

Laboratorio 1

Diseño de algoritmos

Instrucciones generales:

- Debe ser realizado en forma individual.
- Se entrega en el aula virtual a más tardar el 27 de Abril de 2015 en horario de ayudantía/laboratorio. Se recuerda que no se revisarán entregas atrasadas.
- El formato de entrega es un archivo comprimido con todos los códigos fuentes (proyecto si tiene) e informe en pdf (no se aceptará otro formato).
- Tendrá que hacer su trabajo dos lenguajes de programación: C++, Java, C# o Python.
- Para C++ debe incluir un archivo Makefile.
- Si realiza su informe en \LaTeX , tendrá puntaje extra.

Objetivos

- Verificar en forma empírica el tiempo y complejidad de distintos tipos de algoritmos de ordenamiento.
- Realizar análisis asintótico de distintos tipos de algoritmos de ordenamiento.

Enunciado: Ordenamiento

Considere los siguientes métodos de ordenamiento:

- Intercambio directo (burbuja).
- Ordenamiento por división par (Mergesort)
- Ordenamiento por casilleros.
- Ordenamiento Radix.
- Ordenamiento rápido (quicksort)

Para cada uno de ellos:

- Escriba el algoritmo en pseudocódigo.
- Calcule su tiempo y complejidad.
- Implemente en un lenguaje de programación.

- Verifique si el tiempo y la complejidad calculada anteriormente es la misma que la de sus programas computacionales.

Pruebe cada uno de los métodos implementados y mida el tiempo de ejecución de forma exacta considerando 3 instancias de tamaño 1000000 que representen:

- Arreglo ordenado de menor a mayor (“`menorMayor.in`”).
- Arreglo ordenado de mayor a menor (“`mayorMenor.in`”).
- Arreglo desordenado (“`desordenado.in`”).

Hay que destacar que los números no necesariamente tienen que ser correlativos y se pueden repetir.

Entrada La entrada de los datos será un archivo de texto plano. El nombre corresponde al orden del arreglo utilizado descrito más arriba. Cada elemento del arreglo se encuentra en una línea del archivo. El archivo termina con un enter final.

Salida El arreglo ordenado deberá ser escrito en un archivo llamado con el nombre del método y el nombre del archivo de entrada (sin el “`in`”) `MÉTODO_NOMBRE_ARCHIVO_ENTRADA.out`. La característica del archivo es la misma que la entrada. Para realizar los cálculos de tiempo, no considere la entrada ni la salida de datos ni muestre datos por pantalla (esto reduce el tiempo de ejecución).

En base al trabajo realizado, responda las siguientes preguntas:

1. ¿Qué algoritmo es mejor? ¿Es esta elección válida para instancias de cualquier tamaño? De no ser así, indique cuál es el tamaño crítico de la entrada y qué algoritmo es mejor para instancias más pequeñas.
2. Indique si hay algoritmos con igual complejidad. En caso de ser así, determine cuál es mejor tomando en consideración las constantes descartadas.
3. ¿Se cumple la premisa de que las instrucciones simples están acotadas por un tiempo que depende de la implementación? Justifique su respuesta.

Informe

Debe ser escrito en tercera persona y tiempo presente. El formato de impresión a utilizar es el de presentación de memoria de título, encontrado en el aula virtual.

Debe contener los siguientes elementos:

- Portada.
- Índice.
- Introducción (1 página): establece la motivación y los objetivos del trabajo. Indicar en qué máquina (características) se realizaron las pruebas. Además describe brevemente la estructura del resto del documento.

- Marco teórico.
- Desarrollo del trabajo:
 - Algoritmos en pseudocódigo y breve explicación del funcionamiento de cada método de ordenamiento.
 - Cálculo detallado de la complejidad y tiempo de cada uno.
 - En un gráfico, graficar el tiempo de sus pruebas.
 - Detalle de las instancias utilizadas o cómo crearon las instancias para las pruebas (usar columnas para disminuir la cantidad de páginas).
 - Respuesta a las preguntas planteadas.
- Conclusión: presenta un análisis de los resultados obtenidos e indica si se cumplieron los objetivos planteados en la introducción.
- Referencias: deben ser escritas según estándar APA y deben aparecer debidamente señaladas en el documento.

Requerimientos

1. Debe entregar su código, se evaluará el orden, la separación en funciones, la mnemotecnia y el uso de comentarios.
2. Los documentos deberán ser presentados impresos en papel y en formato pdf.

Observaciones importantes

El informe que debe entregar, tiene que mostrar claramente las razones de las decisiones que usted tomo en el sistema, tanto de navegación, interfaz de usuario, funcionalidades que usted presenta. Además el informe debe mostrar las decisiones tomadas con respecto a la implementación y uso del lenguaje.

El sistema debe ser robusto, **se penalizarán las caídas** de cualquier tipo. Debe además estar bien documentado, si así lo desea puede presentar la documentación creada con JavaDoc o Doxygen por ejemplo (no es obligatorio).

La copia de cualquier tipo (incluso de internet) **será evaluada con un 1.0**. Se han dejado a propósito algunos temas a su criterio, la intención es que usted sea autocrítico en su trabajo y se esfuerce en entregar el mejor sistema (incluidos documentos) posible, por lo que un sistema que sólo cumpla con los criterios mínimos mencionados en este enunciado no aseguran un 7, recuerde, sea eficiente, no sólo eficaz.

El enunciado podría sufrir modificaciones.