DESIGN AND ANALYSIS OF ALGORITHMS

SESSION -19

Dynamic programming

Well known algorithm design techniques:.
– Divide-and-conquer algorithms

• Another strategy for designing algorithms is dynamic programming.

– Used when problem breaks down into recurring small Subproblems

• Dynamic programming is typically applied to optimization problems. In such problem there can be many solutions. Each solution has a value, and we wish to find a solution with the optimal value

Dynamic programming

Dynamic Programming is a general algorithm design technique for solving problems defined by or formulated as
recurrences with overlapping sub instances.

Invented by American mathematician **Richard Bellman** in the **1950s to solve optimization problems** .

Main idea:

- set up a recurrence relating a solution to a larger instance to solutions of some smaller instances
- solve smaller instances once
- record solutions in a table
- extract solution to the initial instance from that table

Dynamic programming

Dynamic programming is a way of improving on inefficient *divide and-conquer algorithms.*

• By "*inefficient", we mean that* ***the same recursive call is made over and over.***

• If **same *subproblem is solved several times, we can use table to*** store result of a subproblem the first time it is computed and thus never have to recompute it again.

• Dynamic programming is applicable when the subproblems are dependent, that is, when subproblems share sub subproblems

.• **"Programming" refers to a tabular method**

Elements of Dynamic Programming

DP is used to solve problems with the following characteristics:

- Simple subproblems
  - We should be able to break the original problem to **smaller subproblems that have the same structure**

- Optimal substructure of the problems
  - The **optimal solution to the problem contains within optimal solutions to its subproblems.**

- Overlapping sub-problems
  - there exist some places where we solve the **same subproblem more** than **once.**

Steps to Designing a Dynamic Programming Algorithm

1. Characterize optimal substructure
2. Recursively define the value of an optimal Solution
3. Compute the value bottom up
4. (if needed) Construct an optimal solution.

Principle of Optimality

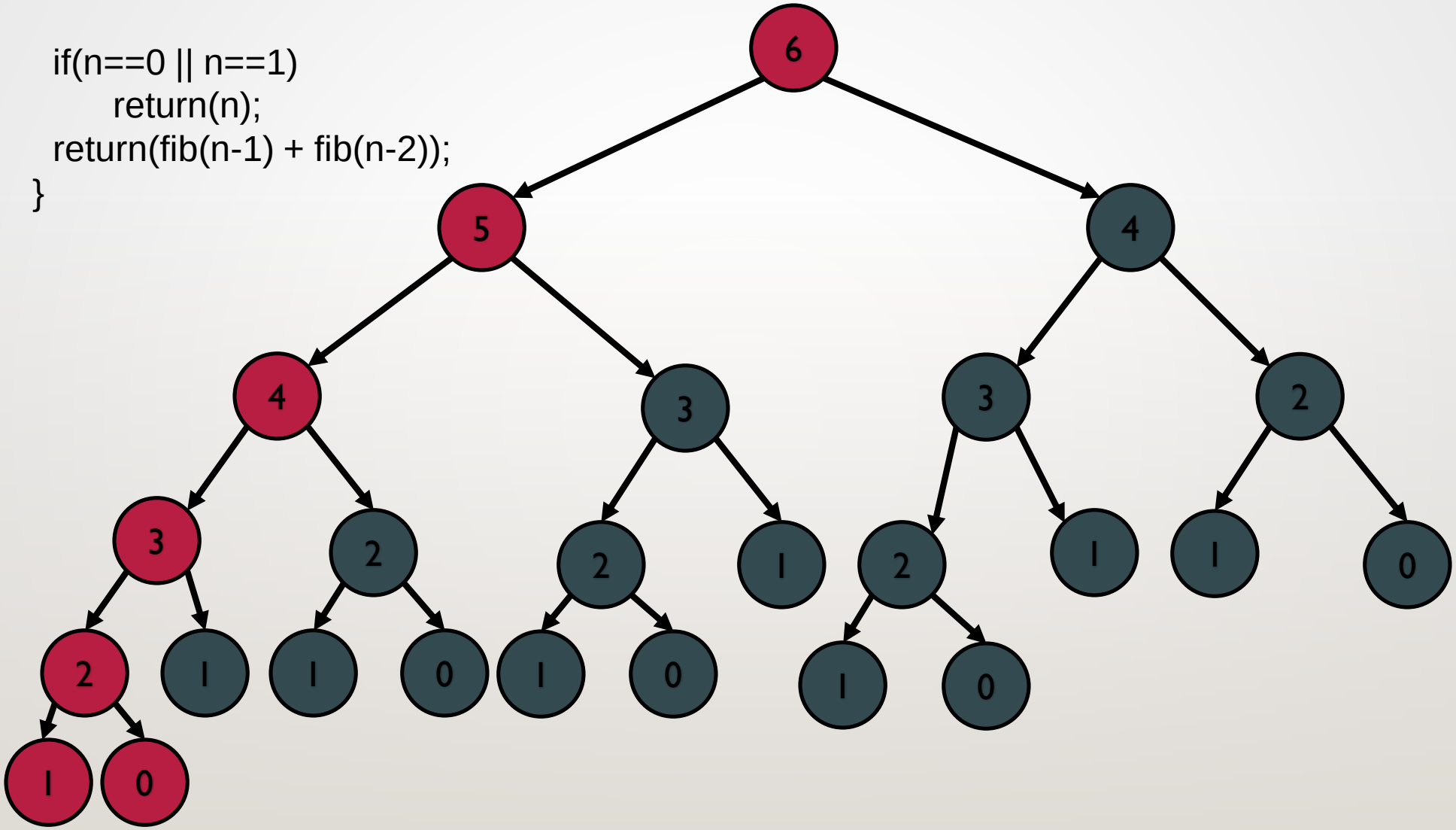
- The dynamic Programming works on a principle of optimality.

- Principle of optimality states that in an optimal sequence of decisions or choices, each sub sequences must also be optimal.

Dynamic programming Example

```
int fib(int n) {

  if(n==0 || n==1)
        return(n);
  return(fib(n-1) + fib(n-2));
}
```



` Dynamic Programming vs.Divide & Conquer

Divide & Conquer	Dynamic Programming
1. Partitions a problem into independent smaller sub-problems	1. Partitions a problem into overlapping sub-problems
2. Doesn't store solutions of sub-problems. (Identical sub-problems may arise - results in the same computations are performed repeatedly.)	2. Stores solutions of sub-problems: thus avoids calculations of same quantity twice
3. Top down algorithms: which logically progresses from the initial instance down to the smallest sub-instances via intermediate sub-instances.	3. Bottom up algorithms: in which the smallest sub-problems are explicitly solved first and the results of these used to construct solutions to progressively larger sub-instances

` Dynamic Programming vs Greedy Method

Dynamic Programming	Greedy Method
1. Dynamic Programming is used to obtain the optimal solution.	1. Greedy Method is also used to get the optimal solution.
2. In Dynamic Programming, we choose at each step, but the choice may depend on the solution to sub-problems.	2. In a greedy Algorithm, we make whatever choice seems best at the moment and then solve the sub-problems arising after the choice is made.
3. It is guaranteed that Dynamic Programming will generate an optimal solution using Principle of Optimality.	3. In Greedy Method, there is no such guarantee of getting Optimal Solution
4. Example: 0/1 Knapsack	4. Example: Fractional Knapsack

Examples

- 0/1 Knapsack Method
- Traveling sales person Problem
- Optimal Binary search tree
- Matrix chain multiplication
- Longest Common Sequence

SAMPLE QUESTIONS

- Differentiate between dynamic programming and greedy method
- Differentiate between dynamic programming and divide and conquer technique
- State principle of optimality
- List out the problems that can be sloved using dynamic programming