

Auxiliar 4 - Ecuaciones de Recurrencia y Teorema Maestro

Profesores: Nelson Baloian, Patricio Poblete
Benjamín Bustos
Sebastian Ferrada

Auxiliares: Vicente Olivares, Ricardo Valdivia
Sebastián Acuña
Valentina Aravena

P1. Ecuaciones de recurrencia

Resuelva las siguientes ecuaciones de recurrencia:

a) Ecuación lineal homogénea con coeficientes constantes

$$T(n) = 3T(n - 1) + 4T(n - 2), \forall n \geq 2$$

$$T(0) = 1$$

$$T(1) = 1$$

b) *Hint*: Utilice cambio de variable

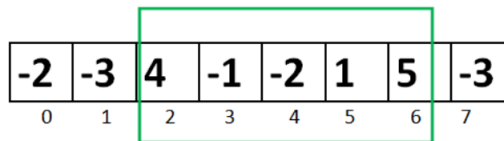
$$X_{n+1} = \sqrt{X_n \cdot X_{n-1}}$$

$$X_0 = 1$$

$$X_1 = 2$$

P2. Subarreglo de suma máxima

Dado un arreglo con enteros positivos y negativos, se quiere conseguir el subarreglo que contiene la suma máxima posible. Un subarreglo es cualquier subconjunto consecutivo del arreglo. Por ejemplo si el arreglo es $[-2, -3, 4, -1, -2, 1, 5, -3]$ el resultado sería el subarreglo $[4, -1, -2, 1, 5]$ de suma total 7.



$$4 + (-1) + (-2) + 1 + 5 = 7$$

- Programe una solución que resuelva el problema, sin importar cuán eficiente es.
- ¿Puede ser más eficiente la solución? Intente mejorar la solución anterior.
- Intente mejorar la solución anterior usando una estrategia dividir para reinar.

(d) Sea aún más eficiente: programe una solución que resuelva el problema en tiempo lineal.

Hint: La clave está en los números que son negativos

Para todos los casos anteriores entregue cuál es su costo en notación Big-O.

P3. Teorema Maestro

Suponga que para resolver un determinado problema se le presentan las siguientes tres alternativas de algoritmo:

- El algoritmo A resuelve el problema de tamaño n dividiéndolo en cinco subproblemas de tamaño $\frac{n}{2}$, resolviendo recursivamente cada subproblema y luego combinando las soluciones en tiempo lineal.
- El algoritmo B resuelve el problema de tamaño n resolviendo recursivamente dos subproblemas de tamaño $n - 1$ y realizando trabajo adicional a tiempo constante.
- El algoritmo C resuelve el problema de tamaño n dividiéndolo en nueve subproblemas de tamaño $\frac{n}{3}$, resolviendo recursivamente cada subproblema y luego combinando las soluciones en tiempo $O(n^2)$.

¿Qué algoritmo escogería usted si el objetivo es minimizar el tiempo de ejecución?