

HashSet (Resumen Clase Semana 5)

HashSet primero se debe definir una función hash y su característica principal es que es determinista, esto quiere decir por ejemplo si tengo un String de 4 caracteres me tiene que devolver 4, es un mapeo de la cantidad de caracteres y el entero que representa esa cantidad de caracteres.

- ¿Qué pasa si no es determinista?

Puedo llegar a obtener elementos repetidos.

- Caso hashCode más grande que el tamaño del hash set?
Se soluciona el problema calculando el módulo.

Recapitulación

- Set: corresponde a la representación de una colección de elementos tal que no existen elementos duplicados.
- HashSet: corresponde a la implementación de set donde, los elementos almacenados en un arreglo, cada elemento tiene asociado un hashCode que es utilizado para determinar su índice en el arreglo.

Colisiones en HashSets ¿Cómo solucionar colisiones en HashSet?

Una opción es ocupar una cadena de separación (separate chaining), mantengo un arreglo con elementos determinados (buckets) pero lo que va almacenar son listas enlazadas.

- Elementos son almacenadas en arreglos de listas enlazadas, o buckets.
- Una vez que el hashCode es utilizado para determinar el número de bucket, todas las operaciones son las mismas de una lista enlazada.

¿Qué pasa cuando se tiene buckets con demasiados elementos?

Una solución para este caso es que el hashset debe automáticamente redimensionar el número de buckets. Redimensionar el número promedio de elementos por bucket, para eso hay que definir el tamaño actual, la cantidad de buckets, y el factor indicador, entonces la ecuación sería $\text{factorIndicador} = (\text{tamanoActual} / \text{buckets.length})$ y el valor obtenido por ejemplo en caso de que el tamaño sea 4 y los buckets 6, el factor indicador sería 0,67, entonces si este valor es menor cantidad de elementos de lo que indica el umbral definido de elementos por ejemplo 0.75 y el factor indicador es 0.67, quiere decir que es suficiente y no hay necesidad de redimensionar.