Análisis de Algoritmos e introducción a Matemáticas Discretas
Tarea 2

Rubén Pérez Palacios Lic. Computación Matemática
Profesor: Dr. Carlos Segura González

29 de agosto de 2021

La solución propuesta al problema es con un algoritmo Divide and Conquere. Para un punto p usaremos la notación $p.x$ y $p.y$, para referirnos a la coordenada x y y del punto p respectivamente. El algoritmo dado un conjunto P de puntos crea una partición de dos subconjuntos PL y PR tal que para todos los puntos $p \in PL, q \in PR$ se cumple que $p < q$ (con orden lexicográfico primero por x y luego por y) y $|PL - PR|$ es a lo más uno; cuenta los puntso dominados de un conjunto por los puntos del otro conjunto, y recursivamente hace lo mismo para PL y PR. Si un punto $p \in P$ domina otro punto $q \in P$ entonces siempre se cuenta esto puesto que se hace en ese momento en el alguna recursión del algoritmo.

Ahora veremos como contar los puntos dominados entre puntos de PL y PR. Notemos que por construcción de PL y PR ningún punto de PR dominara a alguno de PL, entonces solo hace falta contar para cada punto de PL cuantos puntos domina de los puntos de PR. Por construcción todo los puntos de PL tienen una coordenada x menor o igual a los de PR, por lo que para contar para un punto $p \in PL$ cuantos puntos de PR domina solo hace falta ver cuantos de ellos tienen una coordenada y mayor o igual. Para ello supongamos que tenemos ordenados los puntos tanto de PL como de PR por y de menor a mayor, entonces tomamos los primeros puntos $p \in PL$ y $q \in PR$, si $p.y \leq q.y$ entonces p dominara a todos los puntos en PR por lo que acumulamos la cantidaad $|PR|$ a los puntos que p domina, luego descartamos al punto p de PL en caso de qque $p.y > q.y$ entonces descartamos a $q.y$, y repetimos lo mismo para los nuevos primeros puntso de PL y PR. Si un punto $p \in PL$ domina otro punto $q \in PR$ entonces $p.y \leq q.y$ y por construcción lo habrá contado el algoritmo. Este proceso tiene una complejidad de $\theta(M)$, donde $M = |PL|$ si los datos estan ordenados (esto se consigue haciendo uso de un merge sort mientras dividimos los conjuntos), en caso contrario sería de $\theta(M \log(M))$.

Por lo tanto al final el agoritmos habrá contado cuantos puntos dominada cada punto. Ahora analizaremos la complejidad de este con un conjunto de entrada de tamaño N, para ello nos apoyaremos del arbol de recursión de este. Notemos que para cada nodo con M datos este se divide en dos nodos que contienen la $\frac{M}{2}$ datos (difieren en a lo más uno pero este se ignora ya que en cada nodo la complejidad es de $\Theta(M)$), por lo que para todo nodo en el nivel i tendrán $\frac{N}{2^i}$ de datos por lo que ell trabajo de cada nodo es de $\frac{N}{2^i}$, luego la cantidad de nodos en el nivel i es de 2^i, por lo tanto concluimos

$$T(N) = \sum_{i=0}^{log_2(N)} 2^i \frac{N}{2^i} = N \log_2(N) = \theta(N \log(N)).$$