

Tarea 2
Análisis de Algoritmos
Profesores: Jorge Urrutia y Adriana Ramírez

Fecha de entrega: septiembre 09 de 2024.

No se reciben tareas después de esta fecha.

Se entregarán en el classroom del curso en formato pdf legible.

Todas sus respuestas deben estar plenamente justificadas.

1. Cada año, cuando los estudiantes llegan a Hogwarts, el sombrero seleccionador clasifica a los nuevos estudiantes en una de las cuatro casas Gryffindor, Hufflepuff, Ravenclaw o Slytherin. Un profesor pone el sombrero en la cabeza del estudiante y después de tiempo constante el sombrero le dice a qué casa se unirá. Este año, los gemelos Weasley llenaron casi todo Hogwarts con una pegajosa sustancia marrón, momentos antes de la selección anual. Como resultado, la selección tuvo que llevarse a cabo en los pasillos del sótano, donde había tan poco espacio para moverse que los estudiantes tenían que hacer una larga fila. Después de que todos supieran en qué casa estaban, los estudiantes trataron de agruparse por casa, pero había muy poco espacio en el pasillo para que más de un estudiante se moviera a la vez. Afortunadamente, el sombrero seleccionador tomó algoritmos hace muchos años, por lo que sabía cómo agrupar a los estudiantes en tiempo lineal a pesar del poco espacio ¿Qué algoritmo usó el sombrero seleccionador?.
2. Queremos ordenar una lista S de n enteros que contiene muchos elementos duplicados. Supongamos que los elementos de S sólo tienen $O(\log n)$ valores distintos.
 - a) Encuentre un algoritmo que toma a lo más $O(n \log \log n)$ tiempo para ordenar S .
 - b) ¿Por qué no viola la cota inferior de $O(n \log n)$ para el problema de ordenación.
3. Suponga que tenemos dos arreglos ordenados $A[1 \dots n]$ y $B[1 \dots n]$ y un entero k . Describe un algoritmo para encontrar el k -ésimo elemento en la unión de A y B . Por ejemplo, si $k = 1$, tu algoritmo debe regresar al elemento más pequeño de $A \cup B$; si $k = n$, tu algoritmo debe regresar la mediana de $A \cup B$. Puedes suponer que los arreglos no contienen duplicados. Tu algoritmo debe tener complejidad de tiempo $\Theta(\log n)$. Hint: Primero resuelve el caso especial $k = n$.
4. Sea A un arreglo de n números enteros distintos. Suponga que A tiene la siguiente propiedad: existe un índice $1 \leq k \leq n$ tal que $A[1], \dots, A[k]$ es una secuencia incremental y $A[k+1], \dots, A[n]$ es una secuencia decremental.
 - a) Diseña y analiza un algoritmo eficiente para encontrar k .
 - b) Si no conoces el valor de n , cómo resuelves el problema.
5. Rotar un arreglo significa hacer un corrimiento a la derecha de todos los elementos, excepto el último que pasará a la primera posición del arreglo. Por ejemplo $A = [1, 3, 4, 5, 7, 10, 14, 15, 16, 19, 20, 25]$ y después de una rotación $A = [25, 1, 3, 4, 5, 7, 10, 14, 15, 16, 19, 20]$. Dado un arreglo ordenado A de n números enteros distintos que ha sido rotado un número desconocido de veces. Diseña un algoritmo de $O(\log n)$ tiempo que encuentre el k -ésimo elemento del arreglo. Por ejemplo, $A = [15, 16, 19, 20, 25, 1, 3, 4, 5, 7, 10, 14]$ y $k = 5$, la respuesta es 7.

6. You are a young scientist who just got a new job in a large team of 100 people (you the 101-st). A friend of yours who you believe told you that you have more honest colleagues than liars, and that that's all what he can tell you, where a liar is a person who can either lie or tell the truth, while an honest person is one who always tells the truth. Of course, you'd like to know exactly your honest colleagues and the liars, so that you decide to start an investigation, consisting of a series of questions you are going to ask your colleagues. Since you don't wish to look suspicious, you decide to ask only questions of the form "Is Mary an honest person?" and of course, to ask as few questions as possible. Can you sort out all your honest colleagues? What's the minimum number of questions you'd ask in the worst case? You can assume that your colleagues know each other well enough to say if another person is a liar or not. (Hint: Group people in pairs (X, Y) and ask X the question "Is Y honest?" and Y the question "Is X honest?". Analyze all the four possible answers. Once you find an honest person, you can easily find all the others. Challenge: can you solve this enigma asking less than 280 questions in total?)

Generalize the strategy above and show that given n people such that less than half are liars, you can sort them out in honest persons and liars by asking $\theta(n)$ questions.

7. Sea $A[1, \dots, n]$ un arreglo de números reales. Diseña un algoritmo para realizar cualquier secuencia de las siguientes operaciones:

- $Add(i, y)$, suma el valor y al i -ésimo número.
- $Partial - sum(i)$, regresa la suma de los primeros i números, es decir $Partial - sum(i) = A[1] + \dots + A[i]$

No hay inserciones ni eliminaciones; el único cambio es a los valores de los números. Cada operación debe tomar $O(\log n)$ pasos. Puedes utilizar un arreglo adicional de tamaño n como espacio de trabajo.

8. **Permutaciones de Josephus:** Supongamos que n personas están sentadas alrededor de una mesa circular con n sillas, y que tenemos un entero positivo $m \leq n$. Comenzando con la persona con etiqueta 1, (moviendonos siempre en la dirección de las manecillas del reloj) comenzamos a remover los ocupantes de las sillas como sigue: Primero eliminamos la persona con etiqueta m . Recursivamente, eliminamos al m -ésimo elemento de los elementos restantes. Este proceso continua hasta que las n personas han sido eliminadas. El orden en que las personas han sido eliminadas, se le conoce como la (n, m) -permutación de Josephus. Por ejemplo si $n = 7$ y $m = 3$, la $(7, 3)$ -permutación de Josephus es: $\{3, 6, 2, 7, 5, 1, 4\}$.

- a) Supongamos que m es constante. De un algoritmo lineal para generar la (n, m) -permutación de Josephus.
- b) Supongamos que m no es constante. Describa un algoritmo con complejidad $O(n \log n)$ para encontrar la (n, m) -permutación de Josephus. Hint: construya un árbol de búsqueda de rangos.