

Algoritmos y Estructuras de Datos

-DEMO- Examen Parcial -PRACTICA- -NOT THE REAL THING-

Conversion de
Notas:
60-63 = 4
64-69 = 5
70-76 = 6
77-83 = 7
84-89 = 8
90-96 = 9
97+ = 10

Nombre:

Legajo:

Instrucciones: Los ejercicios deberían poder completarse en el enunciado mismo, en caso de quedarse sin lugar puede usar alguna hoja auxiliar.

Rúbrica:

Ej	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
Pts	4	6	6	4	10	10	6	10	6	6	6	10	6	10	100

[1] En un Tournament Tree queremos implementar la primitiva `get_second_best()` que devuelve el segundo mayor elemento en el árbol. Indique cuál es el peor caso para esta primitiva en un árbol de 8 elementos. (dibujar el árbol y explicar brevemente)

[2] Indique qué tipo de algoritmo de sort sería más conveniente en cada uno de los siguientes casos:

a) Un vector de enteros que está casi ordenado a excepción de un elemento. _____

b) Un vector de enteros de 16 bits de 4 millones de números completamente desordenados. _____

c) Un vector de 1000 strings de 128 caracteres cada uno. _____

[3] Realice las siguientes operaciones sobre un árbol binario de búsqueda empezando con el árbol vacío. Indique cómo queda el árbol luego de cada paso.

`insert(29),insert(16),insert(10),delete(16),search(10), delete(29)`

[4] Repaso de ordenes Indique el orden de las siguientes operaciones:

- a) Heapdown (mejor caso) _____
- b) Insert sort (mejor caso) _____
- c) Remove-Max en un tournament heap (mejor caso) _____
- d) Search en un árbol binario de búsqueda (mejor caso) _____

[5] Programe una función que indique si un árbol es un árbol binario de búsqueda (la función recibe un puntero a un nodo).

[6] Dado un vector de ints queremos dejar todos los ceros al principio del vector. Diseñe un algoritmo lo más eficiente posible para resolver este problema y analice el Orden del mismo. Puede usar pseudocódigo.

[7] Explique qué hace la siguiente función sobre un vector y de el Orden de la misma.

```
bool mystery(int* vec, int len) {
    for(i=0;i<1000;i++) {
        int pos = random(0,len-2)
        if (vec[pos] > vec[pos+1]) return False;
    }
    return True;
}
```

[8]. Dada una lista simplemente enlazada crear una función que elimine el “n-ésimo” nodo contando desde el último elemento. (por ejemplo si $n=0$ tiene que eliminar el último nodo, si $n=1$ tiene que eliminar el ante-último nodo, etc).

Codificar la función `bool remove_nth_last(struct Node* head)` en $O(n)$.

Muchos puntos extras: Hacer la función recorriendo la lista una sola vez.

[9]. Indique cuál es el orden de los siguientes algoritmos si sabemos cuánto tiempo se tarda en procesar “n” elementos, es decir le damos $T(n)$ y queremos el orden por ejemplo para $T(n) = 3n+4$ es $O(n)$

- a. $T(n) = 0.01n \log_2 n + n(\log_2 n)^2$ _____
- b. $T(n) = 0.003 \log_4 n + \log_2 \log_2 n$ _____
- c. $T(n) = 2n + n^{0.5} + 0.5n^{1.25}$ _____

[10] Indique el Orden de la siguiente porción de código:

```
int a=0;
int i=N;
while(i>0) {
    a+=i;
    i=i/2;
}
```

[11]. Considere la siguiente variante sobre una pila:

```
create() // crea una pila vacía
pop() //remueve el tope de la pila
top() // obtiene pero no remueve el tope de la pila
push(elem) // agrega el numero x en la pila pero solo si es mas grande que el
tope actual, si no es mas grande que el tope actual hace pops hasta poder
insertar (los elementos popeados se descartan).
```

- a) Indique el estado de la pila luego de hacer las siguientes operaciones:
push(3) push(8) push(15) push (16) push(9) pop() top() pop()
- b) Indique el orden de pop() _____
- c) Indique el orden amortizado de push(elem) _____

[12] Usando únicamente una pila escriba una función que determine si un string es un palíndromo o capicúa. Por ejemplo “neuquen” es un palíndromo, “mendoza” no.

[13] Realizar las siguientes operaciones sobre un árbol 2-3-4 y mostrar el árbol luego de cada una:

insert(4), insert(7), insert(8), insert(11), delete(7), delete(4)

[14] Programar una función que busque si un cierto número existe en una lista circular.