Estructuras de Datos Avanzadas
Tarea

Comparación de Algoritmos de Ordenamiento

El objetivo de esta tarea es determinar empíricamente el desempeño de los algoritmos de ordenamiento vistos en clase.

Instrucciones

Implementar cada uno de los algoritmos de ordenamiento vistos en clase (implemente las dos versiones de Merge Sort vistas en clase). La implementación debe de ser genérica, independiente del tipo de datos a ordenar (tip: que la clase a ordenar implemente la interface "comparable" y ahi definen qué campo de la clase se usa para la comparación.). Adicionalmente, por conveniencia, escriba todos los algoritmos como miembros de una sola clase.

Para cada uno de los siguientes incisos entregar dos figuras (A y B), una para el número de comparaciones y otra para el tiempo de ejecución (medido en una escala apropiada). Cada figura debe de tener el resultado de todos los algoritmos para poder comparar su desempeño. Entregue un documento en .pdf con las gráficas y su interpretación—sus observaciones acerca del desempeño. El trabajo es individual. No incluya código, el código se revisará en el salón. Entregue via Github usando el repositorio correspondiente a esta tarea.

Procedimiento

Datos Ordenados

1. Obtener el archivo business10k.json de la carpeta de datos: sigua la liga a Dropbox que esta en el documento de Ejercicios de Github. Tip: haga una clase censo para almacenar los registros.
2. Comience con los datos ordenados.
3. Ejecutar cada uno de los algoritmos de ordenamiento vistos en clase para $v \geq 5$ valores distintos de n. Escoga los valores de n de manera que se aprecie como cambia el desempeño del algoritmo con el tamaño de la entrada; es decir, escoga tamaños lo suficientemente distintos, e.g., n=100, n=1000, n=2500, n=5000, n=10,000.
4. Grafique la curva de desempeño de cada algoritmo en la figura correspondiente (use A para el número de comparaciones y B para el tiempo).

Orden inverso

1. Obtener el archivo business10k.json
2. Comience con los datos en orden inverso.
3. Ejecutar cada uno de los algoritmos de ordenamiento vistos en clase para $v \geq 5$ valores distintos de n. escoga los valores de n de manera que se aprecie como cambia el desempeño del algoritmo con

el tamaño de la entrada; es decir, escoga tamaños lo suficientemente distintos, e.g.,n=100, n=1000, n=2500, n=5000, n=10,000.

4. Grafique la curva de desempeño de cada algoritmo en la figura correspondiente (use A para el número de comparaciones y B para el tiempo).

Orden Aleatorio

1. Obtener el archivo business10k.json
2. Ejecutar cada uno de los algoritmos de ordenamiento vistos en clase 30 veces para $v \geq 5$ valores distintos de n.
   - n es el número de datos a ordenar. Escoga los valores de n de manera que se aprecie como cambia el desempeño del algoritmo con el tamaño de la entrada; es decir, escoga tamaños lo suficientemente distintos, e.g., n=100, n=1000, n=2500, n=5000, n=10,000.
   - Cada una de las 30 corridas para un algoritmo y un tamaño n debe hacerse sobre un subconjunto de los datos elegidos al azar. Para este fin elabore una función que dado un conjunto m de elementos y un entero $n \leq m$ regrese n elementos escogidos uniformemente al azar sin reemplazo.
   - Promediar los resultados de las 30 corridas para cada n y para cada algoritmo (en total tendrán v valores para cada algoritmo).
3. Grafique la curva de desempeño de cada algoritmo en la figura correspondiente (use A para el número de comparaciones y B para el tiempo).