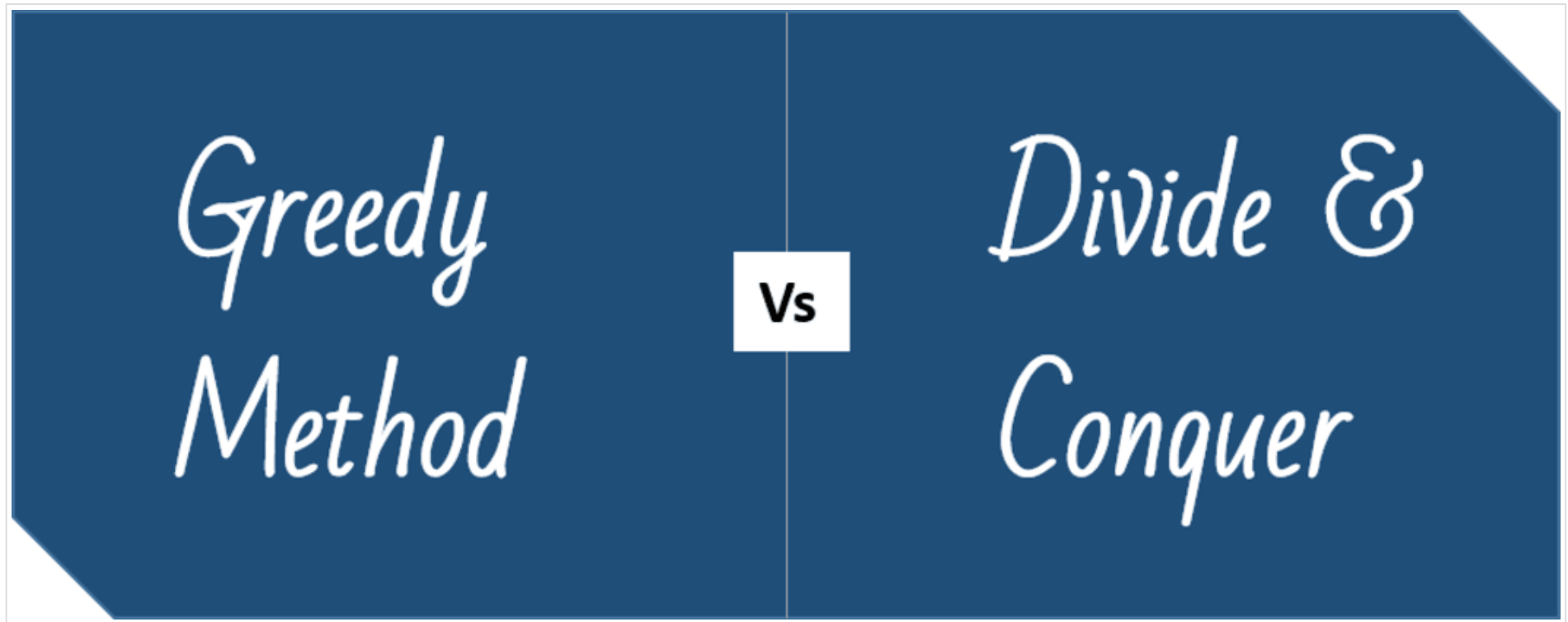# Greedy Vs. Divide and Conquer

In this article, we will discuss greedy vs divide and conquer approach. Greedy algorithms are typically used to solve optimization problems. When problem has multiple solutions, we can use various strategies to find best out of all. Greedy is one of the optimization method. Divide and conquer is general problem solving method, which divides the problem into smaller sub problems, solves the smaller sub problems and solutions of smaller sub problems are combined to generate the solution of original larger problem.

Both the methods are compared in following table.

Divide and conquer	Greedy Algorithm
Divide and conquer is used to find the solution, it does not aim for the optimal solution.	A greedy algorithm is optimization technique. It tries to find an optimal

Divide and conquer	Greedy Algorithm
	solution from the set of feasible solutions.
DC approach divides the problem into small sub problems, each sub problem is solved independently and solutions of the smaller problems are combined to find the solution of the large problem.	In greedy approach, the optimal solution is obtained from a set of feasible solutions.
Sub problems are independent, so DC might solve same sub problem multiple time.	Greedy algorithm does not consider the previously solved instance again, thus it avoids the re-computation.
DC approach is recursive in nature, so it is slower and inefficient.	Greedy algorithms are iterative in nature and hence faster.
Divide and conquer algorithms mostly runs in polynomial time	Greedy algorithms also run in polynomial time but takes less time than Divide and conquer
Examples: Merge sort, Quick sort, Binary search, Strassen's matrix multiplication, convex hull problem, large integer problem, exponential problem	Examples: Knapsack problem, Activity selection problem, Job scheduling problem, Huffman codes, Optimal storage on tapes, Optimal merge pattern.

This table illustrates the difference of Greedy Vs Divide and Conquer approaches.