

OBSERVACIONES DEL RETO 3

María Catalina Ibáñez Piñeres Cod 201922462 Mail m.ibanez@uniandes.edu.co

María Alejandra Pérez Petro Cod 201923972 Mail ma.perezp@uniandes.edu.co

Requerimiento 3 – Realizado por María Catalina Ibáñez Piñeres

Requerimiento 4 – Realizado por María Alejandra Pérez Petro

Análisis de complejidad de cada uno de los requerimientos en Notación O

Carga de datos

La complejidad de la carga de datos es $O(N \log(N))$ donde N es el número de jugadores. Esta complejidad se debe a que la creación de cada RBT tiene una complejidad $O(N \log(N))$ donde N representa cada jugador añadido y $\log(N)$ la complejidad asociada a hacer insertion en el árbol de un jugador.

Requerimiento 1

La complejidad del primer requerimiento es $O(N * \log(N))$ donde N es el número de jugadores en el club de interés. Esta complejidad se debe a que primero se obtienen los jugadores del club solicitado con el map clubName y luego estos se organizan por fecha de entrada al club por medio de Merge sort. El Merge Sort es la operación que más aporta a la complejidad, entonces la complejidad del requerimiento 1 es $O(N * \log(N))$.

Requerimiento 2

La complejidad del segundo requerimiento es $O(N * \log(N))$ donde N es el número de jugadores que juegan en la posición de interes dentro del rango de desempeño, potencial y salario especificado. Esta complejidad se debe a que primero se obtienen los jugadores que juegan en la posición solicitada con el RBT player_positions y luego estos se recorren para obtener solo los jugadores dentro del rango de desempeño, potencial y salario especificado. Finalmente, los jugadores obtenidos se organizan por medio de Merge sort. Por lo que, la complejidad del requerimiento 2 es $O(N * \log(N))$, puesto que el Merge Sort es la operación que más aporta a la complejidad.

Requerimiento 3

La complejidad del tercer requerimiento es $O(NM * \log(NM))$ donde N es el número de índices (salarios) dentro del intervalo de salarios de interés y M es el número de jugadores que tienen un salario N . Esta complejidad se debe a que para poder conocer los jugadores que están dentro del intervalo del salario de interés, toca recorrer primero cada índice (cada salario dentro del intervalo) y luego cada jugador dentro de ese índice. Después de hacer esos recorridos y haber agregado cada uno de los jugadores a un array_list que queda de tamaño NM , esta se organiza por salario del jugador por medio de Merge sort. El Merge

Sort es la operación que más aporta a la complejidad, por lo que la complejidad del requerimiento 3 es $O(NM \cdot \log(NM))$.

Requerimiento 4

La complejidad del cuarto requerimiento es $O(NM \cdot \log(NM))$ donde N es el número de índices (fecha de nacimiento(dob)) dentro del intervalo de fechas de nacimiento de interés y M es el número de jugadores que tienen fecha de nacimiento N. Esta complejidad se debe a que para poder conocer los jugadores que están dentro del intervalo de fechas de nacimiento de interés, toca recorrer primero cada índice (cada fecha de nacimiento dentro del intervalo) y luego cada jugador dentro de ese índice. Después de hacer esos recorridos y haber agregado cada uno de los jugadores a un array_list que queda de tamaño NM, esta se organiza por fecha de nacimiento del jugador por medio de Merge sort. El Merge Sort es la operación que más aporta a la complejidad, por lo que la complejidad del requerimiento 4 es $O(NM \cdot \log(NM))$.

Requerimiento 5

La complejidad del quinto requerimiento es $O(NM)$ donde N es el número de intervalos y M es el número atributos en la propiedad dada por el usuario (es decir, si el usuario escoge wage_eur, entonces M es el número de salarios existentes). Esta complejidad se debe a que se recorre el número de grupos/intervalos N que se van a armar y, luego, los M atributos de la propiedad que hay en cada intervalo. Por lo que la complejidad del requerimiento 5 es $O(NM)$.

Requerimiento 6 (BONO)

La complejidad del sexto requerimiento es $O(N)$ donde N es el número de jugadores en la posición de interés. Esta complejidad se debe a que primero se obtienen los jugadores que juegan en la posición solicitada con el RBT player_positions. Luego, los jugadores que juegan en una posición de interés se recorren para calcular el valor representativo de cada jugador (vr) y posteriormente calcular la diferencia en valor absoluto entre el vr del jugador y el vr del jugador a sustituir. El recorrido es la operación que más aporta a la complejidad por lo que, la complejidad del sexto requerimiento es $O(N)$.