Estructuras de Datos y Algoritmos – IIC2133

I2

8 nov. 2021

Hora de inicio: 14.00
Entrega: hasta las 23.59

0) Responde esta pregunta de manera manuscrita.

a) Nombre completo y número de alumno: _____

b) Me comprometo a **no preguntar ni responder dudas** de esta prueba, ya sea directa o indirectamente, a nadie que no sea parte del equipo docente del curso. **Firma**: _____

Responde sólo 3 de las siguientes 4 preguntas

1) Tú estás preparando un sistema que recibe consultas de forma continua y necesitas mantener una estructura de datos que las pueda responder lo más rápido posible.

Las consultas son de dos tipos:

1. Recibir un valor entero entre 1 y n.
2. Recibir tres valores a, b y k, donde $1 \leq a \leq n$ y $1 \leq b \leq n$, y responder la siguiente consulta – Si tomamos los valores que hemos visto hasta ahora y consideramos solo los que están dentro del rango $[a, b]$, cuál es el k-ésimo menor.

Para esto, tienes que idear una estructura de datos que contenga los métodos $\mathsf{add}(x)$ y $\mathsf{query}(a, b, k)$. La complejidad de cada consulta debe ser a lo más $O(\log^2 n)$. No es necesario que escribas el pseudocódigo completo, pero di con detalle cuáles son las estructuras de datos subyacentes y cómo se modifican para responder las consultas y alcanzar la complejidad deseada. La estructura de datos tiene que poder crearse en tiempo lineal en n, o menor.

Ejemplo de secuencia de consultas:

$\mathsf{add}(1)$ – Los valores vistos son $[1]$.
$\mathsf{add}(4)$ – Los valores vistos son $[1, 4]$.
$\mathsf{add}(10)$ – Los valores vistos son $[1, 4, 10]$.
$\mathsf{add}(4)$ – Los valores vistos son $[1, 4, 10, 4]$.
$\mathsf{add}(8)$ – Los valores vistos son $[1, 4, 10, 4, 8]$.
$\mathsf{query}(1, 9, 3)$ – Buscamos el tercer menor valor en $[1, 4, 4, 8]$, y retornamos 4.
$\mathsf{add}(1)$ – Los valores vistos son $[1, 4, 10, 4, 8, 1]$.
$\mathsf{add}(6)$ – Los valores vistos son $[1, 4, 10, 4, 8, 1, 6]$.
$\mathsf{query}(2, 6, 1)$ – Buscamos el menor valor en $[4, 4, 6]$, y retornamos 4.
$\mathsf{query}(1, 9, 5)$ – Buscamos el quinto menor valor en $[1, 4, 4, 8, 1, 6]$, y retornamos 6. (Fíjate que si ordenamos estos números queda $[1, 1, 4, 4, 6, 8]$, y en la quinta posición está el 6. Esto es equivalente.)

2) N corredores llegan a la meta. Para el corredor que llega primero se registra el tiempo que demoró en terminar la carrera, pero para cada uno de los otros corredores se registra sólo el número de segundos adicionales al tiempo del primero. Con estos tiempos se crea una lista T con N valores, en que el primer valor es 0. Además, con el propósito de hacer diversos estudios, se registra también la diferencia de tiempos, en segundos, entre cada par de corredores; esta es una lista D con $N(N–1)/2$ valores ordenados de menor a mayor. Lamentablemente, se nos perdió la primera lista, T. Da un algoritmo de *backtracking* que a partir de la segunda lista, D, reconstituya T.

Por ejemplo, si $N = 4$ y $T = [\ 0, 40, 65, 110\ ]$, entonces $N(N–1)/2 = 6$ y $D = [\ 25, 40, 45, 65, 70, 110\ ]$.

3)

a) Tenemos una lista L de n pares (a, b) de números enteros: a representa un valor y b representa una prioridad. Construye un árbol binario T con n nodos, en que cada par corresponde a un nodo, que cumpla con la siguiente propiedad: si nos fijamos sólo en los valores a de cada nodo, entonces el árbol es un árbol de búsqueda (no necesariamente balanceado); y si nos fijamos sólo en las prioridades b de cada nodo, entonces el árbol es un *max-heap* (de nuevo, no necesariamente balanceado).

Por ejemplo, si $n = 4$ y $L = \{ (18, 11), (6, 17), (12, 21), (14, 7) \}$, entonces T tiene el par (12, 21) como raíz, (6, 17) como su hijo izquierdo, (18, 11) como su hijo derecho, y (14, 7) como hijo izquierdo de este último.

Resuelve el problema para

$L = \{ (38, 37), (42, 35), (60, 3), (52, 9) (46, 31), (40, 39), (32, 21), (24, 29), (28, 43), (44, 41), (48, 47), (56, 17), (72, 7), (70, 23), (36, 45) \}$, con $n = 15$.

Dibuja el árbol T resultante.

b) Los aproximadamente 25 mil estudiantes de la universidad van a elegir a sus k representantes por votación directa. Cada estudiante puede votar por cualquiera de los 25 mil estudiantes, y los k estudiantes con más votos resultan elegidos. Para emitir un voto, simplemente se escribe el Nº de alumno del estudiante por el cual se desea votar. Describe con suficiente detalle las estructuras de datos y los algoritmos para usarlas que permitan implementar esta votación eficientemente, de modo que en cualquier instante de tiempo se pueda saber cuáles son los k estudiantes que hasta ese momento llevan más votos.

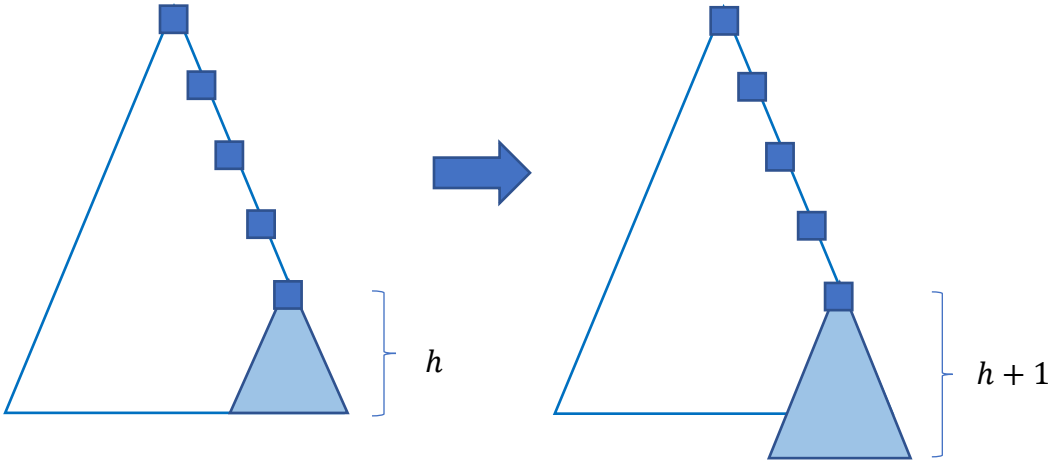
Estructuras de Datos y Algoritmos – IIC2133

I2

8 nov. 2021

4) En esta pregunta vas a responder a una pregunta que ha agobiado a los científicos por años. Si tenemos dos árboles AVL A y B, y una llave x, de tal manera que todas las llaves de A son menores que x, y todas las llaves de B son mayores que x, ¿cuánto tarda armar un árbol AVL C que sea la unión de estas tres partes? Vamos a responder esta pregunta de a poco.

a. Si A y B tienen la misma altura, diseña un algoritmo que crea el árbol C en tiempo constante y justifica por qué funciona. (0 puntos)
b. Si las alturas de A y B difieren en 1, diseña un algoritmo que crea el árbol C en tiempo constante y justifica por qué funciona. (0 puntos)
c. Si las alturas de A y B difieren en 2, diseña un algoritmo que crea el árbol C en tiempo constante y justifica por qué funciona. (0.5 puntos)
d. Imagina que tienes un árbol AVL T de n elementos, ya balanceado, y a alguna fuerza extraña se le ocurre reemplazar un subárbol de altura h por otro (también balanceado) que tiene altura $h + 1$. En específico, el árbol que se reemplaza es uno que se puede alcanzar bajando solo por los hijos derechos.



Demuestra que balancear este árbol toma tiempo $O(\log n)$. Puedes asumir que el árbol nuevo ya respeta el orden correcto de los elementos para ser un ABB. (0.8 puntos)

e. Diseña un algoritmo que reciba los árboles A y B, y el elemento x con las características mencionadas en el enunciado, y cree un árbol C que sea la unión de estas tres partes y que tarde tiempo $O(\log n)$. (0.7 puntos)

Importante: Para responder e. puedes asumir que ya sabes que d. es cierto.