

Tarea Examen 1
Análisis de Algoritmos

Fecha de entrega: Octubre 09 de 2020.
No se reciben tareas después de esta fecha.

Todas sus respuestas deben estar plenamente justificadas.

1. ¿Cuántas comparaciones son necesarias y suficientes para ordenar cualquier lista de cinco elementos? Justifique su respuesta.
2. Dados dos arreglos ordenados A y B de longitud n y m respectivamente. Diseña un algoritmo de tiempo $O(n+m)$ que obtenga un arreglo C que contenga los elementos en común entre A y B , C no debe tener elementos repetidos.
3. Consider the following sorting algorithm:

```
STUPIDSORT( $A[0..n-1]$ ):  
  if  $n = 2$  and  $A[0] > A[1]$   
    swap  $A[0] \leftrightarrow A[1]$   
  else if  $n > 2$   
     $m = \lceil 2n/3 \rceil$   
    STUPIDSORT( $A[0..m-1]$ )  
    STUPIDSORT( $A[n-m..n-1]$ )  
    STUPIDSORT( $A[0..m-1]$ )
```

- a) Prove that STUPIDSORT actually sorts its input.
 - b) Would the algorithm still sort correctly if we replaced $m = \lceil 2n/3 \rceil$ with $m = \lfloor 2n/3 \rfloor$? Justify your answer.
 - c) Show that the number of swaps executed by STUPIDSORT is at most $\binom{n}{2}$.
4. Supongamos que tenemos que ordenar una lista L de n enteros cuyos valores están entre 1 y m . Pruebe que si m es $O(n)$ entonces los elementos de L pueden ser ordenados en tiempo lineal. ¿Qué pasa si m es de $O(n^2)$? ¿Se puede realizar en tiempo lineal? ¿Por qué?
5. Describe an algorithm that, given n integers in the range 0 to k , preprocesses its input and then answers any query about how many of the n integers fall into a range $[a \dots b]$ in $O(1)$ time. Your algorithm should use $\Theta(n+k)$ preprocessing time.

6. Sea A un arreglo de n elementos, tal que cada elemento se encuentra a lo más a k posiciones de su posición ordenada. Diseñe un algoritmo que ordene A en $O(n \log k)$.
7. An abs-sorted array is an array of numbers in which $|A[i]| \leq |A[j]|$ whenever $i < j$. For example, the array $A = [-49, 75, 103, -147, 164, -197, -238, 314, 348, -422]$, though not sorted in the standard sense, is abs-sorted. Design an algorithm that takes an abs-sorted array A and a number k , and returns a pair of indices of elements in A that sum up to k . For example if $k = 167$ your algorithm should output $(3, 7)$. Output $(-1, -1)$ if there is no such pair.
8. **The Hogwarts Sorting Hat**
 Every year, upon their arrival at Hogwarts School of Witchcraft and Wizardry, new students are sorted into one of four houses (Gryffindor, Hufflepuff, Ravenclaw, or Slytherin) by the Hogwarts Sorting Hat. The student puts the Hat on their head, and the Hat tells the student which house they will join. This year, a failed experiment by Fred and George Weasley filled almost all of Hogwarts with sticky brown goo, mere moments before the annual Sorting. As a result, the Sorting had to take place in the basement hallways, where there was so little room to move that the students had to stand in a long line.
 After everyone learned what house they were in, the students tried to group together by house, but there was too little room in the hallway for more than one student to move at a time. Fortunately, the Sorting Hat took Algorithms many years ago, so it knew how to group the students as quickly as possible. What method did the Sorting Hat use? More formally, you are given an array of n items, where each item has one of four possible values, possibly with a pointer to some additional data. Design and analyze an algorithm that rearranges the items into four clusters in $O(n)$ time using only $O(1)$ extra space.
9. Pruebe que el segundo elemento más chico de una lista de n elementos distintos puede encontrarse con $n + \lceil \log n \rceil - 2$ comparaciones.