



CI-1221 Estructuras de Datos y Análisis de Algoritmos

I ciclo de 2018

I Tarea Programada

Fecha límite de entrega:

Primera parte: miércoles 5 de setiembre

Segunda parte: miércoles 19 de setiembre

1. Objetivo de la tarea

El objetivo de la tarea es implementar los algoritmos de ordenamiento y de búsqueda de la mediana ($n/2$ -ésimo elemento más pequeño de un arreglo de tamaño n) vistos en el primer cuarto del curso y comprobar que sus diferencias en cuanto a eficiencia teórica corresponden con la realidad.

2. Algoritmos a implementar

Los algoritmos a implementar son los siguientes: (I) ordenamiento por selección [7 pts.], (II) ordenamiento por inserción [7 pts.], y (III) ordenamiento por mezcla (*merge sort*) [7 pts.] para la **primera parte**, y (IV) ordenamiento por montículos (*heapsort*) [7 pts.], (V) ordenamiento rápido (*quicksort*) [7 pts.], (VI) ordenamiento por residuos (*radix sort*) con dígitos en base $2^{\lceil \lg n \rceil}$ [10 pts.] y (VII) búsqueda de la mediana [7 pts.], para la **segunda parte**.

3. Proceso de comparación

Los pasos a seguir para comparar los algoritmos son los siguientes:

1. Los algoritmos se deben comparar usando arreglos de enteros (positivos y negativos) seleccionados al azar, de tamaño 100 000, 200 000, 300 000, y 400 000. Ejecute cada uno de los algoritmos al menos tres veces con cada uno de los arreglos.¹ Registre en un cuadro cada uno de estos tiempo y su respectivo promedio. Refiérase

¹Si el tiempo mayor es 1,5 veces más grande que el tiempo menor, se debe posiblemente a que la máquina estuvo ocupada en otras cosas mientras hizo el ordenamiento. En tal caso elimine una de

- a la variación de los tiempos en las cinco corridas (es decir, si fueron similares, iguales, muy distintos, etc.). [1 pto. c/ algoritmo].
2. Grafique los tiempos promedio contra el tamaño del arreglo para cada uno de los algoritmos y analice la curva resultante para identificar si su forma es la esperada (es decir, si las curvas de los algoritmos $\Theta(n^2)$ son aproximadamente parabólicas y si las curvas de los algoritmos $\Theta(n)$ y $\Theta(n \log n)$ son aproximadamente lineales).² Asegúrese de indicar en cada gráfico las unidades de tiempo utilizadas: s, ms, etc. [1 pto. c/ algoritmo].
 3. Grafique de nuevo los tiempos promedio de ejecución pero póngalos en un mismo par de ejes. Muestre el tiempo en escala logarítmica para contrarrestar las enormes diferencias que probablemente existan entre los tiempos de ejecución de los algoritmos cuadráticos y los (aproximadamente) lineales. Identifique si la relación entre las curvas fue la esperada [1 pto. c/ algoritmo].

Este proceso debe ser explicado y analizado con detalle en el reporte de la tarea (ver sección 4.1). Por ejemplo, en el reporte se debe indicar si buscar la mediana mediante el algoritmo específico es significativamente mejor que ordenar el arreglo y luego tomar el elemento en el medio.

4. Entregables de la tarea

La tarea tiene dos entregas. La primera es un informe preliminar con un análisis de los algoritmos de ordenamiento por selección, inserción y mezcla, y la segunda un informe final con esos algoritmos y los cuatro restantes: ordenamiento por montículos, ordenamiento rápido, ordenamiento por residuos (con dígitos en base $2^{\lceil \lg n \rceil}$) y búsqueda de la mediana. A cada uno de los reportes debe adjuntarse el código de los algoritmos en el archivo «Ordenador.h» suministrado con este enunciado. Opcionalmente, para la segunda entrega se puede adjuntar la «hoja de chequeo de estilo» adjunta con la que se pueden ganar puntos extra si el informe se adhiere a ciertas reglas de estilo (ver sección 4.1.4). A continuación se detallan cada uno de los entregables.

4.1. Informes

En las secciones que siguen se definen el contenido, los idiomas aceptados, el formato y el estilo de los informes.

4.1.1. Contenido

Las partes de los informes deben ser las de un artículo científico estándar. Es decir, deben tener (al menos) los siguientes componentes o secciones: título, resumen, introducción, metodología, resultados, conclusiones y bibliografía (aunque los nombres pueden

las corridas y vuelva a ejecutarla. Repita esto cuantas veces sea necesario para lograr que el tiempo mayor no sea más grande que 1,5 veces el tiempo menor para cada algoritmo.

²Recuerde que $n \log n < n^{1+\epsilon}$ para $\epsilon > 0$ y n suficientemente grande.

variar a conveniencia). A continuación se describe cada uno de ellos y se dan recomendaciones para facilitar su elaboración.

Título El título debe ser representativo de lo tratado en el trabajo. Evite el uso de títulos interrogativos (por ejemplo: *Cuáles algoritmos de ordenamiento son los mejores*). [1 pto.]

Resumen Como su nombre lo indica, el resumen condensa todo el trabajo en unas pocas líneas (en un párrafo, para ser específico). En el resumen se indican la(s) pregunta(s) a responder, la metodología usada para poder responderla(s), los resultados obtenidos y las conclusiones principales que arrojaron (idealmente, la respuesta a las preguntas). Cada sección del informe se resume aquí en una o dos oraciones para formar *un solo párrafo* que contiene entre cuatro y ocho oraciones. Aunque esta sección aparece de primero en un informe, es la última que se escribe (por razones obvias). (En trabajos en inglés, al resumen se le suele llamar *abstract*). [4 pts.]

Introducción En la introducción se definen las preguntas a responder.³ También se describe brevemente la estrategia a seguir para lograrlo. En esta sección suele predominar el texto y las citas bibliográficas.⁴ [5 pts.]

Metodología En la metodología se describen los detalles de los experimentos que permite responder a las preguntas planteadas en la introducción.⁵ En esta sección abundan las descripciones detalladas de los procedimientos. [10 pts.]

Resultados En esta sección se muestran los resultados obtenidos al realizar los experimentos. Aquí suelen presentarse cuadros y figuras. [21 pts.; ver detalle en sección 3].

Conclusiones En esta sección se establecen las conclusiones arrojadas por los resultados.⁶ Aquí vuelve a dominar el texto. [5 pts.]

Bibliografía En esta sección aparece la información de las fuentes bibliográficas citadas en el texto.⁷ [2 pts.]

4.1.2. Idioma

A sabiendas de la importancia que tiene el dominio del idioma inglés en el desempeño académico del estudiante y en su futuro desempeño profesional, además del idioma español, se acepta y se promueve la escritura de los informes en el idioma inglés. Sin embargo, no se aceptan informes en *spanglish*. Es decir, si se decide hacer el informe

³Para esta tarea la pregunta debería ser si la práctica corresponde con la teoría, es decir, si los algoritmos que en teoría son más rápidos (asintóticamente) también lo son en la práctica.

⁴Para esta tarea posiblemente requiera solo citar el libro de texto del curso.

⁵Aquí debería describir cómo va a demostrar empíricamente que unos algoritmos son mejores que otros. Debe describir las pruebas que va a realizar para demostrarlo.

⁶Aquí se espera que se indique cuáles son los mejores algoritmos para resolver la tarea en cuestión.

⁷Para esta tarea, posiblemente solo necesite citar el libro de texto.

en español se debe procurar usar este idioma al máximo (mucha gente pide prestados términos al inglés innecesariamente, por ejemplo: *merge sort*, *random* y *byte*). Si se usan términos en lenguas extranjeras, estos deben escribirse en cursiva (itálica).

4.1.3. Formato

Los informes *deben* seguir el formato para artículos de revistas (*journals*) usado por el Institute of Electrical and Electronics Engineers para sus publicaciones TRANSACTIONS (excepto las revistas TRANSACTIONS ON MAGNETICS y TRANSACTIONS ON DIELECTRICS, que usan otros formatos). El objetivo es que el estudiante, además de aprender a escribir informes técnicos y científicos, viva la experiencia de tener que someterse a un formato preestablecido (práctica común no solo en la academia sino también en el ámbito empresarial). Las especificaciones del formato y las plantillas para L^AT_EX y Word están disponibles en la página https://www.ieee.org/publications_standards/publications/authors/author_templates.html. L^AT_EX provee también una plantilla con este formato. (Vaya a Archivo▷ Nuevo desde plantilla y seleccione IEEEtran-Journal.lyx. ¡Cuidado!... algunas plantillas tienen un nombre parecido pero su formato es distinto).

4.1.4. Estilo

El docente de este curso promueve el uso de buenas prácticas de estilo en la redacción del informe. Para esto premia con *puntos extra* a los estudiantes que siguen reglas estándar en cuanto a tipografía, entre ellas: uso correcto de la cursiva en entornos matemáticos, uso correcto de los decimales, uso de figuras y cuadros «flotantes» y de referencias a ellos, homogeneidad en los tipos de letra, uso de cursiva para palabras en lengua extranjera y uso de división de palabra al final del renglón para evitar espacio excesivo entre las palabras (en esto suelen tener ventaja los usuarios de L^AT_EX y L^AT_EX con respecto a los usuarios de Word) [aprox. 15 pts. extra].

El estudiante que desee obtener puntos extra por estilo debe llenar y entregar el formulario «Lista de chequeo de estilo - Tarea 1». Este formulario es una hoja de cálculo que automáticamente calcula el puntaje obtenido con base en las reglas que el estudiante indica haber cumplido, ya sea marcándolas (en Numbers) o poniendo un uno o un «verdadero» (en Excel o LibreOffice).

Ejemplos de reglas son las siguientes:

1. Las variables, constantes y funciones matemáticas *de una letra* se escriben en cursiva o inclinadas (*i*, *n*, *x*). Todo lo demás en redonda o sin inclinación (log, máx, cos). Los paréntesis nunca se inclinan.
2. Las páginas están numeradas (excepto quizá la primera).

Para poder afirmar que cumplió una regla, el estudiante debe haberla cumplido en al menos un 90 % de los casos. Por ejemplo, para tener derecho a marcar la regla 2, al menos el 90 % de las páginas deben estar numeradas (excepto quizá la primera). *Si afirma haber cumplido una regla y no la cumplió en el 90 % de los casos, no solo no se le darán los*

puntos sino que además se le castigará con la misma cantidad de puntos (el valor se hará negativo en la hoja de cálculo).

4.2. Código

El código debe escribirse en el lenguaje de programación C++, tomando como base las plantillas publicada junto a este enunciado. Los encabezados de las funciones en la plantilla *no deben ser alterados*, ya que los asistentes usarán un guion (*script*) para revisar la tarea y cualquier cambio en la interfaz de los métodos afectará la compilación. (Sin embargo, se pueden agregar métodos privados y llamarlos dentro de los métodos definidos en la plantilla si fuera necesario). Se debe usar el lenguaje en su versión estándar para evitar problemas de compilación. En particular, el código debe poder ser compilado usando g++ y las librerías estándar de C++. *Si el código no compila, no ordena correctamente o no encuentra la mediana, recibirá una nota de cero.* Para determinar si el algoritmo ordena correctamente se correrá la siguiente prueba:

```
1  for(int i=1; i<n; i++)
2      if( A[i] < A[i-1] )
3          cout << "¡Falló!";
```

Para determinar si encuentra la mediana, se ordenará el arreglo y se comparará el elemento del medio (± 1) con el devuelto por su algoritmo. Si está por debajo del más bajo o por encima del más alto de esos tres elementos se considerará incorrecto. La puntuación de cada algoritmo se especifica en la sección 2.

Si cometió errores en la primera parte de la tarea, se le insta a corregirlos en la segunda parte; sin embargo, no se le reconocerán de vuelta los puntos perdidos.

5. Forma de entrega

La tarea debe entregarse en un archivo comprimido antes de la fecha y hora especificadas, por medio de la plataforma MEDIACIÓN VIRTUAL. La tarea se puede dejar en estado «borrador» (no es necesario «enviarla a revisión»). Es responsabilidad del estudiante verificar que la plataforma haya recibido la tarea y que esté intacta (bajando la tarea y verificando la integridad del comprimido). En caso de que haya múltiples entregas, se considerará solamente la *última*, y si esta se entregó después de la fecha y hora límites, se aplicará la penalización acordada en la CARTA AL ESTUDIANTE: *la calificación recibida por tareas tardías no será más alta que la calificación recibida por cualquiera de los estudiantes que haya entregado la tarea a tiempo.*

Si tiene problemas para subir la tarea y el plazo está cerca de cumplirse, envíela al asistente (sergio.ortegacarpio@ucr.ac.cr) con copia al profesor (arturo.camacho@ecci.ucr.ac.cr). Favor poner en el asunto «Tarea de Estructuras de Datos». En caso de que se reciban múltiples versiones por correo electrónico, se revisará solamente la *primera* enviada (para desalentar el envío de múltiples correos electrónicos). La política

de penalización por entrega tardía mencionada en el párrafo anterior también aplica a entregas por correo electrónico.

6. Revisión de la tarea

El código lo revisa el asistente mediante observación y casos de prueba, pero los informes los revisa el docente en presencia del estudiante. Para esto debe sacar una cita de 15 minutos con el docente usando el sitio de citas Calendly (cuyo enlace está en la plataforma del curso). En la revisión el docente comenta sobre las deficiencias observadas en el documento. Se recomienda que preste atención a las recomendaciones para que las ponga en práctica en futuros informes. Dado que el tiempo de revisión es reducido y el ritmo es rápido, *se sugiere que la grabe la conversación* para que no olvide ningún detalle (cualquier teléfono moderno tiene aplicaciones para ello o se pueden conseguir en tiendas de aplicaciones).