ALGORITMIA Y COMPLEJIDAD

LABORATORIO - TEMA 3

EJERCICIO 1

Se tiene un vector $V[1..N]$ formado por números enteros, de manera que todos ellos distintos, y que están ordenados de manera creciente. Se dice que un vector de estas características es coincidente si tiene al menos una posición tal que es igual al valor que contiene el vector en esa posición.

Por ejemplo, en el vector:

-14	-6	3	6	16	18	27	43
1	2	3	4	5	6	7	8

puede verse que $V[3] = 3$; por lo tanto, este vector es coincidente.

Diseñar un algoritmo Divide y Vencerás que determine, en un orden de eficiencia no superior a $O(log\ n)$, si un vector es coincidente.

EJERCICIO 2

Se tiene un vector $V[1..N]$ del que se quiere obtener información como si estuviese ordenado, pero que no puede ordenarse por motivos de trabajo (tampoco pueden realizarse copias en memoria del vector). Para poder hacerlo se quiere crear un vector de índices $Ind[1..N]$ de tal manera que el valor $Ind[i]$ sea precisamente la posición del vector original con el dato que debería estar en la posición i si efectivamente estuviese ordenado.

Por ejemplo, si el vector $V[1..N]$ fuese:

50	98	10	63	31	25	63	74
1	2	3	4	5	6	7	8

entonces el vector de índices $Ind[1..N]$ sería:

3	6	5	1	4	7	8	2
1	2	3	4	5	6	7	8

ya que $Ind[1] = 3$ es la posición de V donde está 10, que es el elemento que quedaría primero si se ordenase V, $Ind[2] = 6$ es la posición de 25, que sería el segundo elemento en el vector ordenado, etc.

Modificar el algoritmo de ordenación por mezcla MergeSort para obtener el vector de índices de un vector $V[1..N]$ dado como parámetro.

EJERCICIO 3

Se tiene acceso a una función $f(x)$ de la que se sabe que en el intervalo real $[p_1, p_2]$ tiene un único mínimo local en el punto x_0, que es estrictamente decreciente entre $[p_1, x_0]$ y que es estrictamente creciente entre $[x_0, p_2]$. Hay que observar que x_0 puede coincidir con p_1 o con p_2.

Se tiene que buscar, de la manera más eficiente posible, todos los puntos x (si es que existen) del intervalo $[p_1, p_2]$ tales que la función f tome un cierto valor k, es decir, se busca el conjunto de valores $\{ x \in [p_1, p_2]$ tal que $f(x) = k \}$. Para simplificar el proceso, en vez del valor exacto de cada x puede indicarse un intervalo de valores $[\alpha, \beta]$, con $\beta - \alpha < \varepsilon$, donde se encuentre x. Los datos del algoritmo serán el intervalo $[p_1, p_2]$, el valor k que se está buscando, y el valor ε para la aproximación.

EJERCICIO 4

Se quiere programar un robot para poner tapones de corcho a las botellas de una fábrica de reciclado. Se tienen disponibles N botellas y los N corchos que las tapan (N es constante en el problema), pero hay una serie de problemas:

- Las botellas son todas distintas entre sí, igual que los corchos: cada botella solo puede cerrarse con un corcho concreto, y cada corcho solo sirve para una botella concreta.
- El robot está preparado para cerrar botellas, por lo que lo único que sabe hacer es comparar corchos con botellas. El robot puede detectar si un corcho es demasiado pequeño, demasiado grande, o del tamaño justo para cerrar una botella.
- El robot no puede comparar corchos entre sí para "ordenarlos" por grosor, y tampoco puede hacerlo con las botellas.
- El robot tiene espacio disponible y brazos mecánicos para colocar botellas y corchos a su antojo, por ejemplo, en distintas posiciones de una mesa, si es necesario.

Diseñar el algoritmo que necesita el robot para taponar las N botellas de manera óptima.

EJERCICIO 5

En Acelandia el deporte nacional es el tenis. Existe un ranking, donde cada jugador tiene asignado un número de puntos en función de su calidad, es decir, el mejor jugador de ese país es el que tiene más puntos. Cada año se debe seleccionar una pareja entre todos los tenistas de Acelandia para jugar un torneo de dobles a nivel internacional.

El procedimiento de selección es un poco peculiar. Se coloca la puntuación de cada uno de los jugadores en una lista, de forma totalmente aleatoria, sin ningún tipo de ordenación. Una vez hecha la lista, cada jugador solo puede formar pareja con un jugador contiguo dentro de la lista, es decir, que esté delante o detrás de él en esa lista.

Obviamente, el primer jugador de la lista solo puede formar pareja con el segundo y el último con el penúltimo, pero el resto tiene dos posibles opciones para formar la pareja de dobles, correspondientes a los jugadores anterior y posterior de la lista. Con esta restricción, se elige la mejor pareja de dobles posible, que es aquella en la que la suma de los puntos de sus dos componentes sea mayor.

Diseñar un algoritmo cuya función principal siga el esquema de divide y vencerás, que decida qué pareja de dobles debe competir en Acelandia. Razonar cuál es la complejidad del algoritmo.

EJERCICIO 6

Tras su paso por la Sala de las Baldosas y conseguir la Cuna de la Vida, ahora Indiana Croft se enfrenta a un nuevo desafío antes de poder salir del Templo Maldito. Se encuentra en un puente bajo el que se observa una profunda oscuridad.

Afortunadamente, este lugar también aparece en el diario. El puente cruza el llamado Valle de Sombras, que empieza con una pendiente de bajada (la pendiente no es necesariamente constante) para después de llegar al punto más bajo empezar a subir hasta el otro extremo del puente (de nuevo, no necesariamente con pendiente contante). Justo en la parte inferior del valle hay un río, pero el diario no especifica su ubicación con respecto al puente (por ejemplo, no se sabe si el río está a 53 metros desde el comienzo del puente) ni la distancia en altura (es decir, no se sabe si el río está 228 metros por debajo, por ejemplo). En las pendientes hay afiladísimas rocas.

Si Indiana Croft tuviese tiempo, podría encontrar sin problema el punto por donde descolgarse del puente para llegar exactamente al río, ya que tiene un puntero laser para medir alturas que le dice cuántos metros hay desde el puente hasta el suelo en un punto determinado. El problema es que los sacerdotes del templo han averiguado que les han robado la Cuna de la Vida, están persiguiendo a Indiana Croft y le alcanzarán enseguida. El aventurero debe encontrar rápidamente la posición del río para descolgarse y huir seguro.

Diseñar el algoritmo que debería usar Indiana Croft para buscar el punto mínimo del valle en las condiciones indicadas. El algoritmo debe ser eficiente: al menos en el mejor caso debe tener un orden logarítmico. Se puede considerar el tiempo que tarda Indiana Croft en desplazarse a lo largo del puente es nulo y que la estimación del punto del río por donde descolgarse puede tener un error de aproximación de ε metros (ε es una constante dada).

EJERCICIO 7

En Abecelandia, ciudad famosa por sus N bellas plazas y que puede que conozcas, tienen un curioso sistema de carreteras: desde cada plaza sale una calle a todas las otras plazas que comienzan con una letra que se encuentre en su nombre (por ejemplo, de la plaza Aro salen calles que llevan a las plazas que empiezan por R, como las plazas Ruido y Reto, o por O, como la plaza Osa, pero no salen calles a plazas como Duende, Cascada o Tiara). Las calles son de sentido único (de la plaza Aro se puede ir a la plaza Ruido, pero no al revés ya que no cumple la regla de las letras; obviamente, otras plazas como Aro y Osa están conectadas entre sí en ambas direcciones). Todas estas conexiones entre las N plazas están recogidas en un callejero, representado por una matriz de adyacencia de tamaño NxN; así, el valor de Callejero[p, q] indica si se puede ir de la plaza p a la plaza q.

Se acerca el día 26 de Abril, festividad de San Isidoro de Sevilla (patrón de las Letras y, casualmente, de los informáticos), y en el Ayuntamiento de Abecelandia han decidido que para celebrarlo van a invertir la dirección de todas las calles que conectan sus plazas. En ese día no se podrá circular de Aro a Ruido, pero sí se permitirá ir de Ruido a Aro; obviamente, Aro y Osa seguirán conectadas entre sí.

Diseñar formalmente un algoritmo Divide y Vencerás que, teniendo como entrada el callejero de la ciudad (representado por la matriz de adyacencia), obtenga el nuevo callejero válido para el día de San Isidoro de Sevilla, indicando las estructuras de datos que se utilicen.

Entregables: un ejercicio a elegir entre los problemas 3 y 4 y un ejercicio a elegir entre los problemas 6 y 7.