

Reglas del juego:

- Tal como se le dijo antes, durante este examen usted no puede pedir ABSOLUTAMENTE nada prestado a sus compañeros, ni hablar con ellos.
- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkuelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- No se resolverá ninguna duda durante el examen, así que si algo no le queda claro y usted requiere hacer algún tipo de supuesto, ESCRÍBALO y de una breve explicación de la razón por la cual lo supone. Tenga presente que un supuesto no puede contradecir lo que se le está diciendo.
- Recuerde que este es el curso de Estructuras de Datos, por lo que en todo momento es FUNDAMENTAL respetar la filosofía de dicha estructura; por lo que no puede hacer ni código ni supuestos que vayan en contra de dicha filosofía.

1. Corrija la siguiente implementación del algoritmo Quicksort. (0.6)

```
public int Particionar (int izq, int der)
{
    int pivote = lista[(der + izq)/2];
    while (izq < der)
    {
        while (pivote >= lista [izq])
            izq++;
        while (pivote <= lista [der])
            der--;

        if (izq < der)
        {
            lista[der] = lista [izq];
            int aux = lista[der];
            lista[izq] = aux;
        }
    }
    return (der);
}
```

```
public void Ordenar(int desde, int hasta)
{
    if (desde < hasta)
    {
        int ind_partición = Particionar (desde, hasta);
        Ordenar (desde, ind_partición);
        Ordenar(ind_partición+1, hasta);
    }
}
```

2. Para el siguiente arreglo de números, muestre paso a paso cómo funcionan los métodos de ordenamiento Shell y Selección. (0.8)

50	8	42	13	72	2	10	0	4	13
----	---	----	----	----	---	----	---	---	----

3. Explique qué es pivote en el método de ordenamiento Quicksort. ¿Por qué se escoge como pivote el elemento ubicado en la mitad del arreglo?. (0.8)
4. Explique claramente las diferencias entre los métodos de búsqueda Secuencial y Binario. (1.0)
5. ¿Qué consideraciones se deben tener en cuenta para escoger un algoritmo de ordenamiento? (0.6)
6. Para el método de transformación de claves: (1.2)
- ¿Qué es una colisión cuando se usa Hashing?
 - Muestre un ejemplo de colisión con una función Hash definida por usted.
 - Explique dos maneras de solucionar las colisiones.

Puntos Opcionales

1. Como se discutió en clase, conceptualmente un árbol binario es un caso especial de un grafo.
Basándose en la implementación vista de grafo en clase, escriba uno o varios métodos que permitan saber si un grafo se puede considerar árbol binario.
Asuma que usted cuenta con un método llamado NúmeroDeCaminos, que retorna el número de caminos que existen entre dos vértices pasados por parámetro. (0.5)
2. Implemente una búsqueda binaria en un arreglo utilizando recursividad. (0.5)