

Prueba 4

☐ Estructura de Datos

☐ Complejidad Computacional

NOMBRE:

NRO.MATRICULA

Tablas Hash

1. El tiempo necesario para encontrar un ítem en una tabla hash es: _____ [1 pto.]

 $O(1)$

2. Explique que hace una Función Hash? [2 ptos.]

 Transforma un rango de valores en un rango de índices

3. ¿Como se llama el procedimiento que utiliza la siguiente posición disponible en el arreglo después de haber encontrado la posición de inserción original ocupada? [2 ptos.]

 Prueba lineal

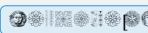
4. Los Clusters Secundarios se forman porque [1 pto.]

- Los índices de muchos valores caen la misma posición.
- Son insertados muchos elementos con el mismo valor.
- La Función Hash no es muy eficiente.
- El largo de los pasos en la secuencia de prueba es el mismo.

 μ

5. Cuando se utiliza Folding como función Hash, el número de dígitos en cada grupo esta

relacionado con _____. [2 ptos.]

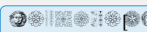
 el tamaño del arreglo

6. En el Encadenamiento Separado (Separate Chaining), el tiempo para insertar un nuevo ítem [1 pto.]

- Es proporcional al factor de carga.
- Es proporcional al número de ítems en la tabla.
- Es proporcional al número de listas.
- Es proporcional al numero de celdas llenas en el arreglo.

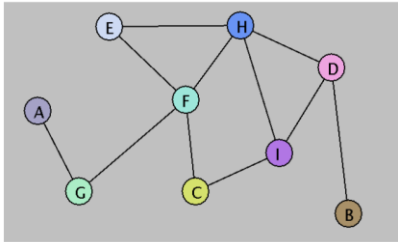
 α

7. ¿Si tengo que almacenar 10.000 elementos en un arreglo utilizando hashing, ¿qué tamaño debiera tener el arreglo? Justifique su respuesta [2 ptos.]

 Transforma un rango de valores en un rango de índices

Grafos

8. Partiendo del nodo "A", ejecute una **Búsqueda en Profundidad** y la **Búsqueda en Anchura** en el grafo. [6 ptos.]



	A	B	C	D	E	F	G	H	I
A	0	0	0	0	0	0	1	0	0
B	0	0	0	1	0	0	0	0	0
C	0	0	0	0	0	1	0	0	1
D	0	1	0	0	0	0	0	1	1
E	0	0	0	0	0	1	0	1	0
F	0	0	1	0	1	0	1	1	0
G	1	0	0	0	0	1	0	0	0
H	0	0	0	1	1	1	0	0	1
I	0	0	1	1	0	0	0	1	0

Profundidad: A G F C I D B H E
Anchura: A G F C E H I D B

Profundidad :

Anchura :

9. ¿Cuál es la Lista de **Adyacencia** asociada a la siguiente matriz? [2 ptos.]

	A	B	C	D
A	0	1	0	0
B	1	0	1	1
C	0	1	0	0
D	0	1	0	0

A:B, B:A->C->D, C:B, D:B

10. El **Árbol de Recubrimiento Mínimo** es un grafo en donde [1 pto.]

- El número de aristas que conectan todos los vértices es el menor posible.
- El número de vértices es igual al número de aristas.
- Los vértices innecesarios son removidos.
- Cada combinación de dos vértices es conectada con el número mínimo de aristas.

un árbol

11. Nombre un tipo de grafo que no contenga ciclos [1 pto.]

un árbol

12. Nombre una aplicación de la **Búsqueda en Profundidad** y la **Búsqueda en Amplitud** [2 ptos.]

- a) b)

Simulación de juegos y buscar el camino más corto entre dos vértices.