Tarea 2

Juan David Vasquez Pomar

Juan Camilo Vasquez

Punto 1

1	void algoritmo1(int n){	
2	int i, j = 1;	1
3	for(i = n * n; i > 0; i = i / 2){	log2(n^2)+2
4	int suma = i + j;	log2(n^2)+1
5	printf("Suma %d\n", suma);	log2(n^2)+1
6	++j;	log2(n^2)+1

1 + log2(n^2)+2 + log2(n^2)+1+ log2(n^2)+1+ log2(n^2)+1

T(n)= O(log2 n^2)

Como podemos ver, es un algoritmo de complejidad O cuadrática

Cuando ejecutamos algoritmo1(int 8) pasaría lo siguiente:

1	void algoritmo1(int 8){	
2	int i, j = 1;	1
3	for(i = 8 * 8; i > 0; i = i / 2){	log2(8^2)+2
4	int suma = i + j;	log2(8^2)+1
5	printf("Suma %d\n", suma);	log2(8^2)+1
6	++j;	log2(8^2)+1

Esto quiere decir, que empezaría elevando 8 al cuadrado, nos da 64, luego le suma el valor de j, que es 1, nos imprimiría 65, luego le aumenta 1 a j.

Por lo cual, el siguiente valor que imprime es:

Suma 35

Suma 19

Suma 12

Suma 9

Suma 8

Suma 8

punto 6)

punto 2)

Tamaño Entrada	Tiempo	Tamaño Entrada	Tiempo
5	0.224sec	35	4.645sec
10	0.135sec	40	49.399secs
15	0.13sec	45	4m16.749s
20	0.138sec	50	NA
25	0.186sec	60	NA
30	0.581sec	100	NA

es O(2^n).

Esto se debe a que la función de Fibonacci recursiva llama a sí misma dos veces para calcular cada número de Fibonacci. Por lo tanto, el número total de llamadas recursivas aumenta exponencialmente con respecto a "n".

El valor mas alto fue de 4 minutos y algunos segundos, pienso que esta solución con recursión es muy ineficiente, debido a que necesita mucho tiempo para calcular.

Punto 7)

Tamaño Entrada	Tiempo	Tamaño Entrada	Tiempo
5	0m0.266s	45	0m0.110s
10	0m0.136s	50	0m0.135s
15	0m0.131s	100	0m0.118s
20	0m.132s	200	0m0.117
25	0m0.137s	500	0m0.110s
30	0m0.117s	1000	0m0.114s
35	0m0.113s	5000	0m0.034s
40	0m0.118s	10000	0m0.033s

```
def fibonacci(n):
    if n < 0:
        print("El número debe ser mayor o igual a cero.")
    elif n == 0:
        return 0
    elif n == 1 or n == 2:
        return 1
    else:
        a, b = 1, 1
        for i in range(3, n + 1):
            c = a + b
            a, b = b, c
        return b
print(fibonacci(10000))
```

Punto 8)

Tamaño Entrada	Tiempo Solución Propia	Tiempo Solución Profesores
100	0m0.036s	0m0.040s
1000	0m0.046s	0m0.037s
5000	0m0.086s	0m0.060s
10000	0m0.103s	0m0.053s
50000	0m0.524s	0m0.154s
100000	0m1.297s	0m0.307s
200000	0m3.203s	0m3.713s