*BÚSQUEDAS BINARIAS DIVIDE Y VENCERÁS*

*ANÁLISIS Y DISEÑO DE ALGORITMOS*

*CORTE 3*

*ARIEL MEJIA SUAREZ*

*FACULTAD DE INGERIERIA*

*INGENIERIA DE SISTEMAS*

*UNIVERSIDAD DE SAN BUENAVENTURA*

*BOGOTA DC*

*2020*

1. Explicar en qué consiste la técnica de programación divide y vencerás. (Definición, características, uso).
   1. *Definición técnica divide y vencerás*: es una técnica algorítmica a cuál se define como la manera de tomar un problema o un arreglo de números, dividirlo en subproblemas del mismo tipo de un tamaño más pequeño para al final hallar la solución.

Para hallar la solución en un problema por medio de este método se deben seguir estos pasos:

1. El problema debe poderse descomponer en varios subproblemas de un mismo tipo, pero de una longitud menor.
2. Estos subproblemas deben solucionarse independientemente y se necesita que sean de menor tamaño y de manera inicial se pueden resolver de manera recursiva o iterativa
3. Y ya como último paso se reúnen los resultados de los dos pasos anteriores para de esta manera hallar la solución que se pide al problema original.

Estos problemas que optan por la solución de divide y veceras siguen un esquema general previamente definido

Se toma un elemento al que se llama pivote posteriormente se evalúa el valor del pivote y este debe estar en el medio del arreglo, vector o lista una vez se encuentra esta posición y luego hace el intercambio. Se repite este paso, pero para los conjuntos de datos que se encuentren a la izquierda y derecha del pivote y este proceso se acaba cuando los elementos encuentran la posición en el arreglo o vector.

* 1. *Características:*
     1. Divide los problemas en subproblemas cosa que o hacen otros algoritmos
     2. Soluciona cada uno por su cuenta y guarda los resultados
     3. Combina todos los resultados de los procesos anteriores y halla la solución final
     4. Este algoritmo hace muchas llamadas de manera recursiva
     5. Un ejemplo muy común de esta técnica es el ejercicio de las torres de Hanoi
     6. Es una técnica recursiva
     7. Es de tipo
     8. Es el método más rápido y eficiente
     9. Notación Big O seria O () loglinear o quasilienar
  2. *Usos:*
     1. Se usan para ordenar exámenes de admisión de acuerdo con los puntajes
     2. Ordenar puntajes de ranquin en juegos concursos
     3. Manejar las producciones de acuerdo con al porcentaje de las ventas

1. Explicar y dar un contexto en qué consiste el problema del algoritmo “búsqueda binaria”

Es un algoritmo eficiente el cual se usa para hallar elementos en una lista esta debe estar ordenados sus elementos y aplica la técnica divide y vencerás la cual aplica el concepto de dividir el problema en varias sub-partes las cuales podrían tener el elemento o dato.

Es muy usado para encontrar un elemento, nombre o número junto con su posición en un arreglo un ejemplo claro podría ser el siguiente:

“*el catálogo estelar Tycho-2 contiene información acerca de las 2,539,913 estrellas más brillantes en nuestra galaxia. Supón que quieres buscar en el catálogo una estrella en particular, con base en el nombre de la estrella. Si el programa examinara cada estrella en el catálogo estelar en orden empezando con la primera, un algoritmo llamado****búsqueda lineal****, la computadora podría, en el peor de los casos, tener que examinar todas las 2,539,913 de estrellas para encontrar la estrella que estás buscando. Si el catálogo estuviera ordenado alfabéticamente por nombres de estrellas, la****búsqueda binaria****no tendría que examinar más de 22 estrellas, incluso en el peor de los casos*.”

Tomado de:

”<https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search>”

Ejemplo Búsqueda Binaria en C++:

#include<iostream>

#include<conio.h>

**using** **namespace** std**;**

int main**(){**

int numeros**[]** **=** **{**1**,**2**,**3**,**4**,**5**};**

int inf**,**sup**,**mitad**,**dato**,**i**;**

char band**=**'F'**;**

dato **=** 1**;**

//Algoritmo de la Busqueda Binaria

inf**=**0**;**

sup**=**5**;**

i**=**0**;**

**while((**inf**<=**sup**)&&(**i**<**5**)){**

mitad **=** **(**inf**+**sup**)/**2**;**

**if(**numeros**[**mitad**]** **==** dato**){**

band**=**'V'**;**

**break;**

**}**

**if(**numeros**[**mitad**]>**dato**){**

sup **=** mitad**;**

mitad **=** **(**inf**+**sup**)/**2**;**

**}**

**if(**numeros**[**mitad**]<**dato**){**

inf **=** mitad**;**

mitad **=** **(**inf**+**sup**)/**2**;**

**}**

i**++;**

**}**

**if(**band **==** 'V'**){**

cout**<<**"El numero se encontro en la pos: "**<<**mitad**<<**endl**;**

**}**

**else{**

cout**<<**"El numero NO se encontro"**;**

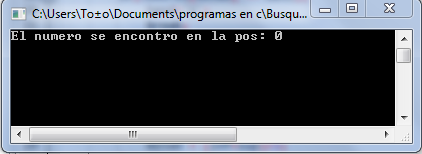
**}**

getch**();**

**return** 0**;**

**}**

Salida:



1. Explicar cómo va a resolver el algoritmo. (descripción del algoritmo, características, elementos clave)
   1. *descripción del algoritmo:* teniendo en cuenta la investigación realizada, se dividió el problema en las siguientes partes:
      1. Investigación de la técnica.
      2. Investigación del algoritmo.
      3. Investigación de la aplicación de la técnica al algoritmo.
      4. Aplicación del código en Python de la técnica.
      5. Aplicación del código en Python al algoritmo.
      6. Código de interacción con el usuario

Nuestro algoritmo parte de dos partes, es una función recursiva en la primera parte del código encontramos una función la cual lleva el código y un main el cual nos sirve para invocar o imprimir las peticiones e imprimir las respuestas

*Código de la Función:*

**def** binary\_search\_it**(**array**,** x**):**

left **=** 0

right **=** **len(**array**)** **-** 1

**while** left **<=** right**:**

mid **=** **(**left**+**right**)//**2

**if** array**[**mid**]** **==** x**:**

**return** mid

**elif** array**[**mid**]** **>** x**:**

right **=** mid**-**1

**else:**

left **=** mid**+**1

**return** **-**1

*Código del main:*

**def** main**():**

array **=** **[]**

array**.**sort**()**

x**=** **input(**"Elemento: "**)**

**while** x **!=** ""**:**

array**.**append**(**x**)**

x **=** **input(**"Elemento siguiente: "**)**

array**.**sort**()**

**print(**array**)**

x **=** **input(**"Ingrese el numero a buscar: "**)**

x **=** x

pos **=** binary\_search\_it**(**array**,**x**)**

**if** **(**pos**>-**1**):**

**print(**"el elemento se encuentra en la pocicion: "**,**pos**+**1**,** "con el numero:"**,** x**)**

**else:**

**print(**"el elemento no se encuetra en la lista"**)**

main**()**

* 1. *Características:*
     1. Se dividió el problema en dos partes.
     2. Tiene una función y un main.
     3. Se aplicó el concepto recursivo.
  2. *Elementos clave:*
     1. Se definieron dos variables iniciales dentro de la función “binary\_search\_it” las variables fueron el arreglo y el dato de entrada
     2. Se inicializo el arreglo en 0 hasta n datos
     3. Se inicio el ciclo while donde se compara los extremos del arreglo para ingresar los datos en el arreglo por el usuario.
     4. Se iniciaron los else que comparan el dato a buscar con los datos guardados dentro del arreglo
     5. En el “main**()**”de declaró nuevamente el arreglo y con la función *“sort()”*
     6. Se ingresan los datos por teclado para llenar el vector
     7. Con un doble intro se define q no se quieren llenar más datos
     8. Se imprime el vector con los datos ya digitados
     9. Se pide el número a buscar
     10. Y se hace el ciclo para buscar el número
     11. Se imprime el número y la posición si se encontró
     12. Si no se imprime el “el elemento no se encuentra en la lista”.

## REFERENCIAS

<https://runestone.academy/runestone/static/pythoned/SortSearch/LaBusquedaBinaria.html>

<https://uniwebsidad.com/libros/algoritmos-python/capitulo-8/busqueda-binaria>

<https://es.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithmshttps://es.khanacademy.org/computing/computer-science/algorithms/mergesort/a/divide-and-conquer-algorithms>

<https://parzibyte.me/blog/2018/11/08/algoritmo-busqueda-binaria-lenguajes-programacion/>

<https://es.qwe.wiki/wiki/Divide_and_conquer_algorithm#Divide_and_conquer>

<https://mapecode.com/busqueda-binaria-en-python/>

<https://www.youtube.com/watch?v=PA54cyycix8>