Informes a mejorar:

informe 2 :MergeSort

informe 6: Select

informe 7: Corte de varillas

informe 8:OptimalBST

(cabe recalcar que el quick sort estaba subido desde la fecha dicha por favor revisar de nuevo)

Foros a mejorar:

Dado que no me dejaba comentar en los foros dejare aca cada uno de estos

**Foro 4:**

**Tiempo de ejecución,(eficiencia, demostración):**

La eficiencia del algoritmo depende de la posición en la que termine el pivote elegido. En el mejor caso, el pivote termina en el centro de la lista, dividiéndola en dos sublistas de igual tamaño. En este caso, el orden de complejidad del algoritmo es O(n•log n). En el peor caso, el pivote termina en un extremo de la lista. El orden de complejidad del algoritmo es entonces de O(n²). El peor caso dependerá de la implementación del algoritmo, aunque habitualmente ocurre en listas que se encuentran ordenadas, o casi ordenadas. Pero principalmente depende del pivote, si por ejemplo el algoritmo implementado toma como pivote siempre el primer elemento del array, y el array que le pasamos está ordenado, siempre va a generar a su izquierda un array vacío, lo que es ineficiente. En el caso promedio, el orden es O(n•log n). Y no es extraño, pues, que la mayoría de optimizaciones que se aplican al algoritmo se centren en la elección del pivote.

Suponiendo que el número total de elementos a ordenar es potencia de dos, es decir, n = 2k. De aquí podemos ver que k = log2(n), donde k es el número de divisiones que realizará el algoritmo.

En la primera fase del algoritmo habrán n comparaciones, en la segunda fase el algoritmo creará dos sublistas aproximadamente de tamaño n/2. El número total de comparaciones de estas dos sublistas es: 2(n/2) = n. En la tercera fase el algoritmo procesará 4 sublistas más, por tanto el número total de comparaciones en esta fase es 4(n/4) = n.

En conclusión, el número total de comparaciones que hace el algoritmo es:

n + n + n + ..... + n = kn, donde k = log2(n), por tanto el tiempo de ejecución del algoritmo en el mejor caso es O(n.log2n)

**Foro 7:**

**Tipos de select:**

**Ordenación parcial desordenada** :La [ordenación](https://www.geeksforgeeks.org/sorting-algorithms/) parcial desordenada es un algoritmo de [ordenación](https://www.geeksforgeeks.org/sorting-algorithms/) de tal manera que los primeros k elementos están ordenados y el resto de elementos están en orden aleatorio. Para encontrar el k-ésimo elemento más pequeño (o más grande). La complejidad del tiempo se reduce a O(k log k) . Pero como K ≤ n, la Complejidad de Tiempo asintótica converge a O(n)

**Selección basada en particiones**: para la selección basada en particiones, se utiliza el [algoritmo de selección rápida .](https://www.geeksforgeeks.org/quickselect-algorithm/) Es una variante del algoritmo [quicksort .](https://www.geeksforgeeks.org/quick-sort/) En ambos, elegimos un elemento pivote y, utilizando el paso de partición del algoritmo de clasificación [rápida](https://www.geeksforgeeks.org/quick-sort/) , organizamos todos los elementos más pequeños que el pivote a su izquierda y los elementos más grandes que él a su derecha.

Pero mientras [Quicksort](https://www.geeksforgeeks.org/quick-sort/) recurre en ambos lados de la partición, [Quickselect](https://www.geeksforgeeks.org/quickselect-algorithm/) solo recurre en un lado, el lado en el que está presente el k-ésimo elemento deseado.

Los algoritmos basados ​​en particiones se realizan en su lugar, lo que da como resultado una clasificación parcial de los datos. Se pueden hacer fuera de lugar al no cambiar los datos originales a costa de **O(n)**espacio auxiliar

**Mediana seleccionada como pivote:** Se puede utilizar un algoritmo de selección de mediana para realizar un algoritmo de selección o un algoritmo de ordenación, seleccionando la mediana de la matriz como el elemento pivote en el algoritmo [Quickselect](https://www.geeksforgeeks.org/quickselect-algorithm/) o [Quicksort](https://www.geeksforgeeks.org/quick-sort/) .

En la práctica, la sobrecarga del cálculo del pivote es significativa, por lo que estos algoritmos generalmente no se utilizan, pero esta técnica es de interés teórico para relacionar los algoritmos de selección y clasificación.

La mediana de una matriz es el mejor pivote para clasificar una matriz porque divide uniformemente los datos en dos partes y, por lo tanto, garantiza una clasificación óptima, suponiendo que el algoritmo de selección sea óptimo.

**Foro 10:**

**¿Qué tipo de aplicación se le puede dar al problema de la mochila?**

**Clave Asimétrica:**

La idea básica para usar una mochila simple en un sistema de [criptografía asimétrica](https://www.ecured.cu/Criptograf%C3%ADa_asim%C3%A9trica) es conseguir una transformación secreta que transforme la mochila simple en una mochila general cuya resolución tenga un coste computacional alto. A esta mochila la llamaremos mochila tramposa. La clave pública será la mochila tramposa y con ella el emisor cifrará el mensaje de la misma forma que [se hacía antes](https://www.ecured.cu/Problema_de_la_mochila#Clave_sim.C3.A9trica). La clave privada estará formada por los parámetros que permiten convertir el mensaje cifrado con la mochila tramposa en un mensaje cifrado con la mochila simple. Una vez obtenido esto el receptor puede descifrar el mensaje fácilmente usando la mochila simple.

En resumen, el esquema se basa en cifrar con una [función unidireccional](https://www.ecured.cu/index.php?title=Funci%C3%B3n_unidireccional&action=edit&redlink=1) basada en un [problema NP-completo](https://www.ecured.cu/index.php?title=NP-completo&action=edit&redlink=1) (el problema de la mochila) que tienen una puerta trampa que aprovecha el receptor para descifrar el mensaje. Si no se dispone de esa puerta trampa, el proceso de descifrado, al ser un [problema NP-completo](https://www.ecured.cu/index.php?title=NP-completo&action=edit&redlink=1), teóricamente tendría un coste computacional muy alto.

En esta idea se basa por ejemplo El [criptosistema de Merkle-Hellman](https://www.ecured.cu/index.php?title=Criptosistema_de_Merkle-Hellman&action=edit&redlink=1). Este algoritmo para hacer la transformación halla:Expresion matematica.png

* Un valor tal que son los elementos de la mochila simple.

Expresion matematica algoritmo.png.png

* Un valor w entero tal que en el grupo de los enteros de módulo u

La mochila tramposa se halla usando la expresión Mochila tramposa.png. Hay que verificar que la mochila tramposa así obtenida no sea una mochila fácil de resolver (no siempre es así).Criptograma.png

Para descifrar el criptograma hay que aplicar para cada

una de las sumas C obtenidas al cifrar. A continuación cada valor suma transformado hay que descifrarlo [de la forma habitual](https://www.ecured.cu/Problema_de_la_mochila#Clave_sim.C3.A9trica) con la mochila simple original, lo cual es trivial.

**Clave Simétrica:**

Un problema de la mochila simple puede usarse como clave secreta de una algoritmo de [cifrado simétrico](https://www.ecured.cu/Criptograf%C3%ADa_sim%C3%A9trica). Para ello lo que hay que hacer es representar la información que se quiera cifrar en binario y se pasa cada bit por la mochila por la secuencia de números del conjunto S. Si un bit es 1 entonces se incluye en la suma el elemento que le corresponde. Si es un 0 entonces no se incluye.

Por ejemplo sea S = {2, 4, 10, 19, 40}. Por tanto m = 5. Supongamos que quiero cifrar el mensaje M = ADIOS. Pasando el mensaje a ASCII/ANSI( A = 01000001, D = 01000100, I = 01001001, O = 01001111, S = 01010011) tenemos el mensaje (agrupo de 5 en 5 ya que m=5)

M = 01000 00101 00010 00100 10010 10011 11010 10011

Haciendo las sumas de cada quinteto obtengo

C = (4), (10+40), (19), (10), (2+19), (2+19+40), (2+4+19), (2+19+40)= 4, 50, 19, 10, 21, 61, 25, 61

Que será el mensaje cifrado.

Para descifrar es receptor, que conoce S, recibe el mensaje C y opera de forma contraria resolviendo el problema de la mochila simple para cada uno de los valores de C.

* Para obtener como suma un 4 la solución es 01000
* 50 -> 00101
* 19 -> 00010
* 10 -> 00100
* 21 -> 10010
* 61 -> 10011
* 25 -> 11010
* 61 -> 10011

Si uno todos los bits y agruparlos en grupos de 8 bits (ANSI/ASCII) obtengo el mensaje original.

Esta forma de cifrar no es segura ya que es evidente que teniendo un suficientes pares, de mensaje original y mensaje cifrado asociado, será muy fácil obtener la clave S con la que se está cifrando. Sin embargo esta forma de cifrar es muy rápida y puede será aprovechada en las aplicaciones de cifrado con [clave asimétrica](https://www.ecured.cu/index.php?title=Problema_de_la_mochila_simple&action=edit&redlink=1)**.**