

Estructura de Datos - Certamen 1

Profesor: **Eduardo Godoy**

28 de octubre de 2018

Nombre:

Rut:

Paralelo:

Puntaje:

Nota:

Instrucciones:

- El puntaje máximo es 100 puntos.
- Tiempo máximo: 120 minutos.
- El trabajo es **individual**. Cualquier intento de copia, será sancionado según dicta el reglamento de la carrera.

Resultados de aprendizaje a evaluar:

1. Conocer e Implementar algoritmos de ordenamiento y estructuras de datos complejas.

Contenido: Este certamen evalúa los siguientes temas:

Tema	Puntajes	
	Total	Obtenido
Problema 1: Complejidad de algoritmos	30 pts.	
Problema 2: Algoritmos de Ordenamientos	40 pts.	
Problema 3: TDA - Listas Enlazadas.	30 pts.	

1. Problema 1

1. 30pts.

- Analice los siguiente algoritmos y a continuación responda.

A_1 : <pre>#include <stdio.h> int main(){ int n=100000; //10⁵ int a[n]; for(int i=0; i<n; i++){ a[i]=2*i; } for(int i=0; i<n; i++){ printf("%d tiene %d\n",i,a[i]); } return(0); }</pre>	A_2 : <pre>#include <stdio.h> int main(){ int n=1000000; //10⁶ int a[2*n]; for(int i=0; i<2*n; i++){ a[i]=2*i; } return(0); }</pre>
--	--

- a) ¿Cuántas veces itera cada uno de los tres “for”? [5 pts]
- b) ¿Cómo calcula el tiempo de CPU de ejecución de cada algoritmo? Calcule. [10 pts]
- c) ¿Cuál es el tiempo de ejecución en notación big O de cada algoritmo? [10 pts]
- d) ¿Qué algoritmo es más eficiente en términos de complejidad temporal? [5 pts]
- a) Los tiempos se deben sumar en notación big O.82ex [3 pts]
El tiempo total será $2n^3 + O(\frac{1}{2}n^2) = O(2n^3 + \frac{1}{2}n^2)$,82ex [2 pts]
es decir, $O(n^3)$.82ex [5 pts]
- b) Los tiempos se deben sumar en notación big O (o bien quedarme con el máximo).82ex [2 pts]
El tiempo total será $2n^3 + O(\frac{1}{2}n^2) + O(2^n) = O(2n^3 + \frac{1}{2}n^2 + 2^n)$,82ex [2 pts]
es decir, $O(2^n)$.82ex [6 pts]
- c) Calculos:
- a) Cada for de A_1 itera $n = 10^5$ veces. 82ex [2 pts]
El for de A_2 itera $2n = 2 \times 10^6$ veces. 82ex [3 pts]
- b) Puede variar entre cada máquina, pero debe ser el resultado user que retorna la ejecución con el comando time. 82ex [5 pts]
Escribir el resultado obtenido para A_1 y para A_2 . 82ex [5 pts]
- c) Para A_1 es $O(n + n) = O(2n) = O(n)$.82ex [5 pts]
- Para A_2 es $O(2n) = O(n)$.82ex [5 pts]
- d) De lo anterior se concluye que A_1 y A_2 tienen la misma complejidad temporal.82ex [5 pts]

¿Cómo será evaluado en este trabajo?			
Ítem	Logrado	Suficiente	No Logrado
Preguntat 1.	Aplica de forma correcta: 5pts	Aplica parcialmente con menos de 2 errores: 3pts	Aplica de forma incorrecta con 3 errores o más: 0pts
Preguntat 2.	Aplica de forma correcta: 10pts	Aplica parcialmente: 5pts	Aplica de forma incorrecta: 0pts
Preguntat 3.	Aplica de forma correcta 10pts	Aplica parcialmente 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Pregunta 4.	Aplica de forma correcta 5pts	Aplica parcialmente 3pts	Aplica de forma incorrecta 0pts
Total de la sección	30pts	15pts	0pts

Nota: En caso de que el ítem no esté presente, tiene ponderación cero.

2. Problema 2

a) **60pts.** Utilizando la técnica del algoritmo Quick-Sort ordene el siguiente arreglo:

8	5	2	6	8	3	1	4	0	7
---	---	---	---	---	---	---	---	---	---

- Considere como criterio de selección del pivote el extremo el primero desde la izquierda.

	piv	8							
	izq								der
	8	5	2	6	9	3	1	4	0
					izq				der
	7	5	2	6	9	3	1	4	0
									izq
									der
	7	5	2	6	0	3	1	4	9
	7	5	2	6	0	3	1	4	8
	piv	7							
	izq								der
	7	5	2	6	0	3	1	4	
									izq
									der
	4	5	2	6	0	3	1	7	
	piv	4							
	izq								der
	4	5	2	6	0	3	1		
			izq				der		
	1	5	2	6	0	3	4		
					izq	der			
	1	3	2	6	0	5	4		
						izq			
					der				
	1	3	2	0	6	5	4		
	izq				der		izq	der	
	1	3	2	0	4	5	6		
	piv	0							
		izq	der						
	0	3	2	1					
					izq				

			der							
	0	2	3	1						
	0	2	1	3						
	0	2	1							
		2	1							
		1	2							
	0	1	2	3	4	5	6	7	8	9

¿Cómo será evaluado en este trabajo?			
Ítem	Logrado	Suficiente	No Logrado
Conocimiento del algoritmo	Aplica de forma correcta 40 pts	Aplica parcialmente con menos de 3 errores 20 pts	Aplica de forma incorrecta con 3 errores o más 0pts
Total de la sección	40pts	20pts	0pts

Nota: En caso de que el ítem no esté presente, tiene ponderación cero.

3. Problema 3

a) **30pts.**

b) Considere el siguiente pseudocódigo, que define una lista doblemente enlazada de nodos:

```
struct Node {
    data; 23ex// Dato almacenado en el nodo
    next; 23ex// Puntero al nodo siguiente (NULL para el último nodo)
}
struct List {
    Node FirstNode; 23ex// La lista apunta al primer nodo; NULL si está vacía
    Node LastNode; 23ex// La lista apunta al último nodo; NULL si está vacía
}
```

Para insertar un nodo newNode después de un nodo node, se define la función insertAfter:

```
function insertAfter(Node node, Node newNode) {
    newNode.next := node.next;
    node.next := newNode;
}
```

Para insertar un nodo newNode al inicio de la lista list, se define la función insertBeginning:

```
function insertBeginning(List list, Node newNode) {
    newNode.next := list.firstNode;
    list.firstNode := newNode;
}
```

A partir de lo anterior, defina:

- 1) Una función insertEnd, que inserta un nodo newNode al final de list. 76ex [15 pts]
- 2) Una función removeAfter, que elimina un nodo Node. 76ex [15 pts]

a) La función queda definida por:

```
function insertEnd(List list, Node newNode) {
    (list.LastNode).next := newNode;
} 81ex [15 pts]
```

b) La función queda definida por:

```
function removeAfter(Node node) {
    aux := node.next;
    node.next := node.next.next;
    destroy aux;
} 81ex [15 pts]
```

¿Cómo será evaluado en este trabajo?			
Ítem	Logrado	Suficiente	No Logrado
insertEnd.	Aplica de forma correcta 15pts	Aplica parcialmente 7 errores 5pts	Aplica de forma incorrecta 0pts
removeAfter.	Aplica de forma correcta 15pts	Aplica parcialmente 3pts	Aplica de forma incorrecta 0pts
Total de la sección	60pts	30pts	0pts

Nota: En caso de que el ítem no esté presente, tiene ponderación cero.