A Comparison of Time Complexity of Greedy, Dynamic Programming and Divide & Conquer

CS F364: Design & Analysis of Algorithms
Lec 16_2, BITS Pilani

Suppose for a given problem we have three algorithms by using the above algorithm design techniques: a greedy algorithm, a dynamic programming algorithm and a divide & conquer algorithm. Divide and conquer algorithm will be least efficient because it is solving each sub-problem even if it is an overlapping sub-problem. Dynamic Programming algorithm will be more efficient than divide and conquer algorithm because it is avoiding the repeated solutions of sub-problems. If we compare the greedy algorithm with dynamic programming algorithm, the greedy algorithm will be more efficient than Dynamic Programming algorithm because greedy algorithm we do not solve any sub-problem for making a choice. In general, the time complexity will be:

Greedy < Dynamic Programming < Divide & Conquer

Example: Consider the single source shortest path problem. Dijkstra's algorithm is greedy and has complexity $O(|E|log|V|)$. Bellman-Ford algorithm is dynamic programming algorithm and has complexity $O(|V||E|)$. Under the assumption of $|E| = O(|V|^2)$, we can design a divide and conquer algorithm by using the recurrence of bellman-ford algorithm of complexity $\Omega|V|^{|V|}$. The Comparison of three algorithm using $|E| = O(|V|^2)$,

Greedy: $O(|V|^2 log|V|)$ < Dynamic Prog.: $O(|V|^3)$ < D&C: $\Omega(|V|^{|V|})$.