Problema B

0. Identificación

Daniel Santiago Tenjo Leal 201815459

Santiago Estupiñán Romero 201813414

1. Algoritmo de Solución

La solución planteada para este problema se puede dividir en tres partes principales que serán explicadas a continuación. Teniendo cuenta que el tamaño del problema es N y 1≤N≤10^4 y que todos los datos que entran son de tipo entero positivo podemos empezar a entender la solución. La primera parte de la solución consiste en la creación de una inner class para almacenar los datos de cada persona k y poder manejar la información de una forma más apropiada, entonces lo primero que se realiza es cargar todos los datos que entran en un arreglo de tamaño fijo de tipo Tupla (la clase que creamos).

En segundo lugar, implementamos el método compareTo dentro de esta clase, este método nos resultara útil para ordenar el arreglo de forma ascendente usando un sort de java. Luego de realizar el sort tenemos un arreglo ordenado ascendentemente sobre el atributo A. Ahora tenemos que encontrar la subsecuencia descendente más grande sobre el atributo B.

Para esta parte nos basamos en una explicación de geeks for geeks. Para encontrar la subsecuencia, decidimos crear un arreglo auxiliar de enteros y inicializarlo en unos. Luego usamos un doble recorrido que incrementa la posición en el arreglo auxiliar donde encuentre un valor de B inferior al que se esta observando, esto se realiza por cada valor lo que permite encontrar el arreglo descendente de mayor tamaño. Luego se lee el arreglo auxiliar para encontrar la longitud de la subsecuencia y los índices de esta. Para luego usar un for para crear el String que se va a retornar.

Ctx C: N:nat ∧ 1≤N≤10^4 ∧ F[0..N] = [f(0),..., f(N)]

Pos R: rta:int ∧ 1≤rta≤N

2. Análisis de complejidades espacial y temporal

T(n) = O(cargaDeDatos) + O(ordenamientoA) +O(ordenamientoB)

O(n)//La carga se realiza en un recorrido de 0 a N

+ O(nlog(n)) // El ordenamiento de Java tiene una complejidad de nLogN

+O(n + n^2 + n + n) // La creación y inicialización del arreglo auxiliar, la búsqueda de la secuencia más larga, la búsqueda de los índices y de la longitud de la secuencia.

O(n) + O(nlog(g)) + O(n^2)

T(n)=O(n^2)

S(n) = O(arregloTuplas)+O(arregloAuxiliar)+O(arregloSolucion)

O(n)+O(n)+O(n)

S(n)= O(n)

3.Comentarios Finales

Luego de implementar la solución explicada anteriormente se realizaron las pruebas planteadas en la guía y se realizaron unas adicionales de nuestra autoría, no se dieron errores en estas pruebas.