

PROGRAMACION 3

Primera entrega

Integrantes:

Colamai, Daniela

Martino Martinez, Fabian

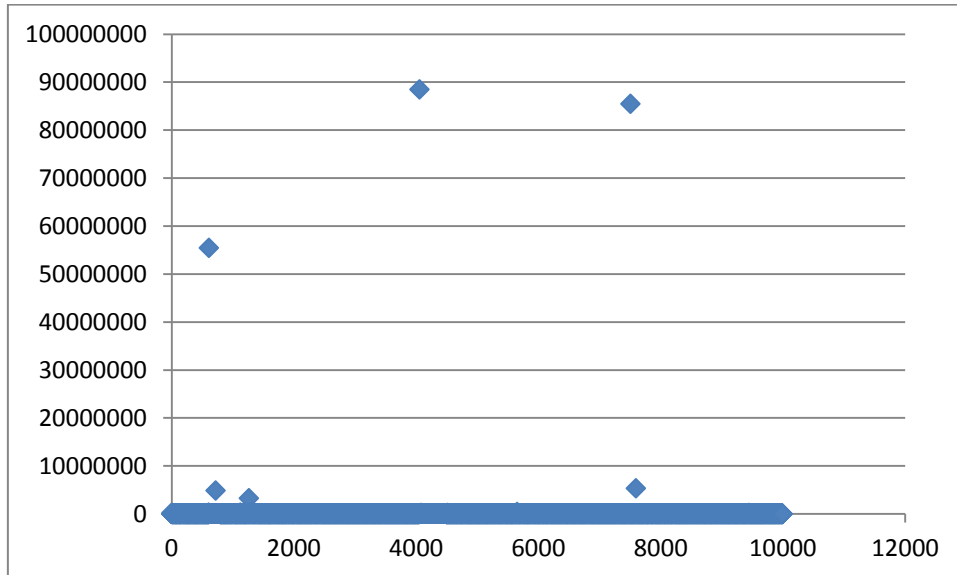
Introducción

Para esta entrega si nos planteó realizar el código para el pre-cargado, el alta y la búsqueda de usuarios utilizando distintos métodos registrando los resultados. También se realizaron comparaciones entre los métodos y sus implementaciones utilizando estadísticas provistas por los resultados de las mismas. Debemos también aclarar que las pruebas solo pudieron ser realizadas sobre el dataset de 500000 elementos debido a errores por falta de memoria.

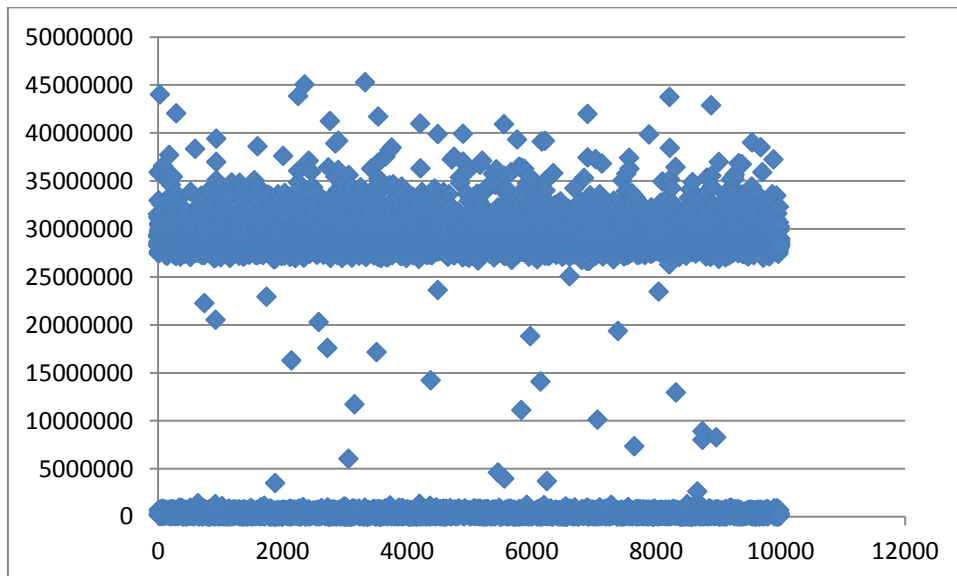
Listas:

En el caso de las listas se implementaron dos métodos distintos para el alta de usuarios por lo que vamos a mostrar los resultados de ambas por separado:

- Alta al principio:

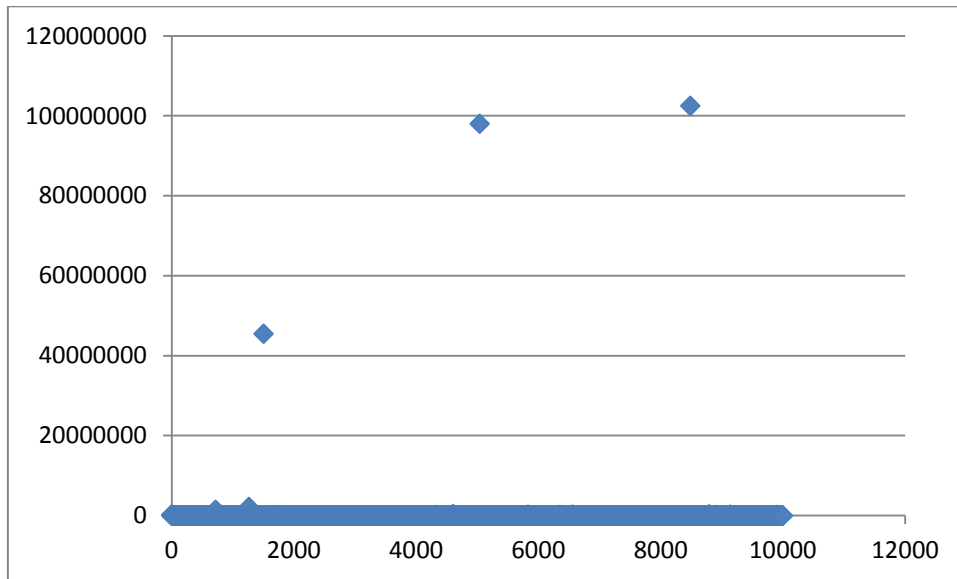


Para el alta de elementos al principio de la lista debemos tener en cuenta que la lista queda invertida, es decir que el primer elemento ingresado es el último en la lista, además según nuestros análisis se tardó 348001123ns para las 10000 altas con un promedio de 34800ns y 88464152ns en el peor de los casos.

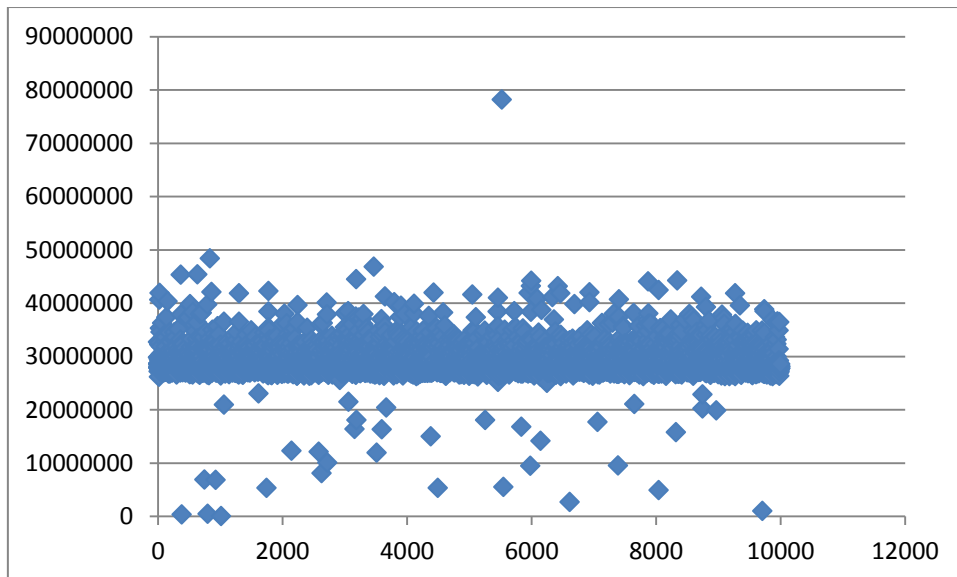


Debido a que la lista no está ordenada no pudimos aplicar métodos más eficientes de búsqueda. El tiempo total de búsqueda es de 937367911ns con un promedio de 93736ns y 45263310ns en el peor de los casos.

- Alta al final:



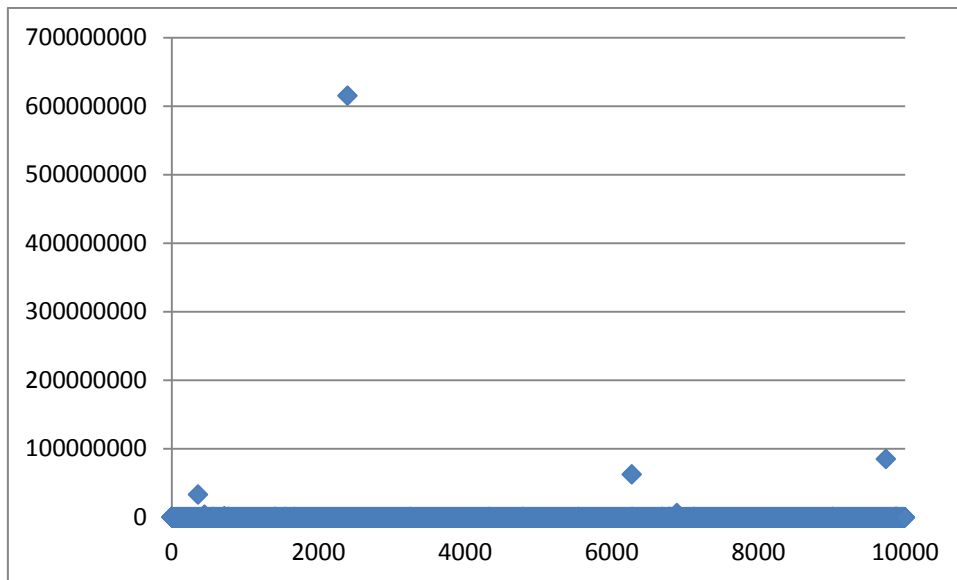
Gracias a que para realizar el alta en la última posición mantenemos un puntero al último elemento evitamos recorrer todos los elementos de la lista cada vez que se realiza un alta de elementos y así reducimos el tiempo. Tarda 329127077ns en realizar el alta para 10000 usuarios, tiene un promedio de 32912ns por alta y 102446918ns como el peor de los casos.



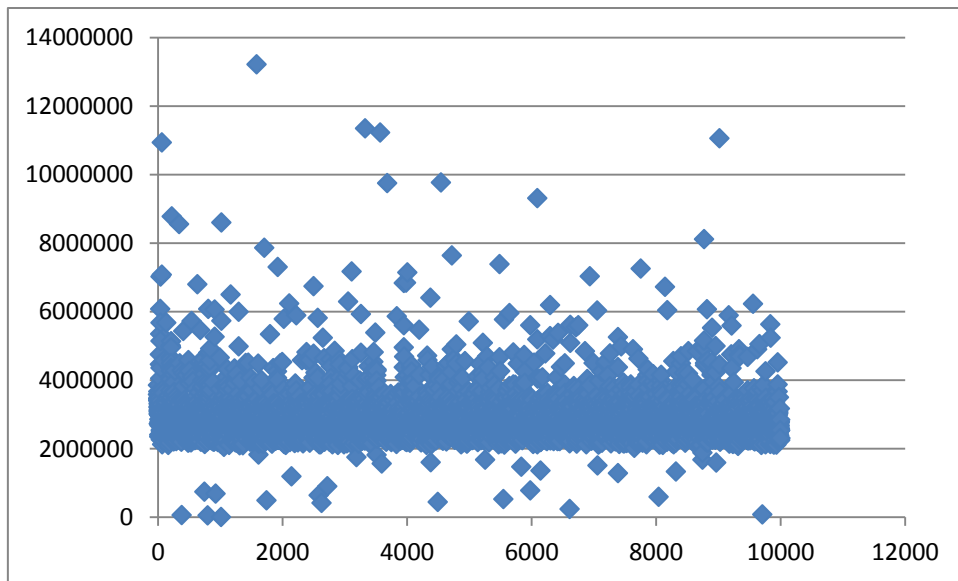
Por las mismas razones que para el alta al principio de la lista tomo 1711588841ns el realizar el alta para 10000 usuarios con un promedio de 171158ns y se registró un tiempo de 78151152ns como el peor de los casos.

Arreglos:

En el caso de los arreglos se genera un arreglo inicial con 10000 posiciones, al llegar al límite de usuarios el tamaño se duplica generando un nuevo arreglo con el doble de posiciones y se vuelvan dos datos a este. Para la búsqueda se recorre el arreglo hasta que encuentra al usuario buscado o recorre todos los elementos del arreglo y gracias a que llevamos la cuenta de cuantos usuarios tenemos registrados no necesitamos buscar en las posiciones que se encuentran vacías.



El alta de elementos lleva en líneas generales un tiempo parejo a excepción de momento específico como el caso donde se requiere aumentar el tamaño del arreglo. En nuestras pruebas el tiempo total suele ser de 886032401ns, tiene un promedio de 88603ns y en el peor de los casos tardo 614856244ns.



Debido a que el arreglo no se encuentra ordenado es difícil saber en qué posición se encuentra el usuario que estamos buscando por lo que los tiempos varían de elemento a elemento. Calculamos un tiempo total de 2009547870ns un promedio de 200954ns y en el peor de los casos tardo 13211157ns.

Conclusión:

Para el caso del alta, la implementación de un puntero al último elemento mejoro la eficiencia de la carga al final de la lista al punto de que sea parejo con el tiempo necesario para el alta al principio de la lista, ambas registraron poco más de un tercio de tiempo del tiempo necesario para el alta en arreglos y esto se debe a que se requiere constantes aumentos de tamaño en este último.

Los métodos de búsqueda podrían ser más efectivos si los elementos se encontraran ordenados pero en general mantuvieron tiempos similares ya sea para lista como arreglos y para ambas el peor de los casos se daba cuando se debía recorrer todos los elementos.