Diseño y Análisis de Algoritmos
2º Grado Ingeniería del Software

1. Sea $E[1..n]$ un vector ordenado de enteros todos distintos. Se pide implementar un algoritmo capaz de encontrar un índice i tal que $1 \leq i \leq n$ y $E[i] = i$, suponiendo que tal índice exista. En caso contrato, el algoritmo debe devolver -1. La complejidad del método desarrollado debe ser $O(log\ n)$, en el caso peor.

   Ejemplo: Dado el vector $E = (-3, -1,0,3,7,9,12,14,16,2\ 5)$, el método devolvería 3.

2. Usando la técnica divide y vencerás, resolver el problema que calcule la mediana de un *array* de n elementos con un algoritmo con complejidad lineal. ¿Qué mejora se obtiene respecto a una versión iterativa? ¿Existe alguna situación en la que sea mejor la solución con divide y vencerás?

   Ejemplo: Dado el *array* $(0,5,1,2,7,0,3,4,6,5)$ su mediana es $(4+3)/2 = 3.5$.

   NOTA: Se recuerda que la mediana de un vector de k elementos es aquél que ocupa la posición $(k+1)/2$ cuando el vector está ordenado de forma creciente.

3. Dadas dos secuencias ordenadas de n y m enteros, respectivamente. Diseñe un algoritmo Divide y Vencerás, eficiente, que calcule:
   a. La mediana de la mezcla resultante de las dos secuencias dadas.
   b. El k-ésimo elemento de la mezcla resultante de las dos secuencias dadas.

   En ambos casos, calcular la complejidad del algoritmo desarrollado.

4. Diseñar un algoritmo con complejidad lineal para calcular los k mayores elementos de un *array* de n enteros, utilizando la técnica divide y vencerás. Calcular el número de comparaciones realizadas en el mejor y en el peor caso, suponiendo que n potencia de 2. Comprueba si es posible aplicar los resultados obtenidos (en cuanto a órdenes de ejecución) a valores de n que no sean potencia de 2.

   Ejemplo: Dado el *array* $(0,5,1,2,7,0,3,4,6,5)$ y $k = 2$, el método debería devolver el par $(7,6)$.

5. Construya un procedimiento que seleccione el k -ésimo menor elemento de un vector de n elementos. La complejidad del procedimiento debe ser lineal.

6. Dado un *array* $E[1..n]$, se dice que un elemento es mayoritario en E si aparece más de $n/2$ veces. Diseñar e implementar un algoritmo para calcular si un *array* E contiene un elemento mayoritario. El algoritmo debe tener complejidad lineal en el caso peor.

   Ejemplo: Dado el *array* $(0,5,1,2,7,0,3,4,6,5)$, el método no tiene ningún elemento mayoritario.

7. Diseñar e implementar un método de ordenación de un *array* similar a la ordenación por mezcla, pero dividiendo el problema en 3 sub-problemas (de tamaños parecidos) en vez de en 2.

8. Sean X e Y dos vectores sin elementos repetidos de dimensión n. Se sabe que X está ordenado crecientemente y que Y lo está decrecientemente, al menos uno en sentido estricto. Diseñe un algoritmo que determine en tiempo logarítmico si hay un índice i tal que $X[i] = Y[i]$.

9. Un vector de enteros de dimensión impar, se dice que es "*centradito*" si sus tres elementos centrales después de ordenarlo son consecutivos. Desarrolle una estrategia para decidir en tiempo lineal si un vector dado es "*centradito*".

10. Se consideran un vector v de n elementos y un entero positivo k. Diseñe algoritmos basados en Divide y Vencerás con complejidad lineal que devuelvan los k elementos:
    a. Centrales de v; es decir, los k elementos que ocupan las posiciones de $n/2 - k/2$ a $n/2 + k/2$.
    b. Más próximos (en valor) a su mediana.
    c. Más alejados (en valor) de su mediana

11. Se consideran un vector v de n elementos y dos enteros positivos k y p. Diseñe un algoritmo con complejidad lineal que devuelva los p elementos de v más próximos al k-ésimo:
    a. Menor elemento de v.
    b. Mayor elemento de v.

12. Dada una matriz booleana, se pide resolver el problema de encontrar el cuadrado de 1s más grande en una tabla cuadrada $n \times n$ utilizando un método:
    a. Directo para resolver el problema.
    b. Basado en divide y vencerás.

    En ambos casos, se pide calcular una cota superior (no demasiado mala) de su tiempo de ejecución.

13. Construya un algoritmo Divide y Vencerás que ordene las filas de la matriz de enteros según el número de 1s que tienen, sin superar la complejidad $O(n \log n)$. No considere la complejidad que supone intercambiar filas.

14. Diseñar e implementar un algoritmo para calcular el producto de dos matrices cuadradas triangulares superiores, siendo el número de filas y columnas potencia de 2. Obtener la complejidad algorítmica de método propuesto.

    NOTA: se recuerda que una matriz es triangular superior si todos los elementos que están por debajo de la diagonal principal valen 0. Simétricamente, se dice que es triangular inferior si todos los elementos encima de la diagonal principal valen 0. Ejemplo de matriz triangular superior:

$$\begin{pmatrix} 1 & 3 & 5 \\ 0 & 2 & 4 \\ 0 & 0 & 7 \end{pmatrix}$$

15. Diseñe un algoritmo basado en la técnica Divide y Vencerás que calcule la traspuesta de una matriz de dimensión $2n$.

16. Sea A una matriz cuadrada de tamaño $2n$, con $n > 0$. Utilizando la estrategia de divide y vencerás, diseñe un algoritmo para rotarla 90º en el sentido de las agujas del reloj.

    Ejemplo. La matriz A se convertirá en A'

$$A = \begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix} \qquad A' = \begin{pmatrix} G & D & A \\ H & E & B \\ I & F & C \end{pmatrix}$$

17. Encontrar un algoritmo eficiente para calcular a^n suponiendo que el coste de multiplicar dos enteros es proporcional a su tamaño.

18. Dados n enteros cualesquiera $a_1, a_2, \ldots, a_n$, necesitamos encontrar el valor de la expresión:

$$\max_{1 \le i \le j \le n} \left(\sum_{k=i}^{j} a_k \right)$$

    que calcula el máximo de las sumas parciales de elementos consecutivos. Por ejemplo, dados los 6 enteros $(-2, 11, -4, 13, -5, -2)$ la solución al problema es 20 (suma de a_2 hasta a_4). Implementar un algoritmo Divide y Vencerás que resuelva el problema.

19. Se tienen n puntos en el plano (pares de coordenadas x e y). Diseñe un algoritmo Divide y Vencerás que devuelva el par de puntos cuya distancia entre ambos es mínima.

20. Los organizadores de la maratón de Madrid, en la que participaron miles de atletas, publicaron los resultados ordenados por el número de dorsal con el que compitieron en la carrera. Diseñe algoritmos con complejidad lineal en el peor de los casos para conocer:

    a. Los cien primeros clasificados, no necesariamente ordenados.
    b. Los diez primeros clasificados, éstos si ordenados.

21. La liga de eSports URJC quiere organizar el calendario de un campeonato entre n jugadores del modo arena de *Dark Soluls III*. Cada uno ha de jugar exactamente una vez contra cada adversario. Además, cada jugador ha de jugar exactamente una partida diaria, con la excepción posible de un sólo día en el que no jugará.

    a. Si n es potencia de dos, diseñe un algoritmo que permita terminar el campeonato en $n - 1$ días.
    b. Siendo n cualquiera, diseñe un algoritmo que planifique el campeonato en n días (n es impar), o en $n - 1$ días (n es ar y mayor que 1).

22. Dadas la cadena ordenada X con K símbolos e Y, con dos símbolos (también ordenados entre sí), se desea obtener un índice j tal que $x_j = y_1$ e $x_{j+1} = y_2$. En caso de no existir, se devolverá $j = -1$. Diseñar un algoritmo basado en Divide y Vencerás con complejidad logarítmica.

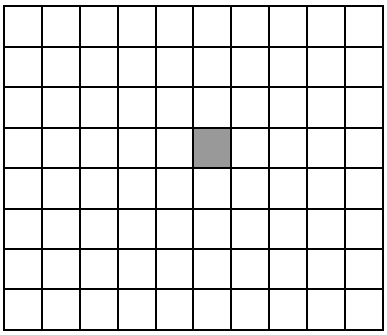
    Ejemplo: Dada la cadena $x =$ "$ADEFHPSTW$", si $y =$ "HP", el método devolvería $j = 5$; por otro lado, si $y =$ "HT", $j = -1$.

23. En una apartada localidad se ha realizado una encuesta para determinar cuál es la palabra más popular. Resulta que esa palabra puede ser inventada, y no estar recogida en el diccionario, ni en ninguna fuente. Los lugareños son de ideas fijas, y de vocabulario contagioso, así que por encuestas anteriores realizadas en la localidad se sabe que cuando una palabra se pone de moda entre ellos, entonces al menos un tercio de la población la elige. Diseñe un algoritmo con complejidad lineal que dado el resultado de la encuesta (la palabra elegida por cada entrevistado) determine la palabra de moda, si es que ésta existe.

24. En una habitación oscura se tienen dos cajones en uno de los cuales hay n tornillos de varios tamaños, y en el otro las correspondientes n tuercas. Es necesario emparejar cada tornillo con su tuerca correspondiente, pero debido a la oscuridad no se pueden comparar tornillos con tornillos ni tuercas con tuercas, y la única comparación posible es la de intentar enroscar una tuerca en un tornillo para comprobar si es demasiado grande, demasiado pequeña, o se ajusta perfectamente al tornillo. Desarrolle un algoritmo Divide y Vencerás para emparejar los tornillos con las tuercas, que use $O(n \log n)$ comparaciones en término medio.

25. En un jardín de infancia los niños se disponen a dormir la siesta después de comer. Para ello, el profesor les quita previamente los zapatos uno a uno y los guarda emparejados, por ejemplo, atando los cordones de cada par. Una vez terminada la siesta, el profesor debe poner de nuevo los zapatos a todos sus alumnos. El problema es que todos los zapatos son del mismo modelo y color, que es el del uniforme del jardín de infancia. Cuando intenta colocar un par de zapatos a un niño puede suceder que el par encaje perfectamente, que le queden grandes, o que le queden pequeños. El objetivo del profesor es que todos encajen perfectamente. Diseñe un algoritmo Divide y Vencerás que resuelva el problema suponiendo que cada par de zapatos solamente le sirve a un niño, que no use un número de comparaciones superior a $O(n \log n)$ en término medio.

26. Tenemos un tablero cuadrado de dimensión 2*m* (tiene, pues 2^{2^m} celdas) y una tesela en forma de L, como puede observarse en la figura. Diseñe un algoritmo Divide y Vencerás, que cubra todo el tablero utilizando teselas en forma de L como las de la figura, supuesto que ya tenemos cubierta una casilla.



27. Solid Snake es aficionado al juego, de tal forma que todas las noches va a jugar una partida de póker. Como es un jugador responsable, para controlar sus pérdidas, anota lo que gana o pierde cada noche como un número entero de euros. Por ejemplo, en el último mes los resultados han sido los siguientes:

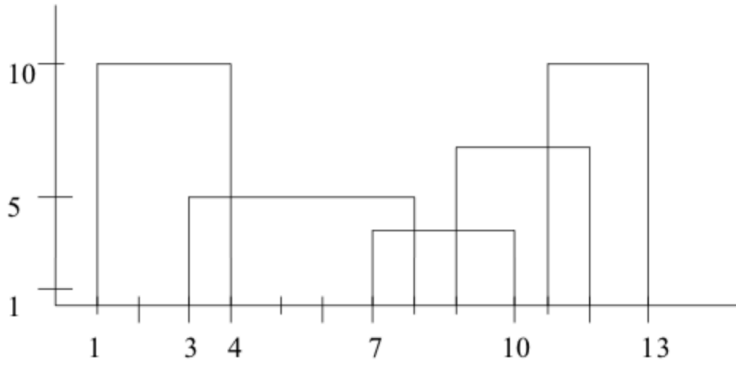
$$29, -7, 14, 21, 30, 47, 1,$$
$$-7, -39, 23, -20, -36, -41, 27,$$
$$-34, 7, 48, 35, -46, -16, 32,$$
$$18, 5, -33, 27, 28, -22, 1,$$
$$-20, 3, -42$$

Como se puede ver, Solid Snake empezó bien el mes, con una ganancia de 29 euros, pero terminó con una pérdida de 42 euros. El beneficio total obtenido a lo largo de todo el mes es -47 euros. Analizando esta información en retrospectiva, un algoritmo de Divide y Vencerás habría detectado que, si hubiera empezado a jugar el día 16 y terminado el día 26, habría maximizado sus ganancias, obteniendo 105 euros, una diferencia de 155 euros con respecto a su situación actual.

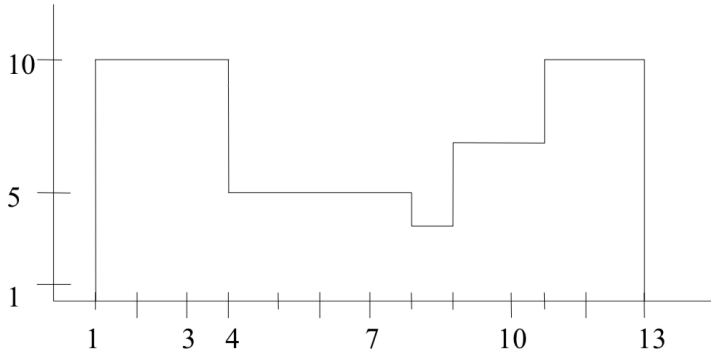
Dado un vector de ganancias/pérdidas, de longitud n, utilizando la técnica Divide y Vencerás, se trata de devolver la secuencia de días (consecutivos) con los que se consigue maximizar el beneficio total, devolviendo los índices de comienzo y fin, y ese beneficio máximo. Calcule la complejidad del algoritmo desarrollado.

28. Dada la localización exacta y la altura de varios edificios rectangulares de la ciudad, obtener su *skyline* (línea de recorte contra el cielo que producen todos los edificios eliminando las líneas ocultas).

Ejemplo: La entrada es una secuencia de edificios, donde un edificio se caracteriza por una 3-tupla de valores (x_{min}, x_{max}, h). Una posible secuencia de entrada podría ser: $(7,10,3), (9,12,7), (1,4,10), (3,8,5), (11,13,10)$, cuya representación gráfica Se muestra en la siguiente figura



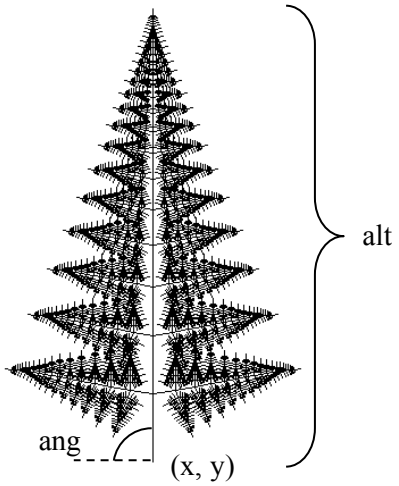
La secuencia de salida asociada debería ser: $(1,4,10), (4,8,5), (8,9,3), (9,11,7), (11,13,10)$, cuya representación gráfica se muestra en la siguiente figura:



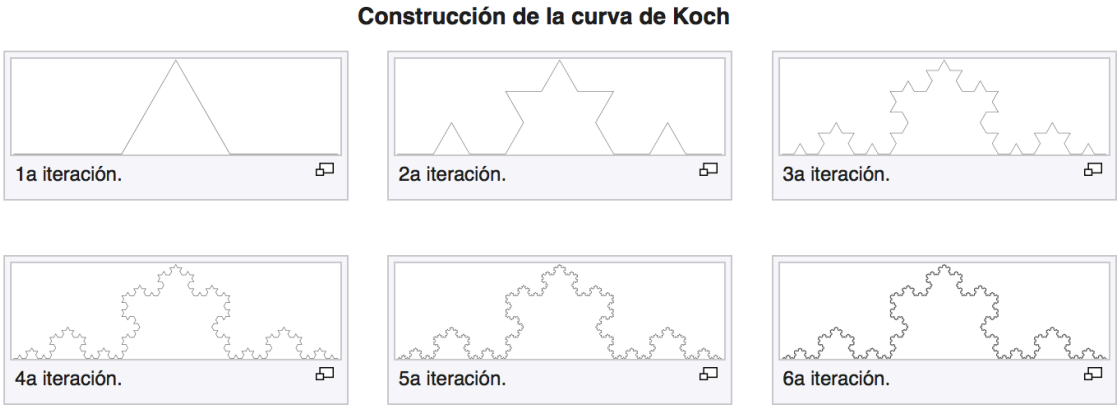
29. La figura de abajo representa el "helecho", una curva fractal que puede ser generada mediante una variedad de procedimientos. En su versión más simple, es posible utilizar un método de divide y vencerás para dibujarlo. Suponer que la cabecera es

$$dibujarHelecho(x, y: int;\ ang, alt: float);$$

que representa un helecho de altura alt, cuyo tallo principal tiene la base en (x, y) y tiene ángulo ang respecto a la horizontal. Escribir un procedimiento para dibujar el helecho, a partir de los parámetros anteriores, utilizando la técnica divide y vencerás.

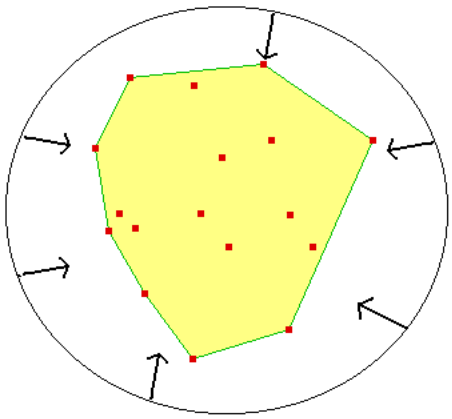


30. La figura fractal "copo de nieve" se construye mediante un proceso que se inicia partiendo en tres un segmento de recta. A continuación, se insertan dos segmentos más en del medio a manera de un triángulo equilátero. El proceso se puede repitir de forma indefinida.



Diseñe un algoritmo Divide y Vencerás que la dibuje este fractal. Deducir la complejidad del algoritmo propuesto.

31. La envolvente convexa, también denominada cierre convexo o *convex hull*, es uno de los fundamentales constructores geométricos con muchas aplicaciones parácticas. Una idea intuitiva del significado del cierre convexo es el contenido de la figura que formaría una banda elástica que rodeara a una nube de puntos una vez que la soltáramos. En la siguiente figura, se representa un ejemplo gráfico.



Se pide diseñar un algoritmo Divide y Vencerás que calcule el cierre convexo y la complejidad del algoritmo desarrollado.