**Proyecto Tercer Tercio-AYPR**

Ángel Nicolas Cuervo Naranjo

Jefer Alexis González Romero

12/05/2021

**Texto seleccionado:**

Evaluaciones Agropecuarias por consenso DEPARTAMENTO DE BOYACÁ

**Enlace:**

<https://www.datos.gov.co/Agricultura-y-Desarrollo-Rural/Evaluaciones-Agropecuarias-por-consenso-DEPARTAMEN/u958-pr9h>

**Ordenamiento:**

Se ordenan de mayor a menor dependiendo del rendimiento por área sembrada.

**Criterio de búsqueda:**

Dado un nombre de cultivo busca si este se encuentra en un municipio de Boyacá (también dado) y si esta devuelve el total del área cosechada.

***Entrega N°2:***

*\*Subprograma modificar*

*\* Algoritmo de ordenamiento N°1*

*\* Algoritmo de ordenamiento N°2*

*\*Algoritmo de búsqueda N°1*

*\*Comparativo de tiempo entre los algoritmos de ordenamiento*

***Entrega N°3:***

*\* Se le pide el nombre del archivo al usuario.*

*\* Dice cuando no se encuentra el municipio o cultivo dado.*

*\* Se guardan los ordenamientos en rutas dadas.*

*\* Ordenamiento con la función sorted() para ver la diferencia de tiempo.*

*\* El funcionamiento que hacía slice ahora lo hacen insert() y pop(), para disminuir el espacio de almacenamiento.*

*\*Algoritmo de búsqueda N°2*

*\*Comparativo de tiempo entre los algoritmos de búsqueda*

***Entrega N°4:***

*\* Algoritmo de Ordenamiento N°4(merge\_sort)*

**Subprogramas**

def modificar(arreglo):

"""Funcion que arregla los datos punto flotantes, elimina el salto de linea del ultimo elemento de cada fila, si encuentra un dato vacio que debería ser un numero lo reemplaza por un cero, los elementos en las columnas que deberian tener una "enhe", cambia esta por una "n"

(list) --> list"""

**Algoritmo de ordenamiento N°1:**

Se usa el algoritmo de ordenamiento “Inserccion”. A un nuevo arreglo se le agrega cada fila del archivo, estas se añaden hasta que se encuentre una fila que tenga menor o igual rendimiento, posicionándola antes de la que tiene menor rendimiento. Si se encuentra con una de igual rendimiento tendrá mayor relevancia la que tenga menor hectáreas sembradas.

**Algoritmo de ordenamiento N°2:**

Se usa el algoritmo de ordenamiento “Burbuja”, añadiendo únicamente, al igual que el ordenamiento N°1, la situación en el que si dos tienen el mismo rendimiento, el que tenga menor número de hectáreas sembradas se posicionará antes.

**Algoritmo de ordenamiento con sorted():**

Con la función sorted el tiempo que requiere para ordenar es menor en comparación con los ordenamientos 1 y 2. Se ordenan las filas con respecto a la columna 7, si hay filas con el mismo rendimiento no se ordenan por el número de hectáreas.

**Algoritmo de ordenamiento N°4:**

Se usa el algoritmo de ordenamiento “merge\_sort”, este ordenamiento usa técnica recurrente, su tiempo de ejecución es menor con respecto a los algoritmos de ordenamiento 1 y 2, pero es mas lento que la función sorted(). Al igual que los ordenamientos 1 y 2, la situación en el que si dos tienen el mismo rendimiento, el que tenga menor número de hectáreas sembradas se posicionará antes.

**Algoritmo de búsqueda N°1:**

Se usa búsqueda secuencial, comparando cada municipio y cultivo de todas las filas con los datos que se piden, se retorna las filas que coinciden con lo dado y al final se da el total de hectáreas cosechadas.

**Algoritmo de búsqueda N°2:**

Con las filas ordenada alfabéticamente por municipios, se usa búsqueda binaria, cuando encuentra el municipio, se busca en las filas que tienen el municipio el cultivo deseado.