A Comparison of Time Complexities of Greedy, Dynamic Programming, and Divide and Conquer:

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 and Conquer Algorithm. Divide and Conquer algorithm will be least efficient because it is solving each subproblem even if it is an overlapping subproblem. Dynamic Programming algorithm will be more efficient than Divide and Conquer algorithm because it is avoiding the repeated solutions of subproblems. If we compare the Greedy algorithm with the Dynamic Programming algorithm, the Greedy algorithm will be more efficient than the Dynamic Programming algorithm because in Greedy algorithm we do not solve any subproblem for making a choice. In general, the time complexities will be:

Greedy < Dynamic Programming < Divide & Conquer

Example: Consider the Single Source Shortest Paths 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 algorithms (using $|E| = O(|V|^2)$):

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