

Análisis de Algoritmos e introducción a Matemáticas Discretas

Tarea 5

Rubén Pérez Palacios Lic. Computación Matemática
Profesor: Dr. Carlos Segura González

15 de enero de 2024

Podemos traducir el problema a tener un arreglo donde guardemos los id's de las pelotas o identificar que no hay pelota en esa posición. Luego nos piden responder el máximo id de las pelotas en un rango, también poder eliminar algunas de ellas y tirar de las pelotas a su derecha, o mover pelotas a distintos lugares.

La complejidad de las query's anteriores debe ser de a lo más $O(\log(n))$, por lo que hacer la fuerza bruta de cada una de ellas, es decir recorrer el rango para saber el máximo, eliminar una y recorrer cada una de las pelotas a su derecha una posición a la izquierda, o recorrer las pelotas para poder mover una pelota; todas estas acciones son de $O(n)$, por lo que requerimos de una estructura de datos para poder reducir esta complejidad.

Un treap es capaz de hacer todas las anteriores query's en $O(n)$. Nuestro treap representara nuestro arreglo, es decir en cada uno de los nodos almacenaremos el id de alguna pelota. La forma de implementar las query's es como sigue:

- Máximo: Debemos agregar una variable *tmax* a nuestra estructura de Nodo, donde almacenemos el máximo de los id's del subárbol de ese nodo. Notemos que esta variable está en función de sus hijos ya que el máximo de los id's del subárbol de ese nodo es el máximo de entre el máximo de los subárboles de sus hijos y del id del nodo. Para mantener actualizado el *tmax* de este nodo basta con actualizar en funciones de sus hijos de él cada vez que hacemos un cambio a la estructura del treap dentro de las funciones split o merge.
- Eliminar: Bastará con cortar nuestro treap alrededor del nodo de la posición que deseamos eliminar y juntar estos treap, para terminar simular el recorrido faltará agregar un nodo al final del treap que represente el espacio vacante.
- Intercambiar: Bastará con eliminar el nodo de la posición de la pelota que deseamos mover e insertarla en el lugar donde deseamos dejarla.

El problema pide un tipo de query más, la cual es poder construir un "muro" la cual nos asegurará que no interferirá con el funcionamiento de las query's máximo e intercambiar, en cambio a la query eliminar si la afectara ya que cuando deseamos eliminar una pelota recorreremos solo las pelotas entre la posición de la pelota eliminada y el muro mas próximo a su derecha. En la implementación de nuestro treap de eliminar solo debemos modificar en donde insertar el espacio vacante, ya que este deberá ser insertado un lugar antes del muro más próximo a su derecha, de no existir entonces si hasta el final. Encontrar el muro deseado tendrá que ser de complejidad a lo más $O(\log(n))$ por lo que bastará tener una estructura de datos donde tenga ordenadas las posiciones y poder preguntar ¿cuál es la posición del muro más próximo a nuestra derecha?