```
EXTENDED-BOTTOM-UP-CUT-ROD(p, n)
  let r[0..n] and s[0..n] be new arrays
  r[0] = 0
  for j = 1 to n
      q = -∞
      for i = 1 to j
          if q < p[i] + r[j - i]
              q = p[i] + r[j - i]
              s[j] = i
      r[j] = q
  return r and s

PRINT-CUT-ROD-SOLUTION(p, n)
  (r, s) = EXTENDED-BOTTOM-UP-CUT-ROD(p, n)
  while n > 0
      print s[n]
      n = n - s[n]
```

1

DP Algorithm for Fibonacci Sequence

Sequence defined by

- $a_1 = 1$
- $a_2 = 1$
- $a_n = a_{n-1} + a_{n-2}$

1, 1, 2, 3, 5, 8, 13, 21, 34, …

Recursive algorithm

```
Fib(n)
1.  If n =1 or n=2, then
2.      return 1
3.  Else
4.      a = Fib(n-1)
5.      b = Fib(n-2)
6.      return a+b
```

Bottom Up Dynamic Programming

```
Fib(n)
1.  If n =1 or n=2, then
2.      return 1
3.  Else
4.      f[1]=1; f[2]=1
5.      For i=3 to n
6.          f[i] ← f[i-1]+f[i-2]
7.      return a+b
```

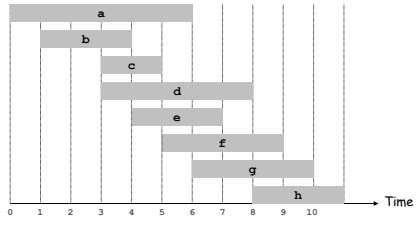
Running Time

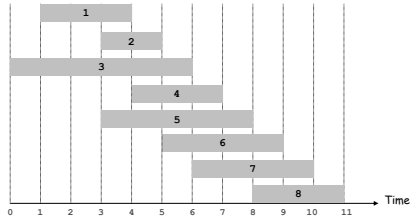
$T(n) = T(n-1) + T(n-2) + \Theta(1)$

$T(n) = \Omega(2^{n/2})$ (exercise: prove using substitution method and the fact that $T(n) \geq 2T(n-2) + 1$)

Running Time: $\Theta(n)$

2





4