

Tarea Programada Estructuras de Datos

Descripción:

Las estructuras de datos hacen referencia a diferentes maneras de organizar la memoria al almacenar los datos. Este tema es de vital importancia para el mundo moderno, debido a la cantidad y variedad de datos que se manejan. Diferentes estructuras presentan diferentes ventajas al organizar los datos, proveyendo variedad de tiempos al momento de acceder, insertar, modificar y eliminar datos; es por ello que es importante su estudio, para entender sus diferencias, desventajas y fortalezas.

Enunciado:

En esta tarea el objetivo consiste en implementar las diversas estructuras de datos vistas en clase y realizar experimentos de medición asociados. Para ello deberán implementar las siguientes estructuras de datos: lista enlazada, árbol binario simple, árbol binario autobalanceado y tabla de dispersión (tabla *hash*). (+5 puntos extra si investigan, implementan y utilizan un segundo tipo de árbol autobalanceado).

Para cada una de las clases debe implementar su constructor, destructor, método de *inserción* y método de *búsqueda*. En el caso de la lista enlazada y del árbol binario simple deberá implementar además un método de *borrado*. En todos los casos deberá hacer un manejo adecuado de memoria y no poseer *ninguna* fuga.

Para cada una de las estructuras implementadas deberá evaluar el rendimiento de las estructuras para 2 diferentes casos: *inserción* de valores ordenados e *inserción* de valores aleatorios. Para cada uno de los grupos de valores insertados, deberá tomar también mediciones de la duración al realizar *búsqueda* aleatorias. En el caso de la lista enlazada y del árbol binario simple deberá medir además la duración del borrado de elementos aleatorios.

Deberá repetir cada una de las mediciones para una cantidad variable de inserciones/búsquedas/borrados, de tamaños: 131072 (2^{17}), 262144 (2^{18}), 524288 (2^{19}), 786432 ($2^{19} + 2^{18}$) y 1048576 (2^{20}). Los valores aleatorios a ser generados para los métodos deben estar en el intervalo de **[0, 2n]** donde **n** es la cantidad de inserciones, búsquedas o borrados a realizar. Considere que el método `rand()` varía de computadora en computadora, y en algunos casos sólo alcanza valores máximos cercanos a 32k, por lo que deberá usar su creatividad para hacerle frente a este problema.

Las mediciones de las duraciones deben hacerse con precisión en milisegundos, por lo que se le recomienda utilizar la biblioteca `<chrono>` para este fin. Además, lo ideal sería tomar 3 veces cada medición y calcular su tiempo con base en el promedio de las 3 mediciones.

Con los datos obtenidos deberá redactar un entregable escrito con los resultados de sus experimentos: tablas con los valores obtenidos y gráficos de comparación entre los

tiempos de duración de las diferentes estructuras. Deberá explicar el comportamiento observado y si dicho comportamiento es el que esperaba por parte de sus estructuras.

Evaluación:

- Lista enlazada y árbol binario simple (15 puntos c/u):
 - Constructor y destructor: 2 puntos
 - Inserción: 5 puntos
 - Búsqueda: 3 puntos
 - Borrado: 5 puntos
- Árbol balanceado y tabla de dispersión (15 puntos c/u):
 - Constructor y destructor: 2 puntos
 - Inserción: 8 puntos
 - Búsqueda: 5 puntos
- Código main (10 puntos):
 - Debe permitir ejecutar las pruebas y verificar el funcionamiento del código
- Documento escrito (30 puntos):
 - Debe incluir los gráficos y tablas con los datos obtenidos.
 - Utilice los estándares para numeración de tablas e imágenes.
 - Puede redactarlo en Word o Latex pero debe entregarse como PDF.
 - Las gráficas puede hacerlas con su herramienta favorita (e.g: Excel, Python, Drive, etc.)
 - Si menciona algo proveniente de algún libro o artículo, recuerde citar.

Se puede trabajar de manera individual o en parejas. Pero solo **una** persona por grupo debe enviar el trabajo, de lo contrario perderán 5 puntos por envío duplicado. (Los asistentes siempre agradecen cuando evitan enviar duplicados)