

Estructuras de Datos Avanzadas

Tarea Examen 1

Profesora: Adriana Ramírez Viguera^{*}

Ayudantes:

Víctor Emiliano Cruz Hernández^{**} Carmen Paola Innes Barrón^{***}

Fecha de entrega: 23 de septiembre de 2024.

1. Una familia de árboles es balanceada, si todo árbol de la familia tiene altura $O(\log(n))$, donde n es el número de nodos en el árbol. Determina si las siguientes familias de árboles binarios, son familias balanceadas. Justifica tu respuesta.
 - a) Cada nodo del árbol es una hoja o tiene dos hijos.
 - b) Existe una constante c tal que, para cada nodo en el árbol, las alturas de sus sub-árboles difieren a lo más c .
 - c) La profundidad promedio de un nodo es $O(\log(n))$.
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.
3. Considera el treap T después de insertar x , con el algoritmo visto en clase. Sea C la longitud del camino derecho del subárbol izquierdo de x . Sea D la longitud del camino izquierdo del subárbol derecho de x . Demuestre que el número total de rotaciones que se realizaron durante la inserción de x es igual a $C + D$.
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.
5. Muestra cómo transformar una *skip-list* L en un árbol binario de búsqueda $T(L)$. Y justifica por qué buscar en $T(L)$ no es más rápido que en L .

^{*} adriana.rv@ciencias.unam.mx

^{**} v23ec02@ciencias.unam.mx

^{***} paoinnes@ciencias.unam.mx

-
6. Demuestra o da un contraejemplo:
- a) Los nodos de cualquier árbol AVL pueden colorearse de rojo y negro para obtener un árbol rojo-negro válido.
 - b) Cualquier árbol rojo-negro satisface las propiedades de árbol AVL.
7. We say that a binary search tree T_1 can be right-converted to binary search tree T_2 if it is possible to obtain T_2 from T_1 via a series of calls to Right-Rotate. Give an example of two trees T_1 and T_2 such that T_1 cannot be right-converted to T_2 . Then show that if a tree T_1 can be right-converted to T_2 , it can be right-converted using $O(n^2)$ calls to Right-Rotate.
8. Suppose a 2 – 4 tree, T , has n_l leaves and n_i internal nodes.
- a) What is the minimum value of n_i , as a function of n_l ?
 - b) What is the maximum value of n_i , as a function of n_l ?
 - c) If T' is a red-black tree that represents T , then how many red nodes does T' have?
9. Diseña e implementa un Treap que incluya al menos las siguientes operaciones:
- `insert(V,P)`: Agrega el valor V al Treap con su respectiva prioridad P
 - `delete(V)`: Elimina la primera ocurrencia de V en el Treap, si existe.
 - `get(V)`: Obtiene el nodo con valor V que se encuentra en el Treap, si existe
 - `get(i)`: Obtiene el nodo cuyo valor sea el i -ésimo más grande en el Treap, si i está en el rango de elementos.
 - `peek()`: Regresa el nodo hasta el tope del Treap;
 - `pop()`: Elimina y regresa el primer elemento al tope del Treap;

Hint para `get(i)`: Haz que cada nodo, mantenga un registro del tamaño del subárbol enraizado derecho e izquierdo.