OBSERVACIONES RETO 4

A continuación, se presenta la comparación entre el rendimiento del código implementado en el reto 4:

Lo primero que se hizo fue elegir 5 parejas de paises al azar y se realizaron los dos requerimientos en cada uno:

Gráfico, Gráfico de barras

Descripción generada automáticamenteGráfico, Gráfico de barras

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

Teniendo la informacion recolectada por los 5 pares de paises podemos hace dos graficas donde se comparen los tiempos y usos de memoria:

Gráfico, Gráfico de barras

Descripción generada automáticamente

Es evidente que en la mayoria de casos es consistente como los tiempos del requerimiento 2 son los mas altos pero no los mas consistentes por otro lado en el requerimiento 1 los tiempos son los menores y los mas consistentes.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Es rotundamente notorio el alto uso de memoria que tiene el requerimiento uno frente al requerimiento 2.

De las dos graficas anteriores se obtuvieron los promedios se grafican junto al requerimiento 3 el cual se realizo 5 veces y tambien se promedio.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Como se puede observa en la grafica es notorio que los tiempos van aumentando de manera progresiva.

Gráfico, Gráfico de barras

Descripción generada automáticamente

El req3 es el requerimiento que mas memoria consume seguido del req1.

Complejidad O(n):

Requerimiento 1:

El requerimiento 1 de los componentes fuertemente conectados presenta dos procesos diferentes el primero es el uso del algoritmo de Kosaraju que tiene de complejidad O(v+e) donde V son los vertices y E los arcos del grafo , y en el segundo proceso del requerimiento 1 es complejidad O(1).

Requerimiento 2:

Para el segundo requerimiento el cual es el del ruta minima que se compone principalmente del algortimo de Dijkstra que tiene como complejidad O(elog(v)) donde V son los vertices y E los arcos del grafo , ademas hay que hallar la ruta entre dos puntos y esa complejidad dependera de la cantidad de vertices entre los dos puntos entonces si hay pocos puntos la complejidad seria O(1) pero si hay demasiados vertices y se acerca a la cantidad total de vertices esto nos representara una complejidad O(v).

Requerimiento 3:

El requerimiento 3 que consiste en la red de expansion minima se usal el algoritmo de prim de complejidad O(elog(v)) donde V son los vertices y E los arcos del grafo, ahora para encontrar la rama más larga en el peor de los casos tendriamos O(v) y todo esto debido a que tendriamos que recorrer todos los vertices.