

# Practica 2

Los utlimos

28 de septiembre de 2021

# Analisis de casos por codigo

## Busqueda con Arbol Binario de busqueda

Operaciones basicas consideradas: comparacion entre la llave a buscar y el elemento del nodo, la comparacion entre el arbol auxiliar y nulo y la asignacion de auxiliar.

Condiciones del mejor caso: que el elemento a buscar este en la raiz.

Condiciones del peor caso: un arbol que tenga n niveles y el elemento este en el nodo hoja.

Funcion de complejidad temporal del mejor caso = 1

Funcion de complejidad temporal del peor caso =  $4n$

Funcion de complejidad temporal del caso medio =  $\frac{1+4\log_2(n)}{2}$

## Busqueda Binaria

Operaciones basicas consideradas: comparaciones de la llave con elementos del arreglo, actualizacion de limite inferior y superior y la asignacion de h.

Condiciones del mejor caso: cuando el numero a buscar esta en medio del arreglo.

Condiciones del peor caso: cuando el numero a buscar esta al final del arreglo.

Funcion de complejidad temporal del mejor caso = 4

Funcion de complejidad temporal del peor caso =  $4\log_2(n)$

Funcion de complejidad temporal del caso medio =  $\frac{4\log_2(n)+4+3\log_2(n)}{3} -$

## Busqueda exponencial

Operaciones consideradas: Comparaciones entre el elemento buscado y los elementos del arreglo. operaciones con exponent2 e i.

Condición de mejor caso: El número buscado está en la primera posición.

Condiciones Peor caso: El numero esta en una posicion  $2^m + 1$  donde  $2^m$  es el numero potencia de 2 mas cercano y menor a n.

Función de complejidad temporal del mejor caso : 1.

Función complejidad temporal del peor caso:  $3 + 4\log_2(n)$ .

Función complejidad caso medio:  $\frac{3}{2} + 2\log_2(n)$ .

## Busqueda de fibonacci

Operaciones consideradas: Comparaciones entre el elemento buscado y los elementos del arreglo, asignaciones a las variables fibonacci dentro del while de analisis.

Condición de mejor caso: El número buscado está en la primera posición.

Condiciones Peor caso: Cuando el número que buscamos se encuentra en el sub-arreglo que tiene mayor tamaño por lo que hará un mayor número de instrucciones.

Función de complejidad temporal del mejor caso : 1.

Función complejidad temporal del peor caso:  $6\log(n)$ .

Función complejidad caso medio:  $6\log(n)$ .

## Busqueda lineal

Operaciones consideradas: Comparaciones entre el elemento buscado y los elementos del arreglo.

Condición de mejor caso: El número buscado está en la primera posición.

Condiciones Peor caso: El número buscado está al final del arreglo.

Función de complejidad temporal del mejor caso : 1.

Función complejidad temporal del peor caso:  $n$ .

Función complejidad caso medio:  $\frac{1}{n} + (n - 1)$ .