|  |  |  |
| --- | --- | --- |
| Dynamic Programming | Divide and conquer | Greedy method |
| 1.Dynamic programming solves optimization problem by combining solution to sub solution | 1. Partition the problem into independent sub-problems | 1. A greedy always makes the choice that looks best at the moment. |
| 2. dynamic programming is applicable when sub-problems are not independent. | 2. Solve the sub-problem recursively. | 2. The hope is a local optimal solution. |
| 3. Solve every sub-problem only once and store the answer in the table form use when it reappear. | 3. combine solution of sub-problems | 3. In greedy algorithm approach, decisions are made from the given solution domain. As being greedy, the closest solution that seems to provide an optimum solution is chosen. |
| 4. The key is to store the solutions of sub-problems to be reused in the future. | 4. A divide and conquer approach will do more work than necessary. | 4. An algorithm is designed to achieve optimum solution for a given problem |
| Example:  Matrix chain Multiplication. | Example:  Merge sort | Example:  Playing cards. |

**Problem Sets**

1. Describe the differences between dynamic programming, divide and conquer and greedy methods

3. Analyse the complexity of LCS

In LCS we should check subsequence of x[1…..m] to see if it is also a subsequence of y[1…n]. As there are 2^m subsequence possible for x, the complexity of this solution would be O(n/2^m).