

PROYECTO DE ESTRUCTURA

ALGORITMOS DE SORTING

El objetivo con este trabajo es usar distintos algoritmos de sorting para comparar su eficiencia en la practica basandose en los tiempos de los casos promedio de cada uno. Para llevar a cabo este proyecto se usaron 3 algoritmos de sorting: quick-sort, tree-sort y heap-sort, con dos implementaciones distintas para este ultimo, una con arreglos y otra con listas dinamicas. Para los casos de prueba, se utilizara un arreglo previamente cargado con numeros aleatorios. La capacidad o cantidad de elementos a probar seran valores que se iran incrementando. Se probara con 10, 100, 1000, 10000, 100000, 1000000 elementos.

QUICKSORT

Se dice que el tiempo de este algoritmo de sorting depende de la implementacion del partition. Para este caso, se selecciona como pivot al primer elemento por lo que el peor caso hace que el partition sea orden N , pero en la practica se comporta como $n * \log n$. Asi mismo veremos que este es el menos eficiente de los tres algoritmos ya que solo se puede ordenar colecciones de hasta 100000 elementos sin presentar problemas.

TREESORT

Si bien en el peor caso este algoritmo es de orden N , en la practica se comporta como un algoritmo de orden $n * \log n$. El algoritmo fue probado con colecciones de hasta 1000000 elementos comportandose bien. Se realizo una prueba con cronometro y se controlo que solo demora 1segundo en ordenar la colección completa.

HEAPSORT (ARREGLOS)

{RESULTADOS DEL HEAPSORT CON ARREGLOS}

HEAPSORT (LISTAS DINAMICAS)

{RESULTADOS DEL HEAPSORT CON LISTAS DINAMICAS}