

Estructuras de Datos Avanzadas

Adriana Ramirez Viguera
Fhernanda Montserrat Romo Olea
Marco Antonio Velasco Flores

Tarea 1

Natalia Abigail Pérez Romero

FACULTAD DE CIENCIAS

Semestre 2024-1

Entrega: 16 Octubre 2023 - 11:59 PM

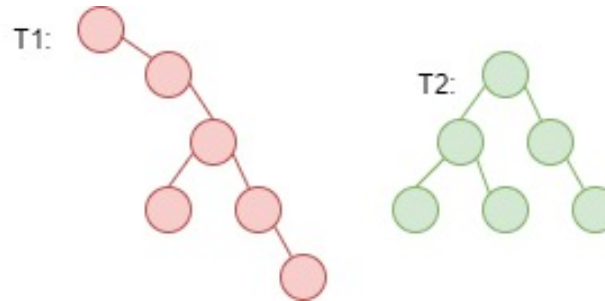
Pregunta 1

Decimos que un árbol binario de búsqueda T_1 puede ser RIGHT-CONVERTED a un árbol binario de búsqueda T_2 si es posible obtener T_2 de T_1 por a través de una serie de llamadas a la operación RIGHT-ROTATE.

- Da un ejemplo de dos árboles T_1 y T_2 tal que T_1 no pueda ser RIGHT-CONVERTED en T_2 .
- Demuestra que si un árbol T_1 puede ser RIGHT-CONVERTED a T_2 , entonces T_1 puede ser RIGHT-CONVERTED usando $O(n^2)$ operaciones RIGHT-ROTATE.

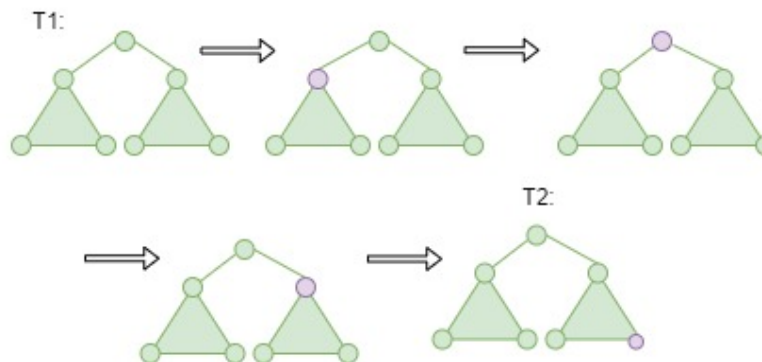
Respuesta

- Da un ejemplo de dos árboles T_1 y T_2 tal que T_1 no pueda ser RIGHT-CONVERTED en T_2 . T_1 es un árbol inclinado hacia la derecha y T_2 es un árbol balanceado de forma que la única forma de transformar



- Demuestra que si un árbol T_1 puede ser RIGHT-CONVERTED a T_2 , entonces T_1 puede ser RIGHT-CONVERTED usando $O(n^2)$ operaciones RIGHT-ROTATE.

Sea u un vértice en T_1 el cual tiene un vértice u' el cual tiene un vértice u' en T_2 de forma que después de un número de operaciones RIGHT-ROTATE u se encuentra en la posición de u' , esto le tomara en el peor de los casos n (el número de vértices) por que ocupará el lugar de todos los vértices en T_1 antes de llegar a la posición de u' . En la figura siguiente podemos observar un ejemplo del movimiento.



Lo anterior es verdad para todos los n vértices por ende si un árbol T_1 puede ser RIGHT-CONVERTED a T_2 , entonces T_1 puede ser RIGHT-CONVERTED usando $O(n^2)$ operaciones RIGHT-ROTATE.

Pregunta 2

Muestra que dado un conjunto T de n nodos x_1, x_2, \dots, x_n con valores y prioridades distintas, el árbol treap asociado a T es único. Hint: utiliza inducción sobre n .

Respuesta

Demostración por contradicción: Suponer que para T tenemos más de un árbol treap asociado. Entonces G y G' son árboles asociados a T , entonces $V(G)$

Pregunta 3

Se pueden utilizar las estructuras de búsqueda de rangos ortogonales para determinar si un punto particular (a, b) está en un conjunto dado, haciendo una consulta al rango $[a : a] \times [b : b]$.

1. Prueba que hacer una consulta así en un árbol KD toma tiempo $O(\log n)$.
2. ¿Cuál es la complejidad para una consulta así en un árbol de rangos?

Respuesta

<Tu respuesta aquí>

Pregunta 4

Describe una secuencia de accesos a un árbol splay T de n nodos, con $n \geq 5$ impar, que resulte en T siendo una sola cadena de nodos en la que el camino para bajar en el árbol alterne entre hijo izquierdo e hijo derecho.

Respuesta

<Tu respuesta aquí>

Pregunta 5

Describe cómo modificar una *skip-list* L para poder realizar las siguientes dos operaciones en tiempo esperado $O(\log n)$:

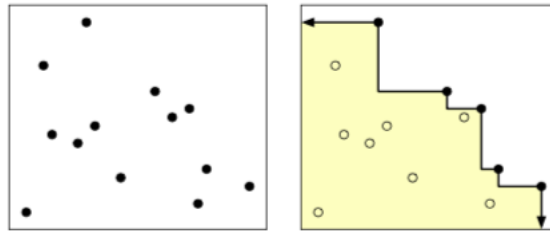
- Dado un índice i , obtener el elemento de L en la posición i .
- Dado un valor x , obtener la cantidad de elementos en L menores a x .

Respuesta

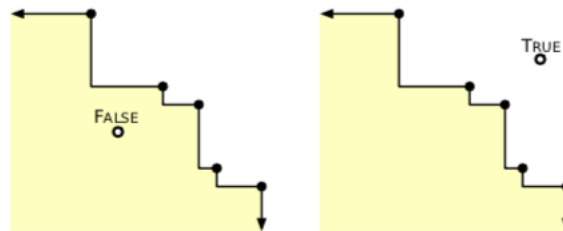
<Tu respuesta aquí>

Pregunta 6

Let P be a set of n points in the plane. The staircase of P is the set of all points in the plane that have at least one point in P both above and to the right.



1. Describe an algorithm to compute the staircase of a set of n points in $O(n \log n)$ time.
2. Describe and analyze a data structure that stores the staircase of a set of points, and an algorithm ABOVE? (x, y) that returns TRUE if the point (x, y) is above the staircase, or FALSE otherwise. Your data structure should use $O(n)$ space, and your ABOVE? algorithm should run in $O(\log n)$ time.



Respuesta

<Tu respuesta aquí>

Pregunta 7

Sea S un conjunto de n segmentos de línea sin cruces entre ellos. Queremos responder rápidamente a consultas del tipo: dado un punto p encontrar al primer segmento en S por el que pasa el rayo vertical con origen en p y dirección hacia arriba. Da una estructura de datos para resolver este problema. Acota el tiempo de consulta y el espacio requerido por tu estructura. ¿Cuál es el tiempo de pre-procesamiento?

Respuesta

<Tu respuesta aquí>

Pregunta 8

En algunas aplicaciones solo nos interesa el número de puntos que caen dentro de un rango y no reportar cada uno de ellos. En este caso nos gustaría evitar el término $O(k)$ en el tiempo de consulta.

1. Describe cómo un árbol de rangos de una dimensión puede adaptarse para que una consulta así se pueda realizar en tiempo $O(\log n)$.
2. Usando la solución al problema para una dimensión, describe cómo se pueden responder consultas de conteo en rangos de d dimensiones en tiempo $O(\log^d n)$.
3. Describe cómo se puede usar la técnica de cascada para mejorar el tiempo de consulta en un factor $O(\log n)$ para dos y más dimensiones.

Respuesta

<Tu respuesta aquí>

Pregunta 9

Diseña e implementa una versión de un Treap que incluya la operación $get(i)$, que regrese la llave con rank i en el Treap. (Hint: Haz que cada nodo, u , mantenga un registro del tamaño del subárbol enraizado en u).

Respuesta

<Tu respuesta aquí>

Pregunta 10

Implementa un `TreapList`, una implementación de la interfaz lista como un Treap. Cada nodo en el Treap debería almacenar un elemento de la lista. Todas las operaciones de la Lista como *get(i)*, *set(i, x)*, *add(i, x)* y *remove(i)* deben tener una complejidad de $O(\log n)$ esperado.

Respuesta

<Tu respuesta aquí>