¿POR QUE EL ALGORITMO DE FIBONACCI ES EXPONENCIAL?

Por el rápido crecimiento del valor de las sucesiones predecesoras a los anteriores números que le se siguen. Y ahora responderemos.

¿CÓMO RÁPIDO CRECE LA FUNCIÓN FIBONACCI??

Para responder a esa pregunta. Calcularemos las raíces de la ecuación característica asociada a la ecuación en recurrencia F(n) = F(n-1) + F(n-2).   
x2 = x + 1   
o sea, las raíces de x2 - x - 1 = 0

Una de ellas, es el llamado número de oro (Golden Ratio), cuyo valor aproximado es 1.61803 y su valor exacto es c = (1+√5)/2.

Cláramente, se verifica c2 = c + 1 y por tanto, para todo n mayor que dos, también   
cn = cn-1 + cn-2

O sea, la progresión geométrica {cn} satisface la misma ecuación en recurrencia que la función de Fibonacci F(n) = F(n-1) + F(n-2). Ahora, por inducción

Como, c0 = 1 = F(2), c = c1 < 2 = F(3).   
obtenemos que cn-2 < F(n)

En conclusión, la función de Fibonacci crece, como mínimo, exponencialmente. Ahora, enlazando las dos desigualdades, para todo n se tiene

cn-4 < F(n-2) < S(n)

y también la función S(n) crece, como mínimo, exponencialmente.

Por tanto, la programación recursiva de la función de Fibonacci tiene una complejidad, como mínimo, exponencial. Y eso, independientemente, de lo bien que gestione el compilador o intérprete correspondiente la programación recursiva.

¿POR QUE EL ALGORITMO DE BINARIO ES LOGARITMICA?

La búsqueda binaria funciona en arreglos ordenados. La búsqueda binaria comienza por comparar el elemento del medio del arreglo con el valor buscado. Si el valor buscado es igual al elemento del medio, su posición en el arreglo es retornada. Si el valor buscado es menor o mayor que el elemento del medio, la búsqueda continua en la primera o segunda mitad, respectivamente, dejando la otra mitad fuera de consideración.

Ejemplificación.

La estrategia que puede seguir P2 es la de preguntar por el número de la mitad de los posibles candidatos, empecemos

P2: 250 --> P1: mayor

P2: 375 --> P1: mayor aca se descartan lo menores a 250

P2: 438 --> P1: mayor en este punto se han descartado 375 números

P2: 469 --> P1: mayor

P2: 484 --> P1: mayor

P2: 492 --> P1: mayor

P2: 496 --> P1: mayor

P2: 498 --> P1: mayor

P2: 499 --> P1: igual

En esta estrategia solo realizo 9 preguntas, ahora si calculamos el logaritmo en base 2 de 500 tenemos que es 8.9. La complejidad de este algoritmo es precisamente O(log2(n))