Estructura de Datos - Certamen 1

Profesor: **Eduardo Godoy**

29 de octubre de 2018

Nombre:

Rut:

Paralelo:

Puntaje:

Nota:

Instrucciones:

- El puntaje máximo es 100 puntos.
- Tiempo máximo: 120 minutos.
- El certamen 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.***
   - Analise los siguiente algoritmos y a continuación responda.

A_1 :

```
#include <stdio.h>
int main(){
  int n=100000; //10^5
  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);
}
```

A_2 :

```
#include <stdio.h>
int main(){
  int n=1000000; //10^6
  int a[2*n];
  for(int i=0; i<2*n; i++){
   a[i]=2*i;
  }
  return(0);
}
```

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]

¿Cómo seré evaluado en este trabajo?			
Ítem	**Logrado**	**Suficiente**	**No Logrado**
Pregunat 1.	Aplica de forma correcta: 5pts	Aplica parcialmente con menos de 2 errores: 3pts	Aplica de forma incorrecta con 3 errores o más: 0pts
Pregunat 2.	Aplica de forma correcta: 10pts	Aplica parcialmente: 5pts	Aplica de forma incorrecta: 0pts
Pregunat 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.

¿Cómo seré 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.*** Considere el siguiente pseudocódigo, que define una lista enlazada de nodos:

```
struct Node {
  data; // Dato almacenado en el nodo
  next; // Puntero al nodo siguiente (NULL para el último nodo)
}
Punteros de la lista {
  Node FirstNode; // La lista apunta al primer nodo; NULL si está vacía
  Node LastNode; // 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`. [15 pts]
2) Una función `removeAfter`, que elimina un nodo `Node`. [15 pts]

¿Cómo seré 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.