

Nombre: _____ Rut: _____

Estructuras de datos Control 1

Prof: Fabián Riquelme Csori
2017-II

1. Considere tres algoritmos A_1 , A_2 y A_3 que se ejecutan en tiempo $T_1 = 2n^3$, $T_2 = O(\frac{1}{2}n^2)$ y $T_3 = O(2^n)$, respectivamente. Fundamente:
 - a) Si A_1 y A_2 se ejecutan secuencialmente en una misma máquina, ¿cuál es el tiempo de ejecución total, en notación big O? [10 pts]
 - b) Si A_1 , A_2 y A_3 se ejecutan en paralelo en tres máquinas de iguales características, ¿cuál es el tiempo de ejecución total, en notación big O? [10 pts]
 - c) En el caso b), ¿es posible que la cantidad de memoria total que se ocupe entre las tres máquinas sea lineal, es decir, $S(n)$? [10 pts]
2. Reproduzca los dos siguientes algoritmos escritos en lenguaje C en su computadora (para compilar: `gcc -o archivo_destino archivo_fuente.c`).

A_1 : <pre>#include <stdio.h> int main(){ int n=100000; //10⁵ int a[n]; for(int i=0; i<n; i++){ a[i]=2*i; } for(int i=0; i<n; i++){ printf("%d tiene %d\n",i,a[i]); } return(0); }</pre>	A_2 : <pre>#include <stdio.h> int main(){ int n=1000000; //10⁶ int a[2*n]; for(int i=0; i<2*n; i++){ a[i]=2*i; } return(0); }</pre>
--	--

- a) ¿Cuántas veces itera cada uno de los tres “for”? [5 pts]
- b) ¿Cómo calcula el tiempo de CPU de ejecución de cada algoritmo? Calcule. [10 pts]
- c) ¿Cuál es el tiempo de ejecución en notación big O de cada algoritmo? [10 pts]
- d) ¿Qué algoritmo es más eficiente en términos de complejidad temporal? [5 pts]

Estructuras de datos Control 1 - Pauta

Prof: Fabián Riquelme Csori
2017-II

1. a) Los tiempos se deben sumar en notación big O. [3 pts]
El tiempo total será $2n^3 + O(\frac{1}{2}n^2) = O(2n^3 + \frac{1}{2}n^2)$, [2 pts]
es decir, $O(n^3)$. [5 pts]
1. b) Los tiempos se deben sumar en notación big O (o bien quedarme con el máximo). [2 pts]
El tiempo total será $2n^3 + O(\frac{1}{2}n^2) + O(2^n) = O(2n^3 + \frac{1}{2}n^2 + 2^n)$, [2 pts]
es decir, $O(2^n)$. [6 pts]
1. c) Sí, podría darse el caso que los algoritmos tengan muchas instrucciones que solo utilizan un espacio muy reducido de memoria. Como vimos en clases, si el tiempo de ejecución de un algoritmo es polinomial o incluso exponencial, entonces el espacio de memoria puede ser polinomial (y un espacio lineal es un espacio polinomial de grado 1). [10 pts]
2. a) Cada **for** de A_1 itera $n = 10^5$ veces. [2 pts]
El **for** de A_2 itera $2n = 2 \times 10^6$ veces. [3 pts]
2. b) Puede variar entre cada máquina, pero debe ser el resultado **user** que retorna la ejecución con el comando **time**. [5 pts]
Escribir el resultado obtenido para A_1 y para A_2 . [5 pts]
2. c) Para A_1 es $O(n + n) = O(2n) = O(n)$. [5 pts]
Para A_2 es $O(2n) = O(n)$. [5 pts]
2. d) De lo anterior se concluye que A_1 y A_2 tienen la misma complejidad temporal. [5 pts]