Dynamic Programming

Short Definition

- Solves optimization problems
- Similar to Divide and Conquer algorithms
- Overlapping subproblems
- Memoization
- Top down vs bottom up

Steps

1. Determine the structure of the optimal solution
2. Recursively define the value of the optimal solution
3. Compute the optimal solution
4. Construct the optimal solution from subproblems

Use Cases

Various Optimization Problems

DNA sequencing

Longest common subsequence

matrix-chain multiplication

Example - Fibonacci Sequence

```
Fib(n)
    if n == 0
        return 0
    if n == 1
        return 1
    return fib(n - 1) + fib(n - 2)
```

Call stack:

fib(5)
fib(4) + fib(3)
fib(3) + fib(2) + fib(2) + fib(1)
fib(2) + fib(1) + fib(1) + fib(0) + fib(1) + fib(0) + fib(1)
fib(1) + fib(0) + fib(1) + fib(1) + fib(0) + fib(1) + fib(0) + fib(1)

Example - Fibonacci Sequence

```
Memo-Fib(n)
    let m[0..n] be a new array
    for i = 0 to n
        m[i] = 0
    return fib(n, m)

Fib(n, m)
    if m[n] == 0
        m[n] = fib(n - 1, m) + fib(n - 2, m)
    return m[n]
```

Example - Rod Cutting

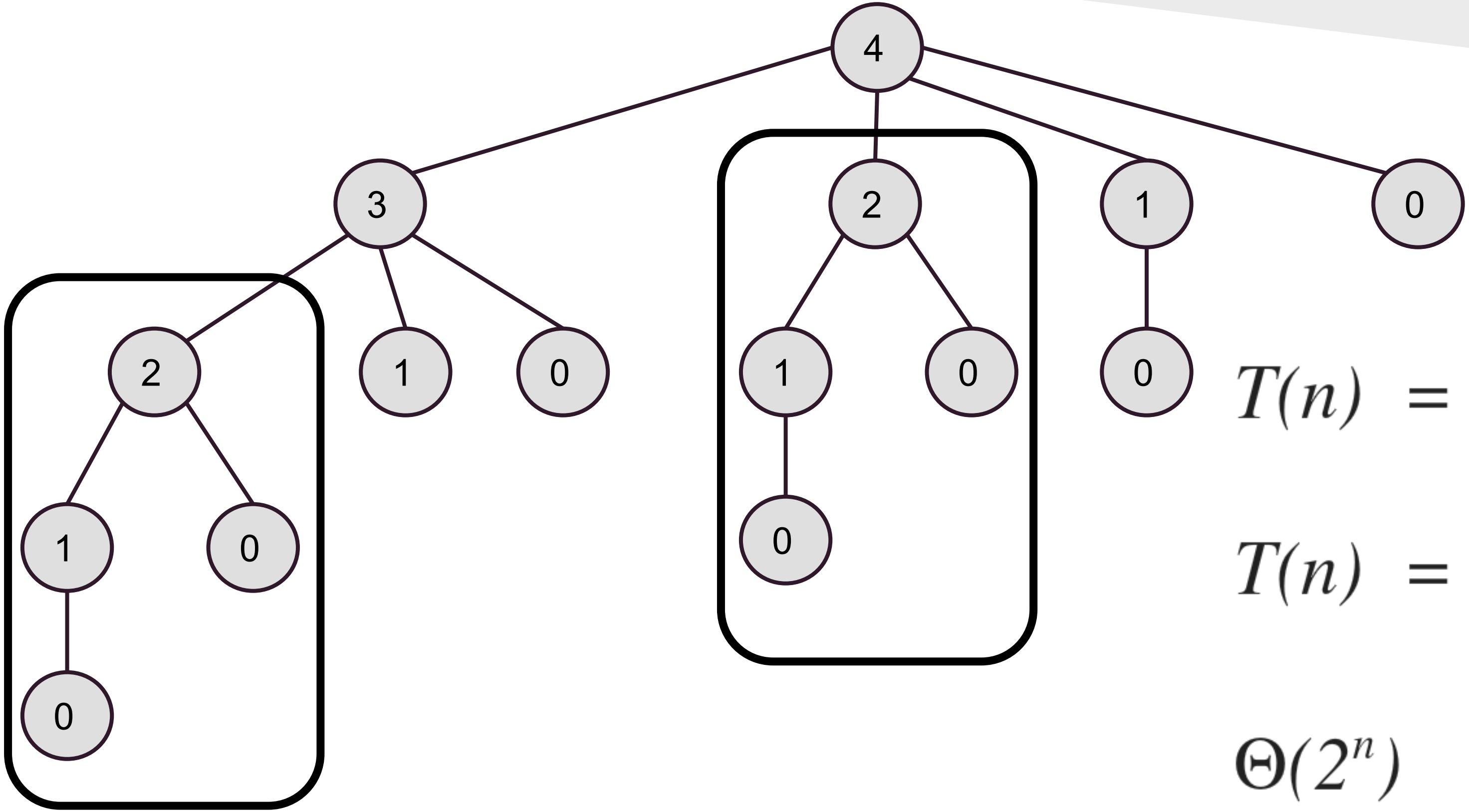
Length i	1	2	3	4	5	6	7	8	9	10
Price P_i	1	5	8	9	10	17	17	20	24	30

Given a rod of length n and a price table P, determine max revenue r_n

Pseudocode

```
Cut-Rod(p, n)
    if n == 0
        return 0
    q = -∞
    for i = 1 to n
        q = max(q, p[i] + Cut-Row(p, n - i))
    return q
```

Cut-Rod(p, 4)



$$T(n) = 1 + \sum_{j=1}^{n-1} T(j)$$

$$T(n) = 2^n$$

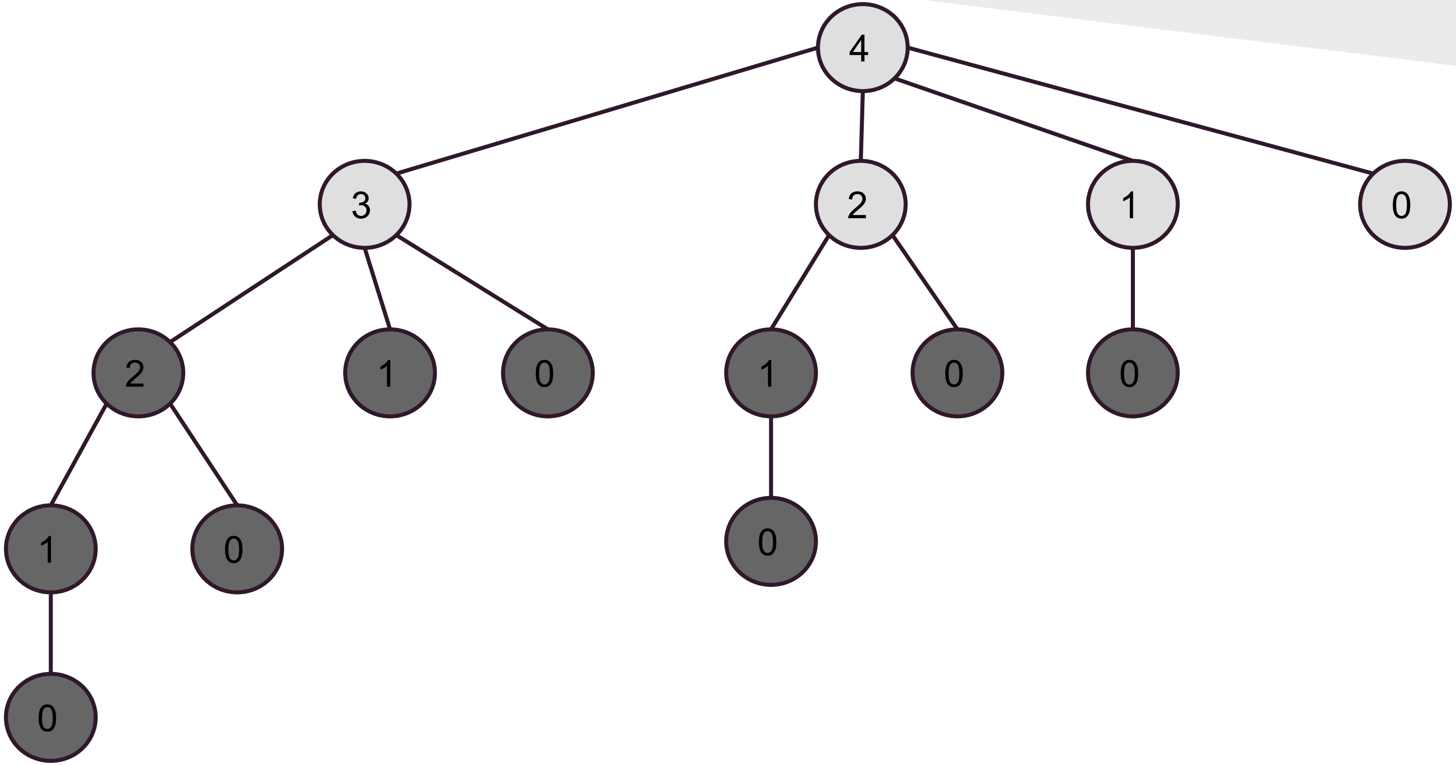
$$\Theta(2^n)$$

Top-Down w/ Memoization

```
New-Cut-Rod(p, n)
    let r[0..n] be a new array
    for i = 0 to n
        r[i] = -∞
    return Memo-Cut-Rod(p, n, r)
```

```
Memo-Cut-Rod(p, n, r)
    if r[n] >= 0
        return r[n]
    if n == 0
        q = 0
    else
        q = -∞
        for i = 1 to n
            q = max(q, p[i] + Memo-Cut-rod(p, n - i, r))
    r[n] = q
    return q
```

New-Cut-Rod(p, 4)



Bottom-Up w/ Memoization

```
Bottom-Up-Cut-Rod(p, n)
    let r[0..n] be a new array
    r[0] = 0
    for j = 1 to n
        q = -∞
        for i = 1 to j
            q = max(q, p[i] + r[j - i])
        r[j] = q
    return r[n]
```

Wrap Up of Cut Rod

Runtime: $\Theta(n^2)$

Code Example

Questions?