Unit 4

| |
|---|
| 1. With an example derive the time complexity of matrix chain multiplication? |
| 2. What is dynamic programming explain its characteristics and components? |
| 3. Briefly discuss about Chinese remainder theorem with an example? |
| 4. Show that the matrix chain multiplication algorithm provides the optimal solution to multiply a group of matrices? |
| 5. Develop an algorithm for Floyd-Warshall method in dynamic programming |
| 6. Write the algorithm for Chinese remainder theorem and specify its time complexity? |
| 7. With an example perform all the arithmetic operations using modulo representation? |

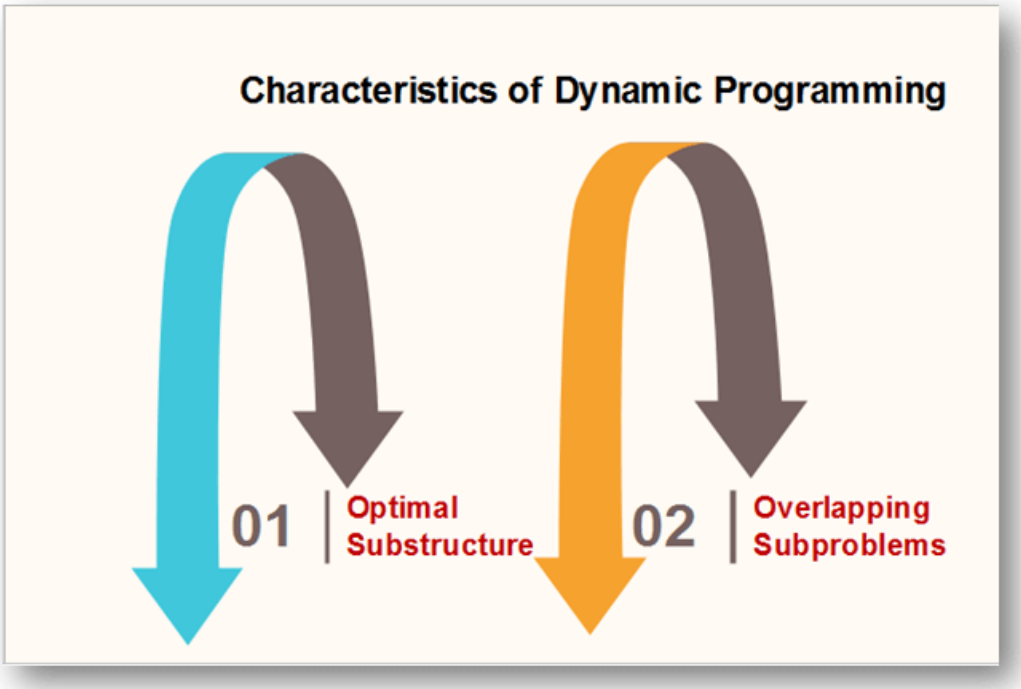
1. https://www.youtube.com/watch?v=prx1psByp7U

2. Dynamic Programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler subproblems and utilizing the fact that the optimal solution to the overall problem depends upon the optimal solution to its subproblems.

**Components of Dynamic programming**

- **Stages**
  The given problem can be divided into a number of subproblems which are called stages. A stage is a small portion of given problem.
- **States**
  This indicates the subproblem for which the decision has to be taken. The variables which are used for taking a decision at every stage that is called as a state variable.
- **Decision**
  At every stage, there can be multiple decisions out of which one of the best decisions should be taken. The decision taken at each stage should be optimal; this is called as a stage decision.
- **Optimal policy**
  It is a rule which determines the decision at each and every stage; a policy is called an optimal policy if it is globally optimal. This is called as Bellman principle of optimality.

**Characteristics of Dynamic Programming**:
Dynamic Programming works when a problem has the following features:-




- **Optimal Substructure:** If an optimal solution contains optimal sub solutions then a problem exhibits optimal substructure.
- **Overlapping subproblems:** When a recursive algorithm would visit the same subproblems repeatedly, then a problem has overlapping subproblems.

If a problem has optimal substructure, then we can recursively define an optimal solution. If a problem has overlapping subproblems, then we can improve on a recursive implementation by computing each subproblem only once.

If a problem doesn't have optimal substructure, there is no basis for defining a recursive algorithm to find the optimal solutions. If a problem doesn't have overlapping sub problems, we don't have anything to gain by using dynamic programming.

If the space of subproblems is enough (i.e. polynomial in the size of the input), dynamic programming can be much more efficient than recursion.

3. Chinese Remainder Theorem: If m1, m2, .., mk are pairwise relatively prime positive integers, and if a1, a2, .., ak are any integers, then the simultaneous congruences $x \equiv a1 \pmod{m1}$, $x \equiv a2 \pmod{m2}$, ..., $x \equiv ak \pmod{mk}$ have a solution, and the solution is unique modulo m, where $m = m1m2 \cdots mk$ .

[ http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/chinese_remainder.pdf

just for reference, no need to by heart]

https://www.youtube.com/watch?v=rnXhH5dvnCo

Best example u will ever find on utube

4. https://www.youtube.com/watch?v=prx1psByp7U

5. https://www.youtube.com/watch?v=oNI0rf2P9gE

6. https://www.youtube.com/watch?v=rnXhH5dvnCo

http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/chinese_remainder.pdf

**Algorithm 1** Chinese remainder theorem solution

**Input:** $y_1, y_2, \cdots, y_n$ are integers in congruent equations and $cp_1, cp_2, \cdots, cp_n$ are relatively co-prime integers.
$x \equiv y_i \ mod \ cp_i$, for $i = 1, 2, \cdots, n$
**Output:** Solution of congruent equations ( i.e. value of $x$).

1. Multiplication of relatively co-prime numbers.
   $P = cp_1 \times cp_2 \times \cdots \times cp_n$
2. Calculation of constant $P_i$.
   $P_i = \frac{P}{cp_i}$, $gcd(P_i, cp_i) = 1$, for $i = 1, 2, \cdots, n$
3. Calculate $P_i^{-1}$ multiplicative inverse of $P_i$ in set $Z_{cp_i}$.
   $P_i \times P_i^{-1} \equiv 1 \ mod \ cp_i$, for $i = 1, 2, \cdots, n$
4. The solution of CRT which is unique value of $x$.
   $x = (y_1 \times P_1 \times P_1^{-1} + y_2 \times P_2 \times P_2^{-1} + \cdots + y_n \times P_n \times P_n^{-1}) \ mod \ P$

7. Modular Arithmetic

https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/modular-multiplication

https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/modular-addition-and-subtraction