Universidad del Valle de Guatemala
Algoritmos y Estructura de Datos

Julio Herrera 19402
Ayleen Rubio 19003

Hoja de Trabajo No. 6
Operaciones con mapas

Condiciones a evaluar

- Agregar un dato.
- Buscar un dato.
- Recorrer el mapa.
- Ordenar el mapa.

Conclusión

- Para agregar un dato se prefiere la utilización de HashMap debido a que este tiene un rendimiento constante, al igual que para buscar un dato, sin embargo, al trabajar con una cantidad de datos muy elevada, la complejidad del HashMap ya no será constante.
- Para recorrer el mapa completo el mapa más eficiente es LinkedHashMap ya que cuenta con el rendimiento de un HashMap y con el ordenamiento de un TreeMap.
- En el ordenamiento de datos se prefiere utilizar TreeMap, ya que de por sí los datos que almacena son guardados en orden, sin embargo, tiene una complejidad logarítmica, la cual no cambia.
- Es más conveniente utilizar LinkedHashMap debido a las propiedades que tiene sobre la eficiencia constante y el ordenamiento con el que viene predeterminado, que en este caso es muy útil para los fines del programa.

Complejidad de tiempo HashMap

La complejidad de tiempo de HashMap depende de la función de hash que utiliza para escoger la llave del valor a guardar, tanto en la complejidad del algoritmo como en la utilización de los valores que devuelve para utilizar como llaves, la complejidad puede variar de O(1) con un hash que no devuelva llaves repetidas y de O(n) en caso de que el hash utilizado cree muchas colisiones de llaves repetidas. Otro factor que influye en la complejidad de un HashMap es el tamaño que este reserva en memoria, ya que a costa de una complejidad menor, se tiene que sacrificar un poco más de espacio en memoria para no tener que agrandar el espacio y copiar la lista en una nueva al sobrepasar el límite cada vez que se agregue un elemento que lo supere.