

# Tarea 1 - Problema 1

Rodriguez Prieto Diego Guadalupe

Profesor/Dr. Eduardo Arturo Rodríguez Tello

## 1. Introducción

El presente reporte nos muestra una manera de resolver la problemática planteada para el problema 1, el cual nos presenta una situación en la que el coordinador del CINVESTAV tiene la encomienda de encontrar la edad mayoritaria  $x$  de un grupo de estudiantes del posgrado de maestría, dicha edad se encuentra almacenada en un vector  $S$  ordenado de menor a mayor de tamaño  $n$  y debe encontrarse en una ocurrencia de  $n/2$ .

## 2. Metodología

Para la resolución de esta problemática, inicialmente debemos documentarnos con opciones de algoritmos que nos propongan una manera efectiva de realizarlo, con ello se presentan dos opciones:

- Búsqueda binaria
- Búsqueda lineal

### 2.1. Pseudocódigo

El problema puede descomponer en 4 partes importantes:

1. **Leer datos:**  $n, x$  y  $S[]$ .
2. **Buscar la edad solicitada:** si  $x \in S$
3. **Verificar si la edad es mayoritaria:**  $x > n/2$
4. **Resultado:** al ser mayoritaria, devolver un *True*, caso contrario, *False*.

Dada la naturaleza que se nos expone, siendo  $S$  un vector ordenado de menor a mayor, se infiere que la mejor opción es la búsqueda binaria (para el paso 2), ya que en estas condiciones es cuando se puede obtener un mejor aprovechamiento del algoritmo presentando una complejidad algorítmica de  $O(\log n)$  [3].

El algoritmo de búsqueda binaria[2], muestra que al dividir el vector a la mitad, podemos hacer en el menor de los casos, una sola búsqueda (siendo que el numero solicitado este en dicha posición), o que haga la mitad de las búsquedas que haría una búsqueda lineal, debido a que se puede ir por la derecha o por la izquierda sea el caso que el numero sea mayor o menor que el pivote (tal como se expresa en el pseudocódigo que se muestra a continuación).

#### Algorithm 1 Búsqueda binaria.

```
1: procedure BINARYSEARCH( $S, x$ )
2:   inicio  $\leftarrow 0$ , fin  $\leftarrow |S| - 1$ 
3:   while inicio  $\leq$  fin do
4:     mid  $\leftarrow$  inicio + (fin - inicio) / 2
5:     if  $S[mid] = x$  then
6:       if mid = 0 or  $S[mid - 1] \neq x$  then
7:         return mid
8:       else
9:         fin  $\leftarrow$  mid - 1
10:      end if
11:    else if  $S[mid] < x$  then
12:      inicio  $\leftarrow$  mid + 1
13:    else
14:      fin  $\leftarrow$  mid - 1
15:    end if
16:  end while
17:  return -1
18: end procedure
```

En el momento en el que encontramos dicho numero, es requerido ver si este se repite la cantidad de veces necesarias para poder

determinar que se pueda clasificar como mayoritario, para ello hay 2 casos para cuando seria falso, siendo que el numero aparezca pero no se repita la cantidad de veces requeridas o que no aparezca y un solo caso para True, que es cuando aparece una cantidad igual o mayor a  $n/2$ .

#### Algorithm 2 Edad Mayoritaria.

```
procedure EDADMAYORITARIA( $S, n, x$ )
2:   primeraAparicion  $\leftarrow$  BINARYSEARCH( $S, x$ )
3:   if primeraAparicion = -1 then
4:     return false
5:   end if
6:   if primeraAparicion +  $n / 2 < n$  and  $S[primeraAparicion + n / 2] = x$  then
7:     return true
8:   else
9:     return false
10:  end if
end procedure
```

## 3. Resultados

Los resultados haciendo uso del algoritmo de búsqueda binaria, fueron notorios hablando de la mejora que este presenta en comparación con la búsqueda lineal, teniendo un 11 % de media, teniendo ambientes de prueba en donde solo se trabajaba con archivos txt de solo 200 elementos para el vector.

## 4. Discusión

Aunque es evidente cómo la implementación donde se hace uso de la búsqueda binaria es mejor con respecto a la búsqueda lineal, no llega a ser tan evidente debido al vector de tamaño tan pequeño como con el que se esta trabajando, pero al hablar de un vector con una cantidad de componentes como lo seria en  $1 \times 10^6$ , el porcentaje de mejora se haría mayor.

## 5. Conclusión

Aunque la implementación del algoritmo de búsqueda binaria es buena y tiene un alto impacto en el tiempo de ejecución del código, se pudiese mejorar aun mas, haciendo uso de otras herramientas como lo es el uso de memoria dinámica especifica, ya que aunque la clase vector ya hace internamente uso de ella [1], podria hacerse mayor uso de ella.

## Referencias

[1] «Duda sobre la clase vector». Accessed: 2024-09-09. (mayo de 2018), dirección: <https://es.stackoverflow.com/questions/163092/duda-sobre-la-clase-vector>.

[2] G. for Geeks, «Binary Search», *Geeks for Geeks*, sep. de 2020. dirección: <https://www.geeksforgeeks.org/binary-search/>.

[3] «Difference Between Linear Search and Binary Search». Accessed: 2024-09-09. (), dirección: <https://www.shiksha.com/online-courses/articles/difference-between-linear-search-and-binary-search/>.