

Documento de análisis

Reto 4-EDA-2021-10-Grupo 12

Estudiante A: Julian Castro del Valle - j.castrod@uniandes.edu.co - 202020847

Estudiante B: Tomás Otero - t.otero@uniandes.edu.co – 202021733

Análisis de complejidad por requerimiento*:

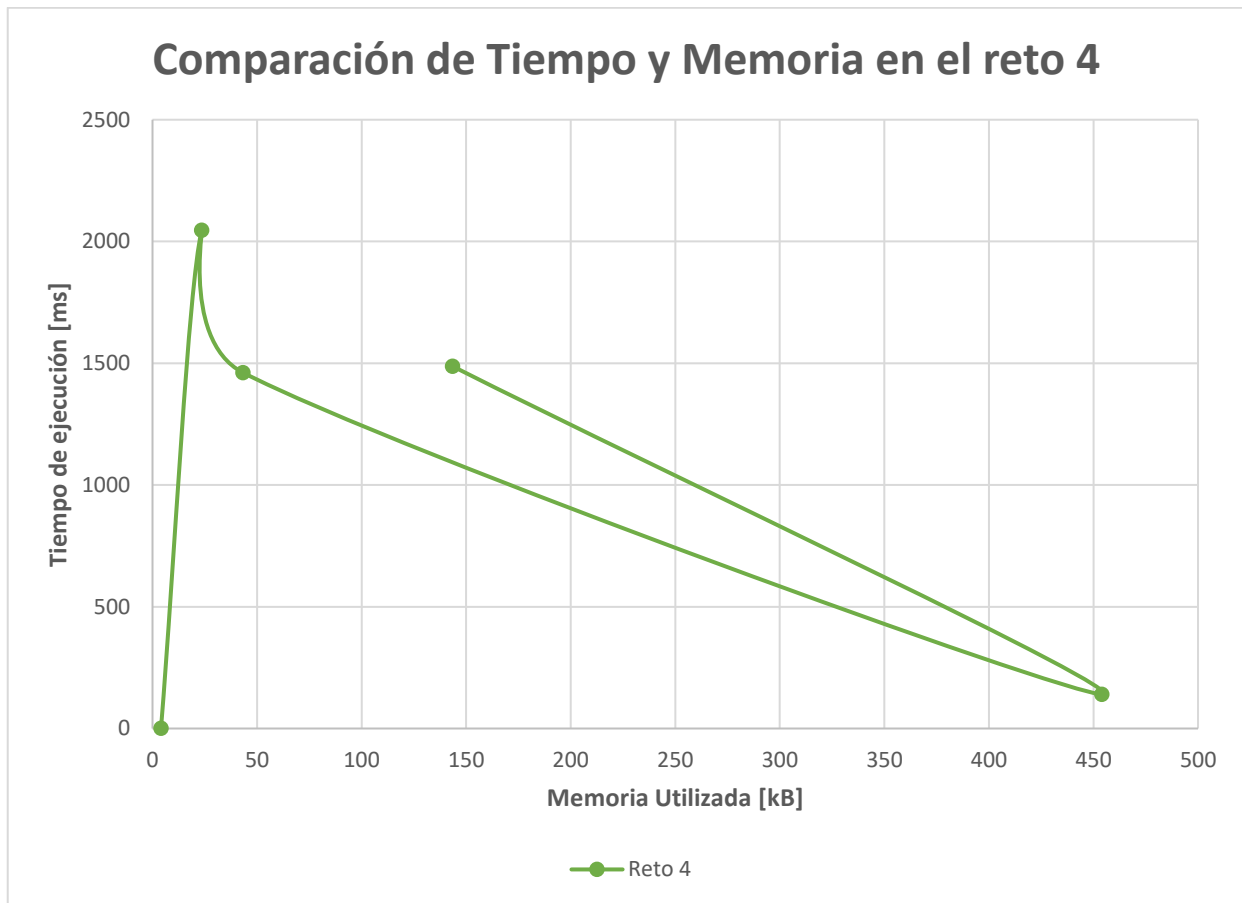
***Aclaración: en los análisis de complejidad que así aparezcan, V es el número de vértices y E el número de arcos del grafo.**

- Requerimiento 1: $O(V+E)$, lineal, ya que utiliza el algoritmo de Kosaraju para encontrar los SCCs.
- Requerimiento 2: $O(N)$ ya que cuenta con un ciclo for para recorrer el mapa de los vértices del catálogo.
- Requerimiento 3: $O(E \log V)$, ya que utiliza el algoritmo de Dijkstra para encontrar el camino mínimo entre los vértices solicitados por el usuario.
- Requerimiento 4: $E \log V$, ya que utiliza el algoritmo de Prim para hallar el MST del grafo.
- Requerimiento 5: $O(N)$, ya que cuenta con un ciclo for para recorrer los vértices adyacentes al ingresado por el usuario.

Se utiliza la estructura de datos tipo arreglo (ARRAY_LIST), al ser la más óptima en este caso en cuanto a velocidad y eficacia para almacenar datos. Adicionalmente, se usa el mecanismo de colisión Linear Probing, ya que también es el más eficiente entre los 2 mecanismos estudiados en clase, y se usan listas de adyacencias para los grafos al ser más eficientes que las matrices.

Análisis de tiempo de ejecución y memoria:

<i>Reto 4</i>		
Requerimiento	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
1.00	143.539	1487.292
2.00	454.113	140.699
3.00	43.219	1461.78
4.00	23.48	2046.141
5.00	4.068	1.657



Gracias a los análisis de tiempo y memoria realizados, se puede ver que este reto fue muy eficiente, posiblemente el mas eficiente. Los requerimientos analizados consumen una cantidad bajísima de memoria y toman tiempos de ejecución muy cortos, y eso que dichos datos fueron tomados en un portátil con apenas 8 GB de RAM y un procesador de 1.8 GHz. Esto nos puede llevar a concluir que, para máquinas lentas, la combinación entre los arreglos, grafos con listas de adyacencia y mapas con Linear Probing es óptima en términos de economización de tiempo y de memoria consumida.