRÁCTICA 4 ALGORITMIA

Nahiara Sánchez García

Índice

Objetivo

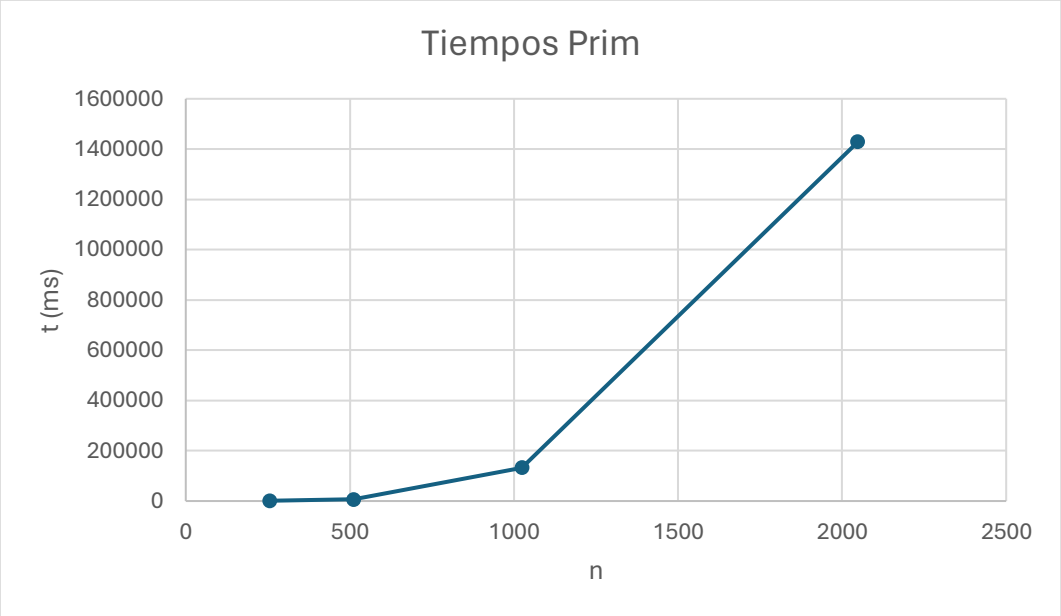
El objetivo de esta práctica es implementar uno de los ejemplos más claros de algoritmos devoradores: el **algoritmo de Prim**. A partir de un grafo de n nodos, va eligiendo la arista mínima hasta seleccionar n-1 aristas. De esta manera, se obtiene el camino de coste mínimo para recorrer un grafo.

Ejercicio 1

1. Tras implementar el algoritmo de Prim que después de leer un nombre de un fichero, calcula y escribe por pantalla la solución óptima, se obtiene un algoritmo de complejidad $O(n^3)$. La complejidad no es la mejor que se podría obtener, pero es la que se pudo implementar. Tiene esta complejidad debido a que presenta tres bucles anidados: un bucle while y dos for.
   A continuación, se implementa la clase PrimTiempos con la finalidad de que vaya creando grafos aleatorios de diversos tamaños y calculando el tiempo que tarda el algoritmo en resolver el problema.
   La tabla de tiempos obtenida es la siguiente:

n	Tiempo Prim (ms)
256	1188,75694
512	6793,7386
1024	132326,25
2048	1428442,74

Y la gráfica será:



Como se puede ver ya en la gráfica, se verifica que la complejidad del algoritmo es $O(n^3)$. Sin embargo, se va a realizar una comprobación a continuación.
n1 = 256
t1 = 1188,75694
n2 = 512
t2 = ?

$t2 = (n2)^3 / (n1)^3 * t1 = 9510{,}055542$ ms

Como se ve, no es el mismo valor que se obtiene en la tabla anterior, que para un n = 512 tarda 6793,7389 ms, pero en el cálculo realizado el tiempo obtenido es superior.

Véase el siguiente caso:

n1 = 256

t1 = 1188,75694

n2 = 1024

t2 = ?

$t2 = (n2)^3 / (n1)^3 * t1 = 76080{,}44434$ ms

En este caso el tiempo obtenido es menor que el de la tabla.

Es decir, no cuadra la complejidad calculada, $O(n^3)$, con la complejidad de los datos de la tabla, que sería una complejidad entre $O(n^3)$ y superior.

Esto podría ser porque el algoritmo depende también de otros factores, como la implementación de la matriz y la manera de gestionar el grafo. Por estas razones, se puede afirmar que el algoritmo tiene una **complejidad al menos $O(n^3)$**.