1. Divide y Vencerás

1. Construya un algoritmo "Divide-y-vencerás que permita calcular el mínimo elemento de un arreglo de N números enteros, donde N es una potencia de 2. Calcule la complejidad de su algoritmo.

2. Construya un algoritmo "Divide-y-vencerás que permita calcular los elementos mínimo y máximo de un arreglo de N números enteros, al mismo tiempo, donde N es una potencia de 2. Calcule la complejidad de su algoritmo.

3. Construya un algoritmo "Divide-y-vencerás que permita calcular la inversa de una Matriz de NxN, donde N es una potencia de 2. Puede asumir la existencia de ciertas funciones sin implementarlas. Calcule la complejidad de su algoritmo.

4. Dado un arreglo ordenado, construya un algoritmo "Divide-y-vencerás que encuentre el elemento del arreglo que mas veces se repite, y calcule la complejidad de su algoritmo.

5. Dado un arreglo que consiste de 2 partes, una parte donde los números son crecientes, y una parte de números decrecientes. Construya un algoritmo "Divide-y-vencerás que encuentre el índice del numero que corresponde a la transición entre la parte creciente y la parte decreciente. Calcule la complejidad de su algoritmo.

6. Use el Algoritmo de Multiplicación de Matrices de Strassen para calcular los siguientes productos:

   a)
   $$\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$$

   b)
   $$\begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 5 & -2 \end{pmatrix}$$

   c)
   $$\begin{pmatrix} 3 & 1 & 3 \\ 2 & 2 & 1 \\ 1 & 2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 5 & -2 & 1 \\ 2 & 2 & -1 \end{pmatrix}$$

2. Teorema Maestro

1. Usando el teorema maestro, de cotas asintóticas (Notación $O - \Theta - \Omega$) de la solución a las siguientes ecuaciones de recurrencia:

   a) $T(n) = 9T\left(\frac{n}{3}\right) + n^2$
   b) $T(n) = 8T\left(\frac{n}{2}\right) + n^3$
   c) $T(n) = 49T\left(\frac{n}{25}\right) + n^{\frac{3}{2}} \log n$
   d) $T(n) = 9T\left(\frac{n}{3}\right) + n^2$
   e) $T(n) = 2T\left(\frac{n}{3}\right) + 1$
   f) $T(n) = 5T\left(\frac{n}{4}\right) + n \log n$
   g) $T(n) = 8T\left(\frac{n}{2}\right) + 4n^3$
   h) $T(n) = 2T\left(\frac{n}{4}\right) + 1$
   i) $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$
   j) $T(n) = 2T\left(\frac{n}{4}\right) + n$
   k) $T(n) = 2T\left(\frac{n}{4}\right) + n^2$
   l) $T(n) = 3T\left(\frac{n}{4}\right) + n \log n$
   m) $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$

2. Use el teorema maestro para mostrar que la solución de la ecuación de recurrencia de la búsqueda binaria:

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$$

Es $T(n) \in \Theta(\log n)$

3. Es posible aplicar el teorema maestro a la ecuación de recurrencia:

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2 \log n$$

Si es así, cual es la solución de dicha ecuación?

3. Programación Dinámica

1. Los números de Stirling del primer tipo están definidos para $k > 0$ como:

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = n \begin{bmatrix} n \\ k \end{bmatrix} + \begin{bmatrix} n \\ k-1 \end{bmatrix}$$

Con las condiciones iniciales para $n > 0$:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1, \qquad \begin{bmatrix} n \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ n \end{bmatrix} = 0$$

Construya un algoritmo recursivo y un algoritmo recursivo usando programación dinámica que permita calcular $\begin{bmatrix} n \\ k \end{bmatrix}$. Calcule la complejidad algorítmica de ambos algoritmos.

2. Los números de Stirling del segundo tipo están definidos para $k > 0$ como:

$$\begin{Bmatrix} n+1 \\ k \end{Bmatrix} = k \begin{Bmatrix} n \\ k \end{Bmatrix} + \begin{Bmatrix} n \\ k-1 \end{Bmatrix}$$

Con las condiciones iniciales para $n > 0$:

$$\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = 1, \qquad \begin{Bmatrix} n \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ n \end{Bmatrix} = 0$$

Construya un algoritmo recursivo y un algoritmo recursivo usando programación dinámica que permita calcular $\begin{Bmatrix} n \\ k \end{Bmatrix}$. Calcule la complejidad algorítmica de ambos algoritmos.

3. La función de Ackermann esta definida como:

$$A(m,n) = \begin{cases} n+1 & \text{si m = 0} \\ A(m-1,1) & \text{si m ¿0 y n = 0} \\ A(m-1, A(m,n-1)) & \text{si m ¿0 y n ¿0} \end{cases}$$

Construya un algoritmo recursivo y un algoritmo recursivo usando programación dinámica que permita calcular la función de Ackermann.

4. Los números eulerianos de primer orden, E_1, están definidos de la siguiente forma:

$$E_1(n,k) = (k+1)E_1(n-1,k) + (n-k)E_1(n-1,k-1)$$
$$E_1(n,0) = E_1(n,n-1) = 1, \qquad E_1(n,1) = 2^n - n - 1$$
$$E_1(0,k) = \begin{cases} 1 & \text{si k = 0} \\ 0 & \text{En el caso contrario} \end{cases}$$

Construya un algoritmo recursivo y un algoritmo recursivo usando programación dinámica que permita calcular dichos números.

5. Los números eulerianos de primer orden, E_2, están definidos de la siguiente forma:

$$E_2(n,k) = (k+1)E_2(n-1,k) + (2n-1-k)E_2(n-1,k-1)$$
$$E_2(n,0) = 1, \qquad E_2(n,n) = 0 \text{ para } n \neq 0$$

   Construya un algoritmo recursivo y un algoritmo recursivo usando programación dinámica que permita calcular dichos números.

6. Construya un algoritmo recursivo que use programación dinámica para calcular a^n, con n un numero entero positivo.

Los siguientes problemas de Optimización se deben resolver usando programación dinámica, primero planteando una relación recursiva que permita resolver el problema, y luego evaluar dicha relación usando programación dinámica:

1. Una empresa de construcción tiene 4 proyectos en progreso. Los 4 proyectos pueden ser completados en 15, 20, 18 y 25 semanas según las asignaciones de recursos actuales. La gerencia quiere reducir los tiempos de finalización y ha decidido asignar 35 unidades de dinero adicionales a los 4 proyectos. Los nuevos tiempos de finalización en función de los recursos asignados son:

Financiamiento Adicional	Proyecto 1	Proyecto 2	Proyecto 3	Proyecto 4
0	15	20	18	25
5	12	16	15	21
10	10	13	12	18
15	8	11	10	16
20	7	9	9	14
25	6	8	8	12
30	5	7	7	11
35	4	7	6	10

   Como deberían asignarse las 35 unidades de dinero de tal forma que el tiempo total de finalización de los 4 proyectos sea mínima? El dinero solo puede ser asignado en bloques de 5 unidades.

2. La siguiente tabla especifica el peso y valor de 5 productos.

Producto	Peso	Valor
1	7	9
2	5	4
3	4	3
4	3	2
5	1	1

   Se tiene una camioneta con una capacidad de 13 unidades de peso que transportara productos. Como debería ser cargada la camioneta con el fin de maximizar el valor total cargado?

3. Un estudiante de Pregrado cuenta con 7 días para estudiar para 4 exámenes finales: De Computación Paralela (CP), de Optimización de Sistemas (OS), de Auditoría de Sistemas (AS), y de Gestión Informática (GI). La nota esperada por cada día que estudia cada asignatura esta dada por:

Días de Estudio	Nota CP	Nota OS	Nota AS	Nota GI
1	2	4	1	4
2	3	4	2	5
3	4	5	4	6
4	5	6	7	6

   Como debe asignar el estudiante los 7 días disponibles de estudio para obtener una nota total máxima?

4. Pedrito Caracol esta en campaña política para obtener un puesto de elección popular. Su partido político posee 6 voluntarios para cubrir 4 barrios diferentes. Se espera que por cada asignación de voluntarios en los barrios, se obtengan la siguiente cantidad de votos:

Voluntarios	Barrio A	Barrio B	Barrio C	Barrio D
0	0	0	0	0
1	4	7	5	6
2	9	11	10	11
3	15	16	15	14
4	18	18	18	16
5	22	20	21	17
6	24	21	22	18

Como debe Pedrito (C) asignar los voluntarios a los 4 barrios con tal de maximizar la cantidad de votos que gane?

4. Clasificación de Problemas

1. Porque $P \subset NP$? Cual es la lógica que lleva a concluir que $P \subset NP$?
2. Si asumimos que P = NP, cual es la complejidad algorítmica de un algoritmo para colorear grafos?
3. Consideremos un algoritmo de fuerza bruta, el cual prueba todas las posibles soluciones de un problema, hasta encontrar la correcta. Cual es la complejidad algorítmica de un algoritmo de fuerza bruta que resuelva los siguientes problemas?:
   a) Coloreado de Grafos.
   b) SAT.
   c) Vendedor viajero.
   d) Programación Lineal Entera.
   e) Programación Lineal Booleana.