Dynamic Programming

Fibonacci numbers

\[ F_0 = F_1 = 1 \]

\[ F_{i+1} = F_i + F_{i-1} \]

\[ F_n \]

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

```
\text{Remember things you already computed}
\text{\underline{Cache} and memorization}
```

```
M[0...n] = -1
Fib(i) = \text{ or } (i \neq 0)
    if (i = 0) return 1
    if M[i] > 0 return M[i]
    r \leftarrow fib(i-1) + fib(i-2)
    M[i] \leftarrow r
    return r.
```

```
\text{Initially}
\text{\underline{Fib}(n)} [\text{\underline{M}}]
M[0] = M[1] = 1
for i = 2 to n do
    \text{\underline{M}}[i] = \text{\underline{M}}[i-1] + \text{\underline{M}}[i-2]
return \text{\underline{M}}[n]
```
**Fib(n):**

\[ a = b = 1 \]

For \( i = 2 \) to \( n \) do

\[ c = a + b \]

\[ a = b \]

\[ b = c \]

return \( c \)

---

**Partition number**

\( p(n) \) is the ways to write \( n \) as a sum of numbers.

- \( 2 = 2, 1+1 \)
- \( 3 = 3, 2+1, 1+1+1 \)
- \( 4 = 4, 3+1, 2+2, 1+1+1+1 \)

**Example:**

\( p(5) = 7 \)

\[ 2^5 = 32 \geq p(5) = 7 \]

\[ \frac{n^2}{2} \leq p(n) \leq 2^n \]

---

**Algorithm:**

\( p(n, d) \) is the number of ways to write \( n \) as a sum of numbers where \( d \) is the first number in the sum.

\[ p(0, d) = \begin{cases} 1 & \text{if } d = 0 \\ 0 & \text{otherwise} \end{cases} \]

For \( i = 1 \) to \( d \): do

\[ n' = n - d \]

\[ s' = p(n', i) \]

For \( i = 1 \) to \( d \) do

\[ s' = p(n'-d, i) + s' \]

return \( s' \).
Recursive function

- # of distinct calls (small)

\[ \frac{\sum_{i}^{n} 2^i}{n} \]

Boundary cases
Base cases

If cache \( g(\ldots, p_n) \) has value, return from cache.

\[ f(\ldots, p_2, p_1) \]

\[ \cup \]

\[ \text{COMPUTATION} \]

store computed value in cache

return it.

Running time

Amortized analysis

\[ \text{# distinct recursive calls} \]

\[ \text{amount of work to compute a single entry} \]

For partition

\[ p(i, d) \]

\[ O(n) \]

\[ \frac{O(n) \text{ distinct recursive calls}}{O(n)} \]

\[ \text{RT: } O(n^3) = O(n^2) \]

Subproblems are defined compactly

Optimal solution has some structure
EDIT DISTANCE

→ HARPELED

XHARX-EVEO

DEL
REPLACE