Tarea 6

Pablo Antonio Stack Sánchez
Programación y algoritmos

29 de septiembre de 2019

1. Problema 1

El peor caso para el algoritmo BubleSort es que se encuentre ordenado al revés. Si esto pasa el while() tendría que dar n iteraciones, si multiplicamos esto por las n iteraciones fijas del for() daría una complejidad de $O(n^2)$.
El mejor caso es cuando ya se encuentra ordenado o cuando puede ser ordenado en una solo iteración del while, en tal caso el algoritmo tendría complejidad $O(n)$.

2. Problema 2

La operación $i >>= 1$ recorre un bit a la derecha el número, esto en decimal equivale a una división por 2. Por lo tanto la complejidad del algoritmo será $O(log(n))$.

El siguiente algoritmo toma valores en $S = 1, 3, 6, 10, 15, 21$. Para calcular un valor k de S se debe utilizar la formula:

$$\frac{k(k+1)}{2} = n$$

Despejamos y calculamos las raices:

$$k^2 + k - 2n = 0$$

$$k = \frac{-1 \pm \sqrt{1 + 4(2n)}}{2}$$

De aqui se observa que el algoritmo tendrá complejidad $O(\sqrt{n})$

3. Problema 3

Ver Pablo_Stack_Tarea06_p3.c

4. problema 4

Ver pila.c

5. problema 5

El de la complejidad más pequeña es:

$$\sum_{i=1}^{n} \frac{1}{i}$$

ya que es una serie armonica y tiende a 1. Por lo tanto la complejidad sería $O(1)$
La segunda es:

$$n + log(n)$$

Ya que todas las demás tienen terminos elevados a alguna potencia. Podría haber dudas de cual es más pequeña si $n+log(n)$ o $4^log2(n)\sqrt(n-3)$ A continuación se prueba que $n+log(n)$ es menos complejo.
Se realiza un cambio de base:

$$4^{\frac{log_4(n)}{log_4(2)}}\sqrt{n-3}$$

simplificando:

$$n^2\sqrt{n-3}$$

$$n^2\sqrt{n}$$

$$n^{\frac{5}{2}}$$

Que es $O(n^2)$. El siguiente es:

$$n^3|cos(n)|$$

Ya que la unica complejidad que se le parece es la de $n^3log_5(n)$ pero $|cos(n)|$ solo toma valores entre 0 y 1 por lo que la complejidad se reduce a $O(n^3)$. por lo tanto el siguiente valor es:

$$n^3log_5(n)$$

y por ultimo:

$$n!$$