

# Analisis Estructuras de datos

Paolo Escobar

July 2, 2018

## **Comparacion de Estructuas-insersion:**

Las estructuras de datos tilizadas tienen tiempos relativamente dispares en tiempos de insercion, de forma teorica nos encontramos con que la peor estructura de datos en terminos de tiempo de complejidad son los arboles, debido a la comprobacion y rebalanceo constante segun cada caso. Mientras que las mejores son las listas; tanto en el mejor como en el peor caso; y el Hash en mejor caso, ya que, en el peor caso se comporta como una matriz. Estos valores son bastante parecidos en bajos volúmenes de datos (10, 20) pero a medida que la cantidad crece las diferencias se ven mas evidentes, en torno a los 1000 datos llegan a ser milisegundos.

## **Eliminacion:**

En lo que corresponde a eliminacion de datos, la mejor opcion es Hash el cual se comporta como un arreglo, mientras que las peores son las listas ya que deben recorrer; toda la lista en el peor caso; en busqueda del nodo

## **Imprimir:**

En este caso particular, es muy interesante ya que al no existir en Hash una forma directa de lograr acceder a todos los nodos lo vuelve la peor situacion posible; teniendo que guardar las llaves en un arreglo para asi poder revisar el Hash llave por llave, comprobando que no hayan colisiones, que de ser el caso; se comporta como una especie de matriz multidimensional. Debido a esto, el Hash es la peor estructura en este caso mientras que los arboles son la mejor debido a su facil ordenamiento.

## **Conclusion:**

La mejor estructura de datos siempre sera la que corresponda al contexto de la implementacion misma, por ejemplo, en un caso donde solo se necesite insertar datos y mostrar estos, la listas seran la mejor opcion. Por otra parte, si se necesita facilidad en torno a ordenamiento y facil busqueda e impresion de los datos nuestra mejor opcion seran arboles. Finalmente, si la implementacion requiere la mejor complejidad posible para busqueda, eliminacion e insercion el Hash es la estructura mas optima.