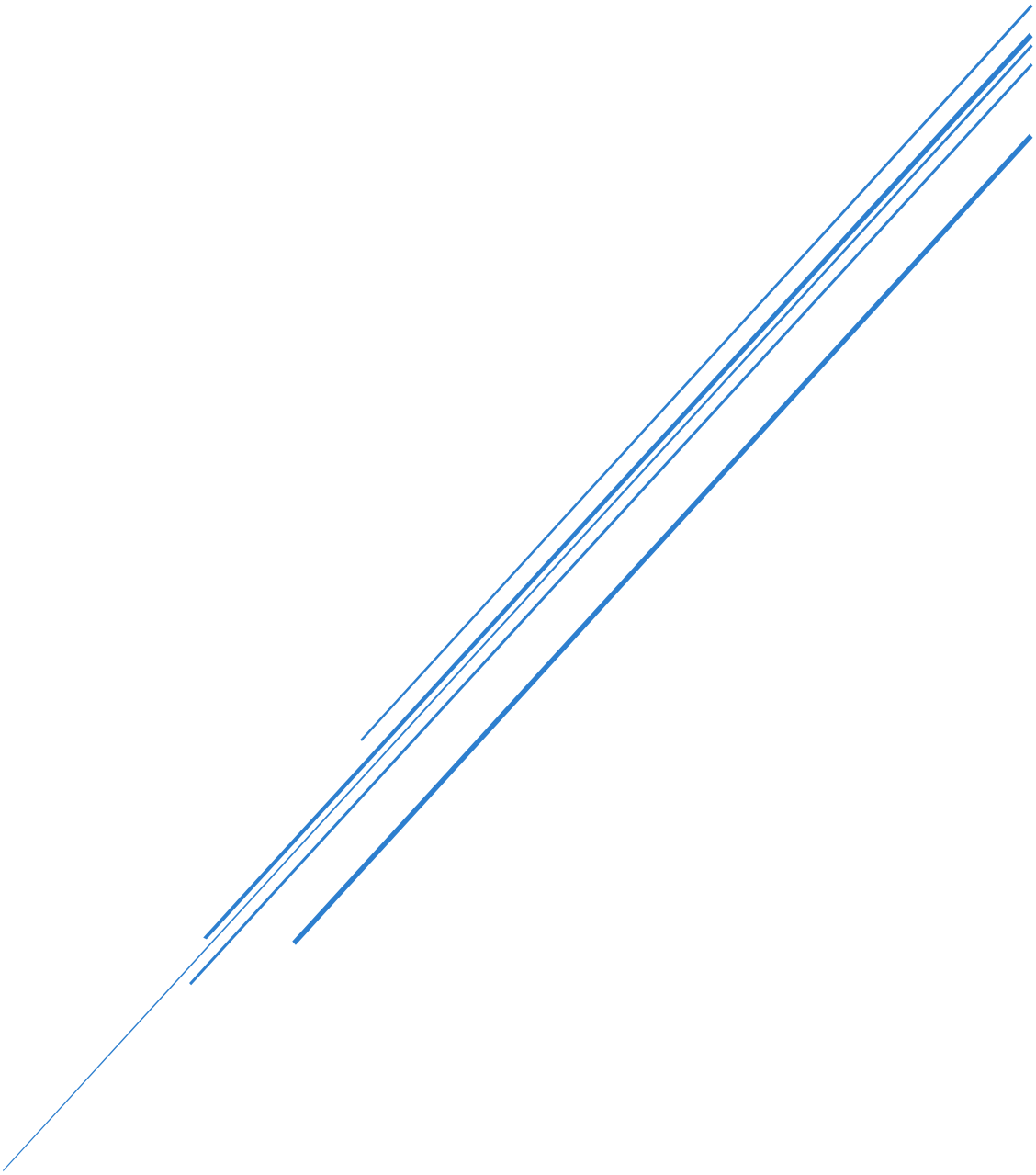
ALGORITMIA

Practicas de laboratorio 4



Contenido

Jorge Blanco Sánchez
UO293697

Algoritmo de Prim

En esta practica se ha implementado el algoritmo de Prim, este calcula el menor coste de unir todos los nodos de un grafo, después de implementar dicho algoritmo se procedió a tomar tiempos, con los siguientes resultados

n	t(s)
256	0,015
512	0,096
1024	0,654
2048	2,762
4096	11,530
8192	53,461
10000	78,015

Este algoritmo usa estructuras de datos intermedias, con las que para cantidades muy grandes de datos hace que se aumente el tiempo de procesamiento, debido a que tiene que almacenar ciertos datos en disco, por ejemplo y para mi equipo el cual posee 16GB de memoria principal y para el caso de 16384 elementos, tarda mas de 2000 segundos

Complejidad del Algoritmo

Este algoritmo usa un bucle while que se estará ejecutando hasta que todas las aristas estén visitadas, esto aproximadamente se ejecutará n veces, aunque en ciertos casos se pueden dar mas iteraciones, esto se debe a que las arista de cada nodo se guardan en un montículo de mínimos, si se da el caso de hay que sacar varias antes de llegar a una no visitada aumenta dichas iteraciones, por otra parte como se almacenan los datos en un montículo de mínimos, el coste de sacar los datos es log n, y dentro del while hay un if que comprueba que dicho nodo no ha sido visitados, haciendo que si entra en la condición se va a llamar a la función addHeap, la cual posee una complejidad de O(n), utiliza un for de la longitud de la matriz, en conclusión, la complejidad del algoritmo es: O(n)*O(n)*O(logn) = lo cual resulta en una complejidad de $O(n^2 \log n)$

Demostración para un caso

$$Tiempo = \frac{4096^2 * \log 4096}{2048^2 * \log 2048} * 2{,}762$$

Esta ecuación nos da un resultado de aproximadamente 12 s, lo cual es muy próximo a nuestro tiempo obtenido de 11,53 s, por lo que podemos concluir que se cumple la complejidad del algoritmo

Jorge Blanco Sánchez
UO293697