

Comenzado el	martes, 1 de diciembre de 2020, 17:00
Estado	Finalizado
Finalizado en	martes, 1 de diciembre de 2020, 18:59
Tiempo empleado	1 hora 58 minutos

Pregunta 1

Finalizado

Puntúa como 1,00

A partir del siguiente bloque de código:

```
set<int> S;
S.insert(-1);
S.insert(4);
S.insert(-2);
S.insert(7);
auto p = S.find(-1);
```

presente mediante un croquis las estructuras de datos presente en memoria luego de su ejecución. Considere la implementación de conjuntos por medio de árboles binarios de búsqueda.

Se adjunta imagen.

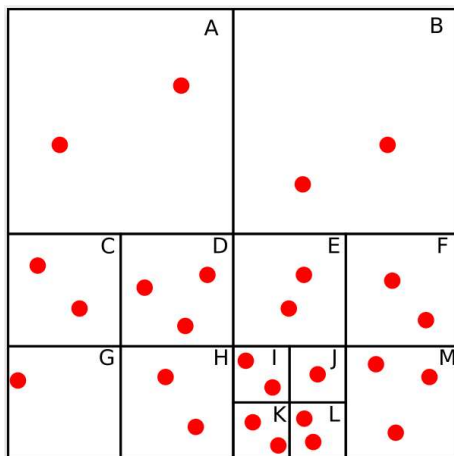


Pregunta 2

Finalizado

Puntúa como 1,00

La Figura presenta una escena de un simulador donde en un dominio cerrado se encuentran partículas que se mueven y rebotan (colisionan) unas contra otras.



Suponga que se dispone de una estructura "partícula" que almacena la información acerca de la posición de cada una de ellas, y siendo que una colisión se detecta cuando dos partículas ocupan parcialmente el mismo espacio. A partir de la figura esquemática ue se dá como ayuda, ¿Que estructuras de datos utilizaria para almacenar las partículas y como determinaría de forma eficiente las colisiones? Seleccione las opciones correctas (las incorrectas descuentan puntos)

Seleccione una o más de una:

- ☒ a. Utilizaría un diccionario TDA que permita hashear la coordenada de la partícula a una región fija en el espacio. Solo busco colisiones de partículas pertenecientes al mismo bucket.
- ☐ b. Un list<particle> desordenado y para encontrar colisiones un doble bucle por todas las partículas.
- ☐ c. Utilizaría un list<particle> para almacenar todas las partículas. Esta lista estará ordenado por distancia al (0,0) para facilitar luego la evaluación de las colisiones.
- ☐ d. Es necesario utilizar un procedimiento recursivo para garantizar que la prueba de colisión sea exhaustiva
- ☒ e. Utilizaría un árbol cuaternario que subdivida telescópicamente el espacio en forma dinámica y cuyas hojas sean set<particle>. Solo busco colisiones entre partículas del mismo set.
- ☐ f. La estructura de datos deberá particionar el espacio de tal manera de reducir las pruebas de colisión

Pregunta 3

Finalizado

Puntúa como 1,00

Dados los enteros **{5,-8,-13,6,42,-10,37,16}** insertarlos, en ese orden, en un **árbol binario de búsqueda (ABB)**. Mostrar las operaciones necesarias para eliminar los elementos 5, -10 y -13 en ese orden.

Se adjunta imagen.

**Pregunta 4**

Finalizado

Puntúa como 1,00

Insertar los enteros **{1, 11, 19, 23, 21, 7, 9, 3, 11}** en una tabla de dispersión cerrada con **B=10** cubetas, con función de dispersión **$h(x) = x \% B$** y redistribución lineal. Luego eliminar el elemento 11 e insertar los elementos 31 y 24, en ese orden. Mostrar la tabla resultante.

**Pregunta 5**

Finalizado

Puntúa como 1,00

Dados los caracteres siguientes con sus correspondientes probabilidades, construir el código binario utilizando el algoritmo de Huffman y encodar la palabra **HOTSALE**, $P(H) = 0.01$, $P(O) = 0.50$, $P(T) = 0.20$, $P(N) = 0.01$, $P(B) = 0.02$, $P(S) = 0.04$, $P(E) = 0.10$, $P(A) = 0.10$, $P(L) = 0.02$. Calcular la longitud promedio del código obtenido.

Adjuntar captura del árbol de Huffman.

Se adjunta imagen.

**Pregunta 6**

Finalizado

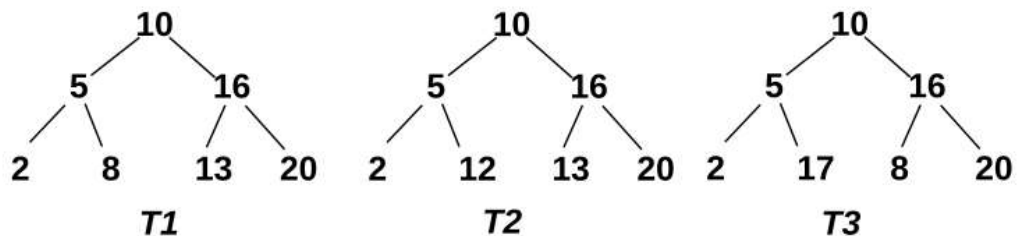
Puntúa como 1,00

Dados los enteros **{0,5,3,3,4,3,6,9,11,2,1}** ordenarlos por el método de **clasificación rápida (quick-sort)**. En cada iteración indicar el pivote y mostrar el resultado de la partición. Utilizar la estrategia de elección del pivote tomando el mayor de los dos primeros elementos distintos.

**Pregunta 7**

Finalizado

Puntúa como 1,00



Cual de los siguientes árboles es un Arbol Binario de Busqueda (ABB)?

Seleccione una:

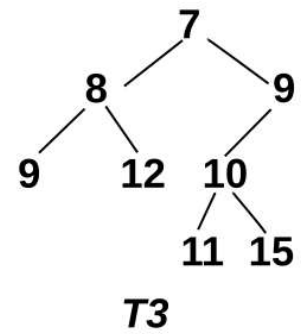
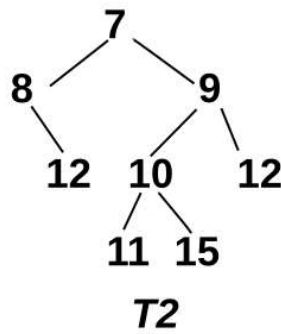
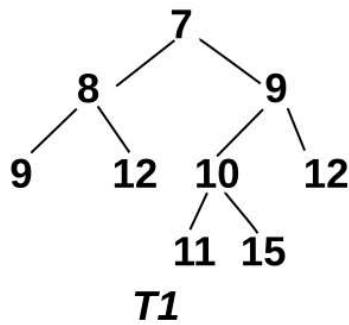
- ☐ a. T3
- ☒ b. T1
- ☐ c. T2

Pregunta 8

Finalizado

Puntúa como 1,00

De los siguiente árboles binarios, cuál es un "Arbol Binario Lleno"?



Seleccione una:

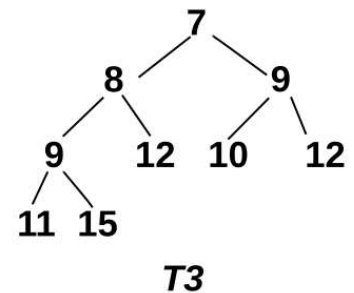
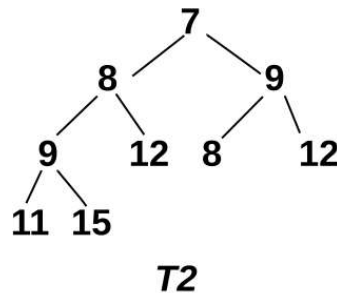
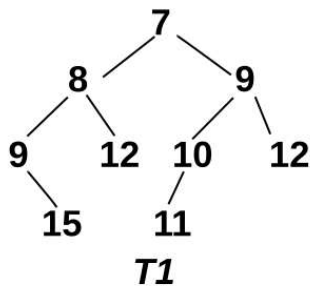
- ☐ a. T3
- ☐ b. T2
- ☒ c. T1

Pregunta 9

Finalizado

Puntúa como 1,00

Dados los siguientes árboles binarios, cuál de ellos satisface las condiciones para ser un "montículo minimal"?



Seleccione una:

- ☒ a. T3
- ☐ b. T2
- ☐ c. T1

Pregunta 10

Finalizado

Puntúa como 1,00

Dado el vector (6,-3,4,2,5,-4,4,-2,3), se aplican 4 algoritmos de ordenamiento (ALG1-ALG4) **con respecto al valor absoluto** y se obtienen los correspondientes vectores de salida. Cada uno de los algoritmos puede ser **válido o no** (el vector resulta estar ordenado), y en caso de ser válido puede ser **estable o no**.

- ALG1 -> (-4,-3,-2,2,3,4,4,5,6)
- ALG2 -> (2,-2,-3,3,-4,4,4,5,6)
- ALG3 -> (-2,2,-3,3,-4,4,4,5,6)
- ALG4 -> (2,-2,-3,3,4,-4,4,5,6)

Seleccione una:

- ☐ a. ALG1, ALG2 y ALG4 son ordenamientos válidos con respecto a valor absoluto, pero no estables. ALG3 es válido y estable.
- ☐ b. ALG1 no es un ordenamiento válido con respecto a valor absoluto. ALG2, ALG3, y ALG4 son válidos y estables.
- ☐ c. ALG1 no es un ordenamiento válido con respecto a valor absoluto. ALG2 y ALG4 son válidos pero no estables. ALG3 es válido y estable.
- ☒ d. ALG1 no es un ordenamiento válido con respecto a valor absoluto. ALG2 y ALG3 son válidos pero no estables. ALG4 es válido y estable.
- ☐ e. ALG1 no es un ordenamiento válido con respecto a valor absoluto. ALG3 y ALG4 son válidos pero no estables. ALG2 es válido y estable.

◀ TPL 3

lra... ▼

TPL 1 - RECUPERATORIO ▶