

**CI-0116 Análisis de Algoritmos y Estructuras de Datos**  
**I ciclo 2024**  
**Prof. Dr. Allan Berrocal Rojas**

**Tarea II**

**Índice**

<b>1</b>	<b>Indicaciones generales</b>	<b>2</b>
<b>2</b>	<b>Objetivo</b>	<b>2</b>
<b>3</b>	<b>Estructuras de Datos a Implementar</b>	<b>2</b>
<b>4</b>	<b>Código Fuente de los Algoritmos</b>	<b>3</b>
4.1	Lista Enlazada . . . . .	3
4.2	Árbol de Búsqueda Binaria . . . . .	3
4.3	Árbol Rojinegro . . . . .	4
4.4	Tabla de Dispersión . . . . .	4
<b>5</b>	<b>Recolección de Datos</b>	<b>4</b>
5.1	Lista Enlazada . . . . .	4
5.2	Árbol de Búsqueda Binaria . . . . .	5
5.3	Árbol Rojinegro . . . . .	6
5.4	Tabla de Dispersión . . . . .	6
5.5	Comparación de Resultados . . . . .	6
<b>6</b>	<b>Reporte de Resultados</b>	<b>7</b>
<b>7</b>	<b>Descripción del Producto a Entregar</b>	<b>7</b>
<b>8</b>	<b>Valor de cada entrega</b>	<b>8</b>



---

## 1. Indicaciones generales

Esta tarea es individual y debe entregarla en la fecha que se indica en la plataforma de Mediación Virtual. Si lo desea, el(la) estudiante puede colaborar con sus compañeros(as) pero de manera profesional. Es decir, antes de discutir los problemas de la tarea con sus compañeros(as), primero dedique al menos un día de trabajo individual y formule sus propias respuestas para la tarea. Aunque colabore con algún compañero(a), todo el material que desarrolla y entrega en su tarea debe ser hecho por usted mismo(a). Cada estudiante debe estar en capacidad de defender correctamente las respuestas de su trabajo de manera verbal si la persona docente o la persona asistente del curso formula una pregunta durante la revisión del trabajo. Si la persona docente sospecha que la tarea es parcial o totalmente producto de plagio, la tarea recibirá una nota de cero y se iniciará el debido proceso estipulado en el [Reglamento de Orden y Disciplina de los Estudiantes Universidad de Costa Rica](#). En el reporte a entregar, debe aparecer la referencia toda fuente externa de producción intelectual que utilizó para durante el desarrollo de este trabajo. Recuerde que se considera plagio presentar como propio, material parcial o totalmente creado por otras personas u obtenido de fuentes de información, como por ejemplo de libros, fuentes de Internet, sistemas de generación de texto basados en modelos de aprendizaje automático y profundo, entre otras.

## 2. Objetivo

El objetivo de la tarea es implementar algunas de las estructuras de datos estudiadas en el curso con el objeto de realizar experimentos con dichas estructuras y sus correspondientes operaciones. En la fase experimental se debe recolectar información sobre el desempeño de los algoritmos. Se espera que la persona estudiante sea capaz de analizar los resultados, reflexionar y comparar la eficiencia teórica de las estructuras de datos y sus algoritmos con la eficiencia observada durante los experimentos.

## 3. Estructuras de Datos a Implementar

Las estructuras de datos a implementar en la **primera parte** son la lista enlazada (*Linked List*) y el árbol de búsqueda binaria (*Binary Search Tree*). En la **segunda parte** debe implementar un árbol rojinegro (*Red-Black Tree*) y una tabla de dispersión (*Hash Table*).



## 4. Código Fuente de los Algoritmos

El código debe escribirse en el lenguaje de programación C++, tomando como base las plantillas publicadas junto a este enunciado «`l1list.h`, `bstree.h`, `rbtree.h`, `chast.h`». Los encabezados de las funciones en las plantillas **no deben ser alterados**, ya que la persona asistente del curso usará 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 bibliotecas estándar de C++. *Si el código no compila o no funciona correctamente, recibirá una nota de cero.*

Agregue los archivos «`*.h`» una función con el nombre `ImprimirDatosDeTarea()` que devuelva una hilera con su número de carné, identificador de la tarea, e identificador de la etapa [**0.5 pts.**]. Por ejemplo, si su número de carné es B12345, esta función deberá devolver la siguiente cadena: `b12345 Tarea 1 Etapa X`. Donde  $X = 1$  en la primera entrega y  $X = 2$  en la segunda entrega.

Debe programar siguiendo las mejores prácticas estudiadas en la carrera incluyendo documentación de todas las clases y métodos que implementa. Para facilitar su labor, puede utilizar herramientas como [doxygen](#).

### 4.1. Lista Enlazada

Implemente la clase *lista enlazada con nodo centinela* usando la plantilla «`l1list.h`» y los siguientes métodos:

- constructor [**1 pto.**]
- destructor [**2 pts.**]
- insertar [**3 pts.**]
- buscar [**3 pts.**]
- borrar [**3 pts.**]

### 4.2. Árbol de Búsqueda Binaria

Implemente la clase *árbol de búsqueda binaria*, usando la plantilla «`bstree.h`» y los siguientes métodos:

- constructor [**1 pto.**]
- destructor [**2 pts.**]
- insertar [**3 pts.**]
- borrar [**3 pts.**]
- buscar llave de manera recursiva [**3 pto.**]
- buscar llave de manera iterativa [**3 pts.**]



- buscar el máximo [1 pts.]
- buscar el mínimo [1 pts.]
- obtener el sucesor de un nodo [2 pts.]
- recorrido del árbol en orden [3 pts.]

### 4.3. Árbol Rojinegro

Implemente la clase *árbol rojinegro*, usando la plantilla «*rbtree.h*» y los siguientes métodos:

- constructor [1 pto.]
- destructor [2 pts.]
- insertar [6 pts.]
- borrar [6 pts.]
- buscar llave de manera recursiva [2 pto.]
- buscar llave de manera iterativa [2 pts.]
- buscar el máximo [1 pts.]
- buscar el mínimo [1 pts.]
- obtener el sucesor de un nodo [2 pts.]
- recorrido del árbol en orden [3 pts.]

### 4.4. Tabla de Dispersión

Implemente la clase *tabla de dispersión*, usando la plantilla «*chasht.h*» y los siguientes métodos:

- constructor [2 pts.]
- destructor [2 pts.]
- insertar [4 pts.]
- buscar [4 pts.]
- borrar [4 pts.]

Las colisiones deben ser resueltas mediante encadenamiento utilizando una *lista enlazada*. [2 pts.] extra si utiliza la clase que programó en la sección 4.1, modificándola para que sea una lista doblemente enlazada. La función de dispersión a utilizar es  $h(k) = k \bmod m$ , donde  $m$  es el tamaño de la tabla (más detalles sobre esto adelante) y  $\bmod$  representa la operación módulo.

## 5. Recolección de Datos

Para cada una de las estructuras de datos que implementa, debe realizar una serie de operaciones y registrar los resultados que posteriormente utilizará para hacer análisis cuantitativos y discutir sobre las diferencias y similitudes entre la teoría y los resultados observados.

### 5.1. Lista Enlazada

#### 5.1.1. Inserción Aleatoria

1. Inserte en una lista vacía  $n = 1\,000\,000$  nodos cuyas llaves sean enteros seleccionados aleatoriamente en el rango  $[0, 2n)$  es decir,  $0 \leq x < 2n$ . [1 pto.]



2. Busque en la lista un total de  $e = 10\,000$  elementos cuyo valor o llave es seleccionado al azar en el mismo rango  $[0, 2n)$  registrando el tiempo de duración del total de búsquedas realizadas incluyendo los casos en que el elemento buscado está en la lista, y los casos en que no lo está. [1 pto.]

### 5.1.2. Inserción Ordenada

1. Inserte en una lista vacía las llaves  $0, 1, \dots, n - 1$ , en ese orden con  $n = 1\,000\,000$ . [1 pto.]
2. Busque en la lista un total de  $e = 10\,000$  elementos seleccionados al azar en el rango  $[0, 2n)$  registrando el tiempo de duración del total de búsquedas realizadas incluyendo los casos en que el elemento buscado está en la lista, y los casos en que no lo está. [1 pto.]

### 5.1.3. Mediciones

1. Repita los pasos anteriores al menos tres veces registrando en un cuadro el tiempo de duración de cada corrida y su respectivo promedio<sup>1</sup> [1 pts.]
2. Para cada uno de los escenarios (inserción aleatoria y ordenada), cree un gráfico con los tiempos promedio contra el escenario. Asegúrese de indicar en cada gráfico las unidades de tiempo utilizadas. Indique si en alguno de los dos casos (inserción de números aleatorios o inserción de números secuenciales) el tiempo de duración fue sustancialmente mayor que en el otro (más del doble), e indique si esto corresponde a lo esperado. Explique. [2 pts.]

## 5.2. Árbol de Búsqueda Binaria

1. Repita los pasos de la sección 5.1.1 con la misma cantidad de llaves pero ahora usando el árbol de búsqueda binaria como estructura de datos. [2 pts.]
2. Repita los pasos de la sección 5.1.2 con la misma cantidad de llaves pero ahora usando el árbol de búsqueda binaria como estructura de datos. [2 pts.]

Para este paso, note que insertar  $n$  llaves ordenadas en un árbol de búsqueda binaria puede tomar demasiado tiempo si  $n$  es grande —¿por qué?—. Para evitar la larga espera, considere crear un método que produzca un árbol idéntico al que se crearía si se insertaran en él las llaves  $0, 1, \dots, n - 1$ , en ese orden, pero sin usar el método de inserción —¿cómo?—. [1 pto.]

<sup>1</sup>Si 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 realizaba estas operaciones. En tal caso elimine una de 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 corrida.



3. Repita los pasos de la sección 5.1.3 realizando un gráfico comparativo de los resultados de las búsquedas pero utilizando ahora el árbol de búsqueda binaria como estructura de datos de referencia. [3 pts.]

### 5.3. Árbol Rojinegro

1. Repita los pasos de la sección 5.1.1 con la misma cantidad de llaves pero ahora usando el árbol rojinegro como estructura de datos. [2 pts.]
2. Repita los pasos de la sección 5.1.2 con la misma cantidad de llaves pero ahora usando el árbol rojinegro como estructura de datos. [2 pts.]
3. Repita los pasos de la sección 5.1.3 realizando un gráfico comparativo de los resultados de las búsquedas pero utilizando ahora el árbol rojinegro como estructura de datos de referencia. [3 pts.]

### 5.4. Tabla de Dispersión

1. Repita los pasos de la sección 5.1.1 con la misma cantidad de llaves pero ahora usando como estructura de datos una tabla de dispersión de tamaño  $m$  cuyo factor de carga sea  $\alpha = 1$  (es decir,  $m = n$ ). [2 pts.]
2. Repita los pasos de la sección 5.1.2 con la misma cantidad de llaves pero ahora usando como estructura de datos una tabla de dispersión de tamaño  $m$  cuyo factor de carga sea  $\alpha = 1$  (es decir,  $m = n$ ). [2 pts.]
3. Repita los pasos de la sección 5.1.3 realizando un gráfico comparativo de los resultados de las búsquedas pero utilizando ahora la tabla de dispersión como estructura de datos de referencia. [3 pts.]

### 5.5. Comparación de Resultados

1. Haga un gráfico con los tiempos de duración promedio para el ejercicio de búsqueda de llaves insertadas de manera aleatoria. Incorpore en el gráfico todas las estructuras de datos implementadas. Identifique si la relación entre las curvas es la esperada. [1 pto. c/cada estructura]
2. Haga un gráfico con los tiempos de duración promedio para el ejercicio de búsqueda de llaves insertadas de manera ordenada. Incorpore en el gráfico todas las estructuras de datos implementadas. Identifique si la relación entre las curvas es la esperada. [1 pto. c/cada estructura]



3. Compare los tiempos de duración de las búsquedas sobre llaves aleatorias e identifique si algunas de las estructuras de datos fueron sustancialmente mejor que otras para este ejercicio. Identifique aquellas que tardaron la mitad o menos de tiempo en realizar las búsquedas. Argumente si esto corresponde con lo esperado.
4. Compare los tiempos de duración de las búsquedas sobre llaves ordenadas e identifique si algunas de las estructuras de datos fueron sustancialmente mejor que otras para este ejercicio. Identifique aquellas que tardaron la mitad o menos de tiempo en realizar las búsquedas. Argumente si esto corresponde con lo esperado.

Sea cuidadoso(a) en esta sección ya que los resultados que genere los va utilizar en el reporte escrito de la entrega.

## 6. Reporte de Resultados

Luego de recolectar datos como se indicó en la sección anterior, usted debe discutir los resultados observados que ya tiene registrados en su hoja de datos.

Para escribir el reporte, debe utilizar como base la plantilla de artículos científicos de la IEEE. Para simplificar el proceso, el docente le brindará una plantilla que puede utilizar en **Microsoft Word** o en **L<sup>A</sup>T<sub>E</sub>X**.

El reporte escrito tiene un valor de [10 pts.] para la primera entrega y [5 pts.] para la segunda entrega. Algunos de los elementos a revisar son: redacción, ortografía, cumplimiento de las secciones solicitadas, calidad y correctitud de los gráficos generados, claridad del análisis, profundidad, pertinencia y correctitud del análisis, integridad del contenido (correcto uso de referencias a material utilizado), correcto uso de la nomenclatura aplicable.

El reporte se debe entregar en **formato PDF únicamente**. NINGÚN OTRO FORMATO SERÁ ACEPTADO.

## 7. Descripción del Producto a Entregar

La tarea tiene dos entregas. Cada entrega contiene un informe escrito en formato PDF y el código fuente de los algoritmos implementados.

En cada entrega debe incluir el código fuente de las estructuras de datos correspondientes así como los resultados obtenidos al realizar las operaciones solicitadas. Primera entrega: lista enlazada



(*Linked List*) y árbol de búsqueda binaria (*Binary Search Tree*). Segunda entrega: árbol rojinegro (*Red-Black Tree*) y tabla de dispersión (*Hash Table*).

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

Entregue su trabajo en su repositorio privado de control de versiones en una estructura de directorios **exactamente** como se muestra a continuación: [0.5 pts.]

- \tareas\_programadas\tp2\entrega1

Coloque aquí la versión del código de la primera entrega así como el documento de reporte en formato PDF.

- \tareas\_programadas\tp2\entrega2

Coloque aquí la versión del código de la segunda entrega así como el documento de reporte en formato PDF.

Asegúrese de utilizar un archivo `.gitignore` para evitar subir al repositorio de control de versiones archivos temporales o autogenerados durante el proceso de compilación. [0.5 pts.]

Es responsabilidad de la persona 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.

Si tiene dudas sobre cómo utilizar el repositorio de control de versiones, busque al asistente del curso o a la persona docente **con antelación** para recibir ayuda. No espere hasta el día de la entrega para solicitar ayuda porque corre el riesgo de que no se le pueda atender.

## 8. Valor de cada entrega

El siguiente cuadro muestra un resumen de los elementos a evaluar en cada una de las dos entregas de la tarea.





Cuadro 1: Resumen de puntajes por entrega

Etapas	Elemento	Puntaje
1	Implementación de lista enlazada	12
	Función <code>ImprimirDatosDeTarea()</code>	0.5
	Datos de ejecución lista enlazada	7
	Implementación de árbol de búsqueda binaria	22
	Función <code>ImprimirDatosDeTarea()</code>	0.5
	Datos de ejecución árbol de búsqueda binaria	8
	Gráficos comparativos sobre búsquedas aleatorias	2
	Gráficos comparativos sobre búsquedas ordenadas	2
	Reporte escrito	10
	Estructura de directorios	0.5
	Buen uso del repositorio y <code>.gitignore</code>	0.5
<b>Total</b>		<b>65 pts.</b>
2	Implementación de árbol rojinegro	26
	Función <code>ImprimirDatosDeTarea()</code>	0.5
	Datos de ejecución árbol rojinegro	7
	Implementación de tabla de dispersión	16
	(opcional) uso de su propia lista en tabla de dispersión	2
	Función <code>ImprimirDatosDeTarea()</code>	0.5
	Datos de ejecución tabla de dispersión	7
	Gráficos comparativos sobre búsquedas aleatorias	2
	Gráficos comparativos sobre búsquedas ordenadas	2
	Reporte escrito	5
	Estructura de directorios	0.5
	Buen uso del repositorio y <code>.gitignore</code>	0.5
<b>Total</b>		<b>67 pts.</b>
		2 (opcional)