Tarea 3

Universidad Nacional Autónoma de México
Facultad de Ciencias
Programación Declarativa

Fong Baeza Luis Fernando Yang
`fernandofong@ciencias.unam.mx`

Lezama Hernández María Ximena
`lezama@ciencias.unam.mx`

26 de Marzo 2018

1. Sea A = {a$_1$ a$_2$ ..., a$_n$} números naturales tales que $1 \leq$ a$_i$ $\leq n^2$. Muestra un algoritmo que ordene A en tiempo O(n).
   Solución:
   Tenemos n números, entonces hay que recorrer un cierto número de veces esa lista para que podamos ordenarla en el tiempo requerido, pero sabiendo que no podemos excedernos o podríamos caer en O(n^2). Entonces, si analizamos los números por dígitos, tenemos que para cada número, a lo más tiene 3 dígitos puesto que la codificación como combinación lineal de la base 10, como no puede exceder de n^2, entonces tenemos que sus dígitos son a lo más tres, entonces lo que vamos a hacer, va a ser recorrer todos los números, que son n, pero los vamos a ordenar con respecto al dígito menos significativo, es decir, si tenemos {99, 87, 56, 71, 20, 25, 121, 5, 23, 11} en la primera iteración debería de tener: {20, 11, 121, 71, 23, 5, 25, 56, 87, 99}, en la segunda vuelta, deberíamos de eliminar, todos los que ya no tienen dígito siguiente, esos primeros, así conforme nos los encontremos y con el resto, repetir recursivamente pero con el dígito siguiente, es decir {5, 11, 20, 121, 23, 25, 56, 71, 87, 99}, entonces, ponemos al frente a los que ya no tengan dígito siguiente y repetimos recursivamente con los que sí dejando 5, 11, 20, 23, 25, 56, 71, 87, 99, 121. La ordenamos en tiempo lineal puesto que solo recorrimos todos los elementos a lo más 3 veces, que esto toma θ(3n) que para fines prácticos es O(n). □

2. Dados A y B conjuntos de enteros, $\mid A \mid= n =\mid B \mid$, y un $x \in Z$, diseña un algoritmo que en tiempo $O(nlog(n))$ encuentre dos números $a \in A$, $b \in B$ tales que $a + b = x$.
   Solución:
   Tenemos dos conjuntos de n números enteros, así que para facilitar las cosas, vamos primero que ordenar ambos conjuntos de enteros A y B y eso nos toma $O(nlog(n))$. Con los dos conjuntos ordenamos ahora tenemos que encontrar la x entonces, buscamos desde el menor del conjunto A y fijamos esa cantidad, y se la restamos a la x que tenemos ahora, y ese nuevo numero lo buscamos por búsqueda binaria en el conjunto B, en caso de no estar, continuamos con el siguiente del conjunto A, y decimos que no se encuentra si $A[i]$ es mayor a x, y si $A[i]$ es el final del arreglo A. □

3. Sea $[a_1, a_2, ..., a_n]$ un arreglo de enteros y $x \in Z$. Muestra un algoritmo que en tiempo $O(nlog(n))$ encuentre la pareja (a_i, a_j)tal que $a_i + a_j \leq x$y que maximice $a_i + a_j$.
   Solución:
   Parecido al anterior, ordenamos ambos conjuntos de enteros A y B y eso nos toma $O(nlog(n))$. Primero checamos si la suma de los menores es menor a x, si no, no existe una pareja que cumpla con que $a_i + a_j \leq x$, de otra manera, recorremos ambos arreglos ordenados, y vamos moviendo y mientras la suma sea menor a x, iteramos alternadamente entre A y B, si uno de

los dos resulta ser mayor a x regresamos el índice previo, y continuamos con el otro arreglo hasta que este también resulte ser mayor a x y regresemos el indice anterior. Recordemos también que terminaremos de iterar si la suma resulta ser igual a x $\square$

4. Sea A un arreglo con n elementos, cada elemento en el arreglo es verde, rojo o azul; además, se cuenta con las siguientes operaciones:

   - $color(A, i)$: devuelve el color del elemento $A[i]$.
   - $swap(A, i, j)$: intercambia los elementos $A[i]$ y $A[j]$.

   Muestra un algoritmo que en tiempo lineal ordena A así: verdes, rojos y luego azules. _Solución:_ Recorremos el arreglo y por cada que encontremos primero un verde, hacemos un swap con su anterior hasta que este su antecesor sea un verde o no exista (sea la posición 0), cuando acabemos con el verde, repetimos la acción con los rojos, hacemos un swap con el anterior al menos que sea verde, rojo, o no exista (sea la posición 0), y finalmente con el azul, y dejamos de hacer el swap con el anterior al menos que sea azul, rojo, o no exista (sea la posición 0). Debemos de darnos cuenta que mediante ordenamos, asumimos que si hay un color igual al cual estar ordenando detrás de ti, significa que a partir de ese color hacia atrás ya están ordenados. $\square$

5. Muestra que es posible multiplicar dos polinomios lineales $ax + b$ y $cx + d$ sin usar más de tres multiplicaciones.

   Solución:
   Primero realizamos la primera multiplicación.

   $$(ax + b)(cx + d) = acx + adx + bcx + bd$$

   Si factorizamos la x

   $$acx + adx + bcx + bd = acx + (ad + bc)x + bd$$

   Y sumamos un cero

   $$acx + (ad + bc)x + bd = acx + (ad + bc + ac + bd - ac - bd)x + bd$$

   $$acx + (ad + bc + ac + bc - ac - bc)x + bd = acx + ((ac + b)(c + d) - ac - bd)x + bd$$

   Como podemos ver tenemos ahí solo tres multiplicaciones:

   1. ac
   2. (ac + b)(c + d)
   3. bd

   Por lo que podemos multiplicar dos polinomios lineales sin usar más de tres multiplicaciones. $\square$