De: Silvia Jara

1. Explicar porque existe la complejidad logarítmica para la búsqueda binaria.

Una condición importante que se tiene que tener en cuenta en una búsqueda binaria es que el vector tiene que estar ordenado. Para implementar este algoritmo se compara el elemento a buscar con un elemento central del vector: Si el valor de éste es mayor que el del elemento buscado se repite el procedimiento en la parte del vector que va desde el inicio de éste hasta el elemento tomado, en caso contrario se toma la parte del vector que va desde el elemento tomado hasta el final. De esta manera obtenemos intervalos cada vez más pequeños, hasta que se obtenga un intervalo indivisible. Si el elemento no se encuentra dentro de este último entonces se deduce que el elemento buscado no se encuentra en todo el vector.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |

n/2

n/8 n/4

……………………………..Complejidad Logarítmica.

1. Explicar la complejidad exponencial para el algoritmo de Fibonacci de doble recursividad.

Se trata de funciones que duplican su complejidad con cada elemento añadido al procesamiento. El cálculo recursivo de la serie de Fibonacci, es muy poco eficiente (se calcula llamándose a sí misma la función con los dos números anteriores: F(n)=F(n-1)+F(n-2)).

Fibonacci (n)

Si n > 2 : retornar 1

Sino

Retornar Fibonacci (n - 2) + Fibonacci (n - 1)



………………….. =

Complejidad Exponencial = O )