Reto 1

Carga:

Tiempo: 27323ms

Memoria:1321174kb

Requerimiento 1:

Tiempo: 2546ms

Memoria:22kb

Requerimiento 2:

Tiempo: 3545ms

Memoria:17

Requerimiento 3:

Tiempo: 1451ms

Memoria: 4kb

Requerimiento 4:

Tiempo: 80472ms

Memoria: 14kb

Reto 2

Carga:

Tiempo: 268320ms

Memoria: 21572124kb

Requerimiento 1:

Tiempo: 34780ms

Memoria: 327kb

Requerimiento 2: Tiempo excedido

Requerimiento 3: Tiempo excedido

Requerimiento 4:

Tiempo: 120240ms

Memoria: 423kb

Para empezar, podemos evidenciar que la función de carga del reto 1, de 27323ms, es mucho más rápida que la del reto 2, de 268320ms, debido a su simplicidad. La función de carga del reto 2 está compuesta por varias funciones que hacen recorridos para organizar los datos en las tablas de hash, mientras que en el reto 1 todos los datos de meten directamente a un arreglo. La simplicidad de carga para el primer reto también hace que el uso de memoria sea reducido, la memoria utilizada para la carga de datos en el reto 1 fue de 1321174kb mientras que en el reto 2 fue de 21572124kb. El primer requerimiento del reto 1 es más eficiente en tiempo y memoria que el primer requerimiento del reto 2, pues el tiempo de ejecución es de 2546ms mientras que en el reto 2 es de 34780msy la memoria utilizada es de 22.1kb en el reto 1 y de 327kb en el 2. En cuanto al segundo y tercer requerimiento del reto 2 nos dimos cuenta de que no era la forma más eficiente posible pero no tuvimos tiempo de mejorar el código. Esto hace que el tiempo de carga de estos requerimientos del reto 1 sea masivamente mejor que en el reto 2. Para el ultimo requerimiento, también se evidenció mejor eficiencia temporal y de memoria en el reto 1 que en el 2. En tiempos el reto 1 se demoró 80472ms mientras que el reto 2 se demoró 120240ms. En memoria el reto 1 consumió 14kb mientras que el reto 2 consumió 423kb.