Algorithm Problem Solving (APS): Dynamic Programming

Niema Moshiri UC San Diego SPIS 2019

Imagine you're in a heist crew en route to your next job



- Imagine you're in a heist crew en route to your next job
- Your target is a truck carrying electronic goods



- Imagine you're in a heist crew en route to your next job
- Your target is a truck carrying electronic goods
- The modified Honda Civic you're driving has a weight limit

- Imagine you're in a heist crew en route to your next job
- Your target is a truck carrying electronic goods
- The modified Honda Civic you're driving has a weight limit
 - You know how much each item costs and weighs

- Imagine you're in a heist crew en route to your next job
- Your target is a truck carrying electronic goods
- The modified Honda Civic you're driving has a weight limit
 - You know how much each item costs and weighs
 - Which items do you steal to maximize your profits?

• Input: A list of n items, where the i-th item has weight w_i and value v_i ,

and a capacity x

• Input: A list of n items, where the i-th item has weight w_i and value v_i , and a capacity x

Output: A set of items such that the sum of their weights is below x
 and the sum of their values is maximized

W:	3	8	10	6
V:	\$5	\$5	\$13	\$6

$$x = 20$$

Ex

Can we use a Greedy Algorithm?

W:	3	8	10	6
V:	\$5	\$5	\$13	\$6

$$x = 20$$

Ex

Can we use a Greedy Algorithm?

W:	3	8	10	6
V:	\$5	\$5	\$13	\$6

$$x = 20$$

Nope! Not guaranteed to be optimal!

• The first number is **0**



- The first number is 0
- The second number is 1

0, 1

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3, 5

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3, 5

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3, 5, 8

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3, 5, 8

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3, 5, 8, **13**

- The first number is 0
- The second number is 1
- Each subsequent number is the sum of the two before it

0, 1, 1, 2, 3, 5, 8, 13, ...

• Input: An integer n

• Input: An integer n

Output: The n-th number of the Fibonacci sequence (0-based indexing)

- Input: An integer n
- Output: The *n*-th number of the Fibonacci sequence (0-based indexing)

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

Write a recursive Python function!

Output: The n-th number of the Fibonacci sequence (0-based indexing)

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

Recursive Fibonacci

```
def fib rec(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib_rec(n-1) + fib_rec(n-2)
```

Recursive Fibonacci

```
def fib rec(n):
   if n == 0:
                                   Yay! 😄
       return 0
   elif n == 1:
       return 1
    else:
       return fib rec(n-1) + fib rec(n-2)
```

Recursive Fibonacci

```
def fib rec(n):
                              Hold up... 🦂
   if n == 0:
       return 0
   elif n == 1:
       return 1
   else:
       return fib rec(n-1) + fib rec(n-2)
```

Recursive Fibonacci: SLOW!!!

rec_fib(5)

```
def fib_rec(n):
    if n in {0,1}:
        return n
    return rec_fib(n-1) + rec_fib(n-2)
```

rec_fib(5)

```
def fib_rec(n):
    if n in {0,1}:
        return n
    return rec_fib(n-1) + rec_fib(n-2)
```

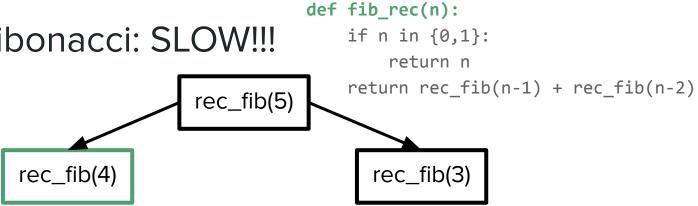
rec_fib(5)

```
def fib_rec(n):
    if n in {0,1}:
        return n
    return rec_fib(n-1) + rec_fib(n-2)
```

rec_fib(5)

```
def fib_rec(n):
    if n in {0,1}:
        return n
    return rec_fib(n-1) + rec_fib(n-2)
```

```
def fib_rec(n):
                                 if n in {0,1}:
                                     return n
                                 return rec_fib(n-1) + rec_fib(n-2)
                  rec_fib(5)
rec_fib(4)
                                   rec_fib(3)
```



```
def fib_rec(n):
                                 if n in {0,1}:
                                     return n
                                 return rec_fib(n-1) + rec_fib(n-2)
                 rec_fib(5)
rec_fib(4)
                                   rec_fib(3)
```

```
def fib_rec(n):
                                 if n in {0,1}:
                                     return n
                                 return rec_fib(n-1) + rec_fib(n-2)
                 rec_fib(5)
rec_fib(4)
                                   rec_fib(3)
```

Recursive Fibonacci: SLOW!!! rec_fib(5) rec_fib(4) rec_fib(2) rec_fib(2) rec_fib_rec(n): if n in {0,1}: return n return rec_fib(n-1) + rec_fib(n-2) rec_fib(3)

def fib_rec(n): Recursive Fibonacci: SLOW!!! if n in {0,1}: return n return rec_fib(n-1) + rec_fib(n-2) rec_fib(5) rec_fib(4) rec_fib(3) rec_fib(2) rec_fib(3)

Recursive Fibonacci: SLOW!!!

return n

return rec_fib(n-1) + rec_fib(n-2)

rec_fib(2)

rec_fib(4)

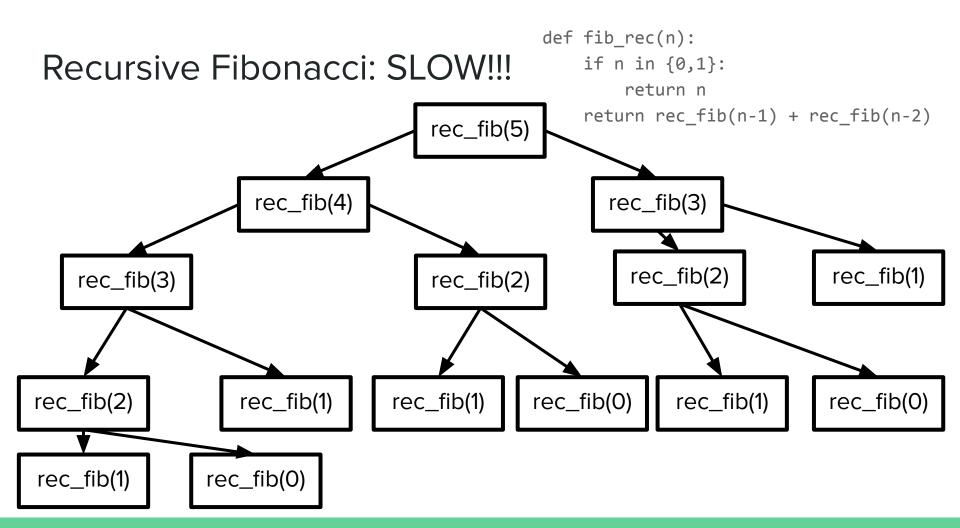
rec_fib(3)

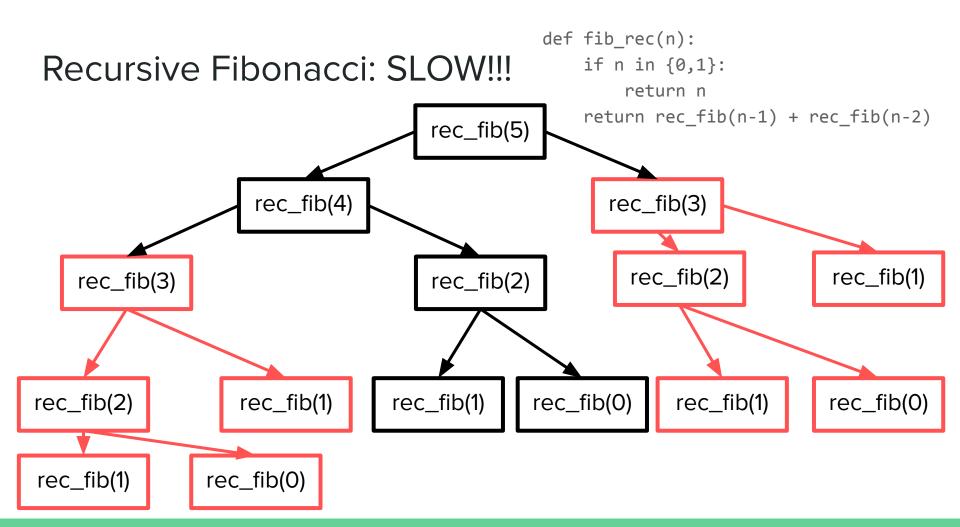
def fib_rec(n):

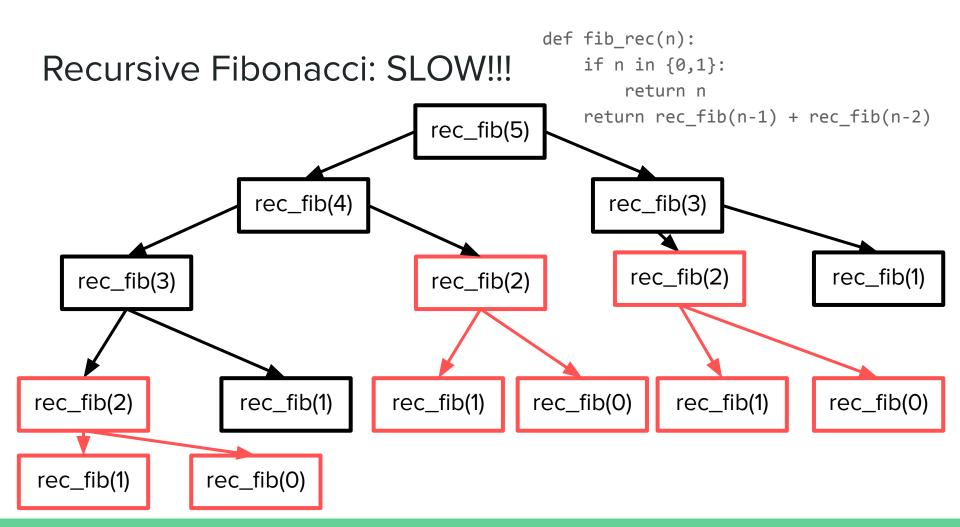
rec_fib(3)

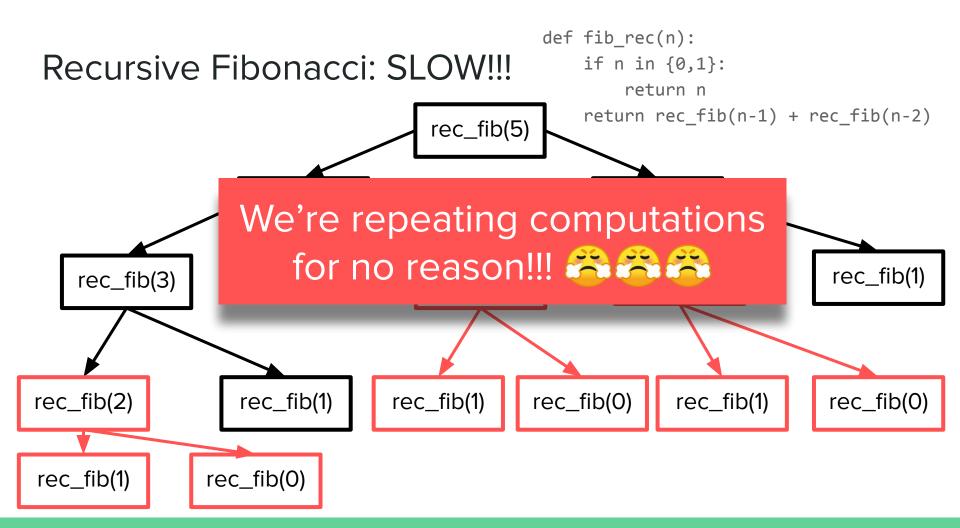
Recursive Fibonacci: SLOW!!! rec_fib(5) rec_fib(4) rec_fib(2) rec_fib(2) rec_fib_rec(n): if n in {0,1}: return n return rec_fib(n-1) + rec_fib(n-2)

def fib_rec(n): Recursive Fibonacci: SLOW!!! if n in {0,1}: return n return rec_fib(n-1) + rec_fib(n-2) rec_fib(5) rec_fib(4) rec_fib(3) rec_fib(2) rec_fib(3) rec_fib(2) rec_fib(1)









2 Fast 2 Fibonacci

We saw that, in computing rec_fib(5), we ended up calling rec_fib(3)

twice and rec_fib(2) two times

2 Fast 2 Fibonacci

We saw that, in computing rec_fib(5), we ended up calling rec_fib(3)
 twice and rec_fib(2) two times

The values of rec_fib(3) and rec_fib(2) never change, so once we compute them the first time, why not just save them somewhere?

2 Fast 2 Fibonacci

- We saw that, in computing rec_fib(5), we ended up calling rec_fib(3)
 twice and rec_fib(2) two times
 - The values of rec_fib(3) and rec_fib(2) never change, so once we compute them the first time, why not just save them somewhere?
- Memoization: Saving the results of expensive functions somewhere and returning the saved result when the same inputs occur again

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

n:	0	1	2	3	4	5
fib(<i>n</i>):						

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$
 fast_fib(5)

n:	0	1	2	3	4	5
fib(<i>n</i>):	0					

$$F(n) = egin{cases} 0 & ext{if } n=0 \ 1 & ext{if } n=1 \ F(n-1)+F(n-2) & ext{if } n>1 \end{cases}$$
 fast_fib(5)

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1				

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$
 fast_fib(5)

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1			

$$F(n)=egin{cases} 0 & ext{if } n=0 \ 1 & ext{if } n=1 \ F(n-1)+F(n-2) & ext{if } n>1 \end{cases}$$
 ast_fib(5)

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2		

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2	3	

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2	3	5

$$F(n)=egin{cases} 0 & ext{if } n=0 \ 1 & ext{if } n=1 \ F(n-1)+F(n-2) & ext{if } n>1 \end{cases}$$
 fast_fib(5)

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2	3	5

1. **Define** the solution **recursively** from top-down

1. **Define** the solution **recursively** from top-down

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results

n:	0	1	2	3	4	5
fib(<i>n</i>):						

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results
- 3. **Fill** the **memoization table** from bottom-up

n:	0	1	2	3	4	5
fib(<i>n</i>):	0					

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results
- 3. **Fill** the **memoization table** from bottom-up

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1				

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results
- 3. **Fill** the **memoization table** from bottom-up

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1			

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results
- 3. **Fill** the **memoization table** from bottom-up

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2		

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results
- 3. **Fill** the **memoization table** from bottom-up

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2	3	

$$F(n) = \begin{cases} 0 & \text{if } n = 0\\ 1 & \text{if } n = 1\\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- 1. **Define** the solution **recursively** from top-down
- 2. Create a memoization table to store the results
- 3. **Fill** the **memoization table** from bottom-up

n:	0	1	2	3	4	5
fib(<i>n</i>):	0	1	1	2	3	5

• Input: A list of n items, where the i-th item has weight w_i and value v_i , and a capacity x

Output: A set of items such that the sum of their weights is below x
 and the sum of their values is maximized

Let P(i, j) denote the maximum possible profit of a selection of the first i

items that have a total weight of at most *j*

- Let P(i, j) denote the maximum possible profit of a selection of the first i
 items that have a total weight of at most j
- When we haven't chosen any items (i = 0), the profit must be 0

- Let P(i, j) denote the maximum possible profit of a selection of the first i
 items that have a total weight of at most j
- When we haven't chosen any items (i = 0), the profit must be 0
 - \circ **P(0, j)** = **0** for all $0 \le j \le x$

- Let P(i, j) denote the maximum possible profit of a selection of the first i items that have a total weight of at most j
- When we haven't chosen any items (i = 0), the profit must be 0
 - \circ **P(0, j)** = **0** for all $0 \le j \le x$
- Negative weights (j < 0) are impossible, so they should be disallowed

- Let P(i, j) denote the maximum possible profit of a selection of the first i
 items that have a total weight of at most j
- When we haven't chosen any items (i = 0), the profit must be 0
 - P(0,j) = 0 for all $0 \le j \le x$
- Negative weights (j < 0) are impossible, so they should be disallowed
 - $P(i, j) = -\infty$ for all j < 0

• For all remaining $1 \le i \le n$ and $0 \le j \le x$, we could either:

- For all remaining $1 \le i \le n$ and $0 \le j \le x$, we could either:
 - Add item i to our pack, so our profit would be $v_i + P(i-1, j-w_i)$

- For all remaining $1 \le i \le n$ and $0 \le j \le x$, we could either:
 - Add item i to our pack, so our profit would be $v_i + P(i-1, j-w_i)$
 - We get the value of item $i(v_i)$ added to the best scenario of looking at only the first i-1 items with j- w_i total weight

- For all remaining $1 \le i \le n$ and $0 \le j \le x$, we could either:
 - Add item i to our pack, so our profit would be $v_i + P(i-1, j-w_i)$
 - We get the value of item $i(v_i)$ added to the best scenario of looking at only the first i-1 items with j- w_i total weight
 - Don't add item i to our pack, so our profit would be P(i-1, j)

- For all remaining $1 \le i \le n$ and $0 \le j \le x$, we could either:
 - Add item i to our pack, so our profit would be $v_i + P(i-1, j-w_i)$
 - We get the value of item $i(v_i)$ added to the best scenario of looking at only the first i-1 items with j- w_i total weight
 - Don't add item i to our pack, so our profit would be P(i-1, j)
 - No value added, so it's just our best scenario of looking at the first *i*-1 items with the same (*j*) total weight

- For all remaining $1 \le i \le n$ and $0 \le j \le x$, we could either:
 - Add item i to our pack, so our profit would be $v_i + P(i-1, j-w_i)$
 - We get the value of item $i(v_j)$ added to the best scenario of looking at only the first i-1 items with j- w_i total weight
 - Don't add item i to our pack, so our profit would be P(i-1, j)
 - No value added, so it's just our best scenario of looking at the first *i*-1 items with the same (*j*) total weight
 - Take the maximum of these two possible options

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1, j - w_i), P(i-1, j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1, j-w_i), P(i-1, j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

<i>V</i> : \$5	\$5	\$13	\$6

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ j (total weight)} \end{cases}$$

									<i>)</i> (Otal	weig	111)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0																					
1																					
2																					
3																					
4																					

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\left\{v_i + P(i-1,j-w_i), P(i-1,j)\right\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \end{cases}$$

$$j \text{ (total weight)}$$

					ĺ					ĺ	weigi				ĺ						Т
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0																					
1																					
2																					
3																					
4																					

W:	3	8	10	6
V:	\$5	\$5	\$13	\$6

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1, j-w_i), P(i-1, j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

									J) (T	otal '	weig	nt)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1																					
2																					
3																					
4																					

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ j (total weight)} \end{cases}$$

									<i>)</i> (.Otal	weig	111)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1																					
2																					
3																					
4																					

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

										<i>)</i> (ι	Otal	weigi	111)									
<u>_</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s see	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
items seen)	1	?																				
פר ס	2																					
/ (number of	3																					
₹	4																					

$$P(i,j) = \begin{cases} 0 & \text{if } i=0 \\ -\infty & \text{if } j<0 \\ \max\left\{v_i + P(i-1,j-w_i), P(i-1,j)\right\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \text{ j (total weight)} \end{cases}$$

				-					_	<i>j</i> (t	otal	weig	ht)				_					
Ē		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
items seen)	0	0	0	0	0	0	0	(Λ.		ф г					0	0	0	0	0	0
	1	?								Ac	Id:	\$!) +	0	0							
i (number of	2																					
qunu	3																					
<i>i</i> (r	4																					

W:	3	8	10	6
V:	\$5	\$5	\$13	\$6

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max \left\{ v_i + P(i-1,j-w_i), P(i-1,j) \right\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \text{ j (total weight)} \end{cases}$$

										J (1	.ota.	weig	,									
Ē		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
items seen)	0	0	0	0	0	0	0			Λ.		ф г					0	0	0	0	0	0
item	1	?							,	Ac	Id:	\$!) +	0	0							
er of	2									_		_										
(number of	3								Г	$) \cap I$	า ' †	A	44	. ф	\cap							
j.	4								_		- L		au	. ψ	0							

Example: The O-1 Knapsack Problem
$$v$$
: $\begin{vmatrix} x = 20 & w \\ & & \end{vmatrix}$

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \end{cases}$$

$$j \text{ (total weight)}$$

≘		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	0	0	0	0	0	0	0										0	0	0	0	0	0
	1	0								Ac	ld:	\$!	5 +	0	0							
	2																					
	3								Г)oı	า't	Δα	hh	٠ \$	\cap							
	4										1 (au	. Ψ								

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

										<i>)</i> (ι	.otai	weig	nt)									
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		0																				
2	2	0																				
3	3	0																				
4	ļ.	0																				

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

										<i>)</i> (ι	otai	weig	nt)									
Ē		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
items seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0																			
er of	2	0	0																			
i (number of	3	0	0																			
<i>i</i> (r	4	0	0																			

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ j (total weight)} \end{cases}$$

										J) (T	otai	weig	nt)									
Ē		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
items seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0																		
er of	2	0	0	0																		
i (number of	3	0	0	0																		
<i>j</i> (r	4	0	0	0																		

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

									<i>J</i> (t	oτal	weig	ht)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	-5																	
2	0	0	0																		
3	0	0	0																		
4	0	0	0																		

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1, j-w_i), P(i-1, j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

										<i>J</i> (1	otal	weig	nt)									
	(0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	(0	0	0	5																	
2	(0	0	0	5																	
3	(0	0	0																		
4	(0	0	0																		

W:	3	8	10	6
V:	\$5	\$5	\$13	\$6

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

										<i>j</i> (1	otal	weig	ht)									
<u>(</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
items seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	5																	
er of	2	0	0	0	1 ₅																	
<i>i</i> (number of	3	0	0	0	5																	
<i>i</i> (r	4	0	0	0	T ₅																	

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \end{cases}$$

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \text{ j (total weight)} \end{cases}$$

									_	<i>j</i> (†	total	weig	ht)									
<u>5</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	√ 5															
er of	2	0	0	0	1 5	5	1 ₅															
i (number of	3	0	0	0	5	5 •	5															
) (r	4	0	0	0	T ₅	T ₅	T ₅															

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

									<i>J</i> (1	totai	weig	nt)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	1 5	1 5														
2	0	0	0	 	 5	 4 5	1 ₅														
3	0	0	0	₅	 5	 4 5	1 ₅														
4	0	0	0	15	5	1 ₅															

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

									<i>)</i> (1	total	weig	ht)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	5	5														
2	0	0	0	1 5	1 5 1 1 1 1 1 1 1 1 1 1	1 ₅	1 ₅														
3	0 -	0	0	5	5	5	1 ₅														
4	0	0	0	5	T ₅	T ₅	6														

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ j (total weight)} \end{cases}$$

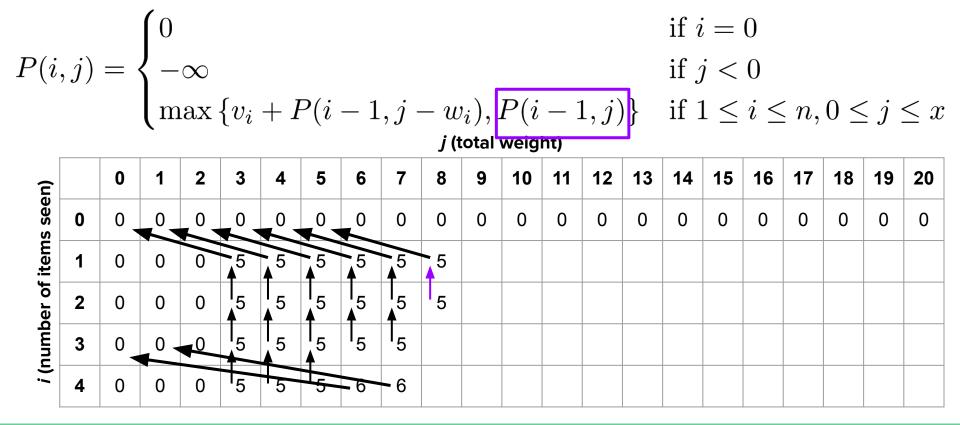
J (total weight)																					
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	1 5	5	√ 5	1 5													
2	0	0	0	5 •	 4	 4 5	 4 5	 4 5													
3	0	0 -	0	5	 5	 4	1 ₅	¹ 5													
4	0	0	0	5	5	5	- 6	- 6													

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \end{cases}$$

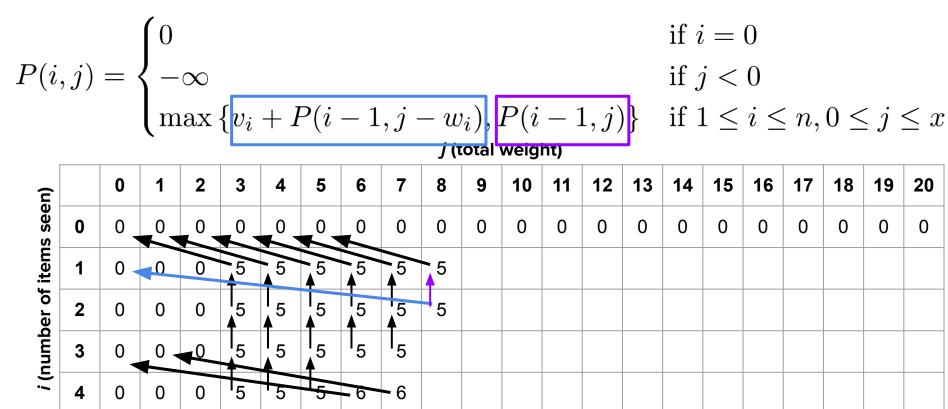
$$j \text{ (total weight)}$$

										<i>j</i> (t	total	weig	ht)									
Ē		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	1 5 1	5	5	5												
er of	2	0	0	0	5	5	1 5 ▲	5	5													
(number of	3	0	0 -	0	5	5	1 ₅	1 ₅	1 ₅													
) (r	4	0	0	0	5	5		-6	- 6													

8 10 6 W: x = 20\$5 \$13 \$6







10

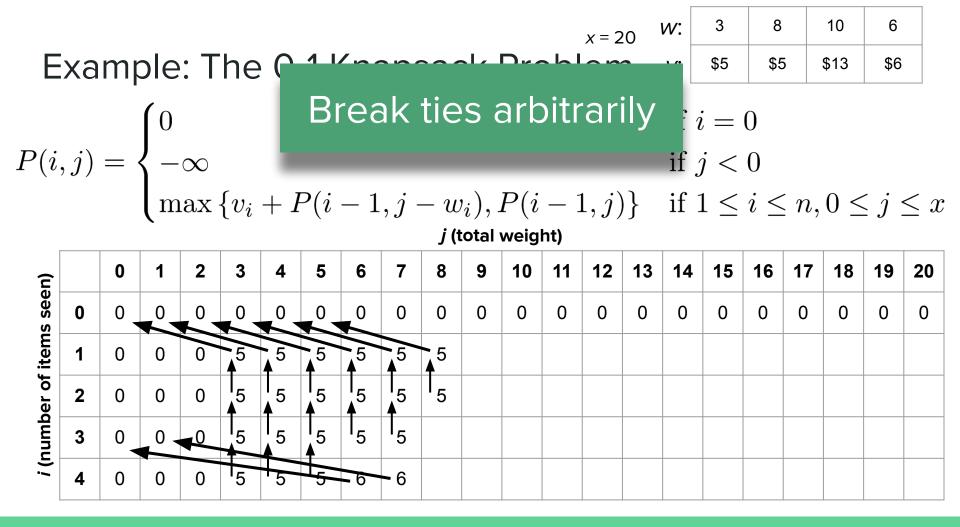
\$13

\$5

W:

6

\$6



$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

			`							j (†	total	weig	ht)									
رة		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5												
er of	2	0	0	0	5	5	1 5 ▲	5	5	5												
(number	3	0	0.	0	5	5	1 ₅	1 ₅	15	5												
; ;	4	0	0	0	5	5	5	- 6	-6	- 6												

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{j (total weight)} \end{cases}$$

_										<i>J</i> (1	ιοται	weig	nt)									
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	5	5	5	5	5	1 5	√ 5											
	2	0	0	0	1 ₅	5 •	1 ₅	5	1 ₅	1 ₅	1 ₅											
	3	0	0 -	0	5	5	 5	1 ₅	1 ₅	15	1 ₅											
	4	0	0	0	1 5	5	1 5	- 6	- 6	- 6												

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

			`							<i>j</i> (1	total	weig	ht)									
<u>5</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5	5											
er of	2	0	0	0	 5	 5	1 ₅	5 •	1 ₅	1 5	1 ₅											
(number	3	0	0 -	0	5	5	5	5	1 ₅	15	1 ₅											
, T	4	0	0	0	5	5	5	- б	- 6	- 6	-11											

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ } j \text{ (total weight)} \end{cases}$$

										<i>j</i> (†	total	weig	ht)									
מפפוו)		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	5	5	5	5	5	5	1 5	5										
	2	0	0	0	1 ₅	5	1 ₅	5	5	5	5	1 ₅										
	3	0	0 -	0	5	5	5	5	1 ₅	15	15											
	4	0	0	0	5	5	5	- б	-0	- 6	-11											

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \hline j \text{ (total weight)} \end{cases}$$

Ē		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
items seen)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	5	5	5	5	5	5	5	5										
er of	2	0 -	0	0	5	5	 5	5	5	 5	5	1 ₅										
number	3	0	0 -	0	5	5	5	1 ₅	1 ₅	15	15	13										
<i>i</i> (r	4	0	0	0	5	5	5	- б	- 6	0	-11											

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \text{ j (total weight)} \end{cases}$$

									<i>j</i> (1	total	weig	ht)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	1 5 1 5	5	5	5	1 5 1 5	5										
2	0 -	0	0	5	5	5	1 ₅	1 ₅	5	1 ₅	1 ₅										
3	0	0 -	0	5	5	5	5	5	5	5	13										
4	0	0	0	15	5	5	- 6	- 6	- 6	-11	13										

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

										<i>j</i> (t	total	weig	ht)									
<u>C</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0.	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5	5	5	5									
er of	2	0 •	0	0	5	5	5	5	1 ₅	5	1 ₅	1 ₅										
(number	3	0	0 -	0	5	5	5	5	1 ₅	5	5	13										
) (r	4	0	0	0	+ ₅	5	5	- б	6	- 6	-11	T ₁₃										

Example: The O-1 Knapsack Problem $v: \begin{bmatrix} x = 20 \\ v : \end{bmatrix}$ \$5 \$13 \$6

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ j (total weight)} \end{cases}$$

			`							<i>j</i> (t	total	weig	ht)									
<u></u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0.	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5	5	5	5									
er of	2	0 -	10	0	5	5	5	1 5	 5	j 5	5	5	10									
(number	3	0	0 -	0	5	5	5	5	5	5	5	13										
<i>i</i> ,	4	0	0	0	5	5	5	- б	0	- 6	-11	43										

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ } j \text{ (total weight)} \end{cases}$$

										j (†	total	weig	ht)									
<u>E</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0,	0	0	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5	1 5 1 5	5	5									
er of	2	0 -	0	10	5	5	5	5	5	5	5	5	10									
(number	3	0	0 -	0	5	5	5	5	5	5	5	13	13									
, T	4	0	0	0	5	5	1 5	- б	-0	- 6	-11	43	13									

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0.	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	5	5	5	5	5	5	5	5								
2	0 -	-0 -	0-	5	5	5	5	5	1 ₅	5	5	10	10								
3	0	0 -	0	5	5	5	5	5	5	5	13	13	13								
4	0	0	0	15	5	5	- 6	-6	- 6	-11	13	13	13								

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \\ & \text{ j (total weight)} \end{cases}$$

										<i>j</i> (t	otal	weig	ht)									
ت		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0_	0	0-	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5	5	5	5	5	5							
er of	2	0 -	0	0	5	5	5	5	5	1 5 ▲	1 5	5	10	10	10							
(number	3	0	0 -	0	5	5	5	5	5	5	5	13	13	13								
),(r	4	0	0	0	5	15	5	- б	- 6	- 6	-11	13	13	13								

		0													if i	; =	0					
P(i, j	() =	$\left\{ -\right.$	∞												if j	<i>i</i> <	0					
		$\lfloor m \rfloor$	ax	$\{v_i$	+1	P(i)	- 1	,j	-u	(i)	P(i	_ í	[1,j])}	if 1	l ≤	$i \leq$	$\leq n,$	0 <	$\leq j$	$\leq x$	•
										total												
<u> </u>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

8

\$5

W:

10

\$13

6

\$6

										<i>j</i> (t	otal	weig	ht)									
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	0	0	0	0	0	0	0	0	0	0	0.	0-	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	5	5	5	\ <u>{</u>	5	5	5	5	5	5	5							
	2	0 -	0	0	5	5	\ ₅	5	5	5	1 5	5	10	10	10							
	3	0	0 -	0	5	5	5	5	5	5	5	13	13	13	18							
•	4	0	0	0	5	 5	5	- 6	- 6	- 6	-11	13	13	13								

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ & \text{ j (total weight)} \end{cases}$$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0.	0-	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	5	5	5	5	5	5	5	5	5							
2	0 -	0	0	5	5	5	5	5	5	5 •	5	10	10	10							
3	0	0 -	0	5	5	5	5	5	5	15	13	13	13	18							
4	0	0	0	5	5	$\frac{1}{5}$	- б	- б	6	-11	43	13	կ ₃	18							

6

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \end{cases}$$

$$\underbrace{\int_{j \text{ (total weight)}}}$$

									<i>)</i> ('	Ulai	weig	111,									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0.	0-	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	5	5		5	5	5	5	5	5	5	5					
2	0 -	0	0	5	5=	5	5	5	5	5	5	10	10	10	10	10					
3	0	0 -	0	5	5	5	5	5	5	5	13	13	13 •	18	18	18					
4	0	0	0	15	5	5	- б	- 6	- 6	-11	43	13	43	18	1 ₁₈	18					

<i>x</i> = 20					
Example: The 0-1 Knapsack Problem	V:	\$5	\$5	\$13	\$6
·					

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \hline j \text{(total weight)} \end{cases}$$

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ j \text{ (total weight)} \end{cases}$$

_			_							<i>j</i> (t	total	weig	ht)									
ج ا		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0.	0.	0-	0	0	0	0	0	0	0	0	0	0
items	1	0	0	0	5	5	5	5	5	5	5	5	5	5	5	5	5	5				
er of	2	0 -	10-	0	5	5		54	5	5	5	5	10	10	10	10	10	10				
(number	3	0	0.	0	5	5	5	5	5	5	5	13-	13	13	18	18	18	18				
),(r	4	0	0	0	5	5	5	- 6	- 0	-6	-11	13	13	13	18	18	18	- 19				

<i>x</i> = 20					
Example: The 0-1 Knapsack Problem	V:	\$5	\$5	\$13	\$6
·					

3

4

<i>x</i> = 20	W:	3	8	10	6	
lem	V:	\$5	\$5	\$13	\$6	

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1,j-w_i), P(i-1,j)\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \hline j \text{ (total weight)} \end{cases}$$

ካ3

13 13 13 18 18 18

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1, j-w_i), P(i-1, j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

			m	ax	$\{v_i$	+I	P(i)	– 1	,j	-u	$\langle i \rangle, i$	P(i		(1,j))	if I	1 ≤	$i \leq$	n	0 <	$\leq j$	$\leq x$
-				ı	1 1			i		<i>j</i> (t	total	weig	ht)		ī	i	ī					
<u></u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0_	0.	0-	0	0	0-	0-	0_	0.	0 -	0	0	0
items	1	0	0	0	5	5	5	5	5		5	5	5.	5	5	5	5	5	5	5	5	5
oer of	2	0 -	0	0-	5	5 -	5	5	5		5	5.	110	10	10	10	10	10	10	10	10	10
number	3	0	0 -	0	5	5	1 ₅	5	5	15	5	13-	13-	13.	18.	18	18	18	18	18	18	18
),	4	0	0	0	15	5	15	- 6	- 0	- 6	-11	կ3	¹ 13	կ3	18	18	18	-19	-19	-19 -	19	19

Example: The O-1 Knapsack Problem v: $\begin{bmatrix} x = 20 \\ y = 20 \end{bmatrix}$ v: $\begin{bmatrix} 3 \\ 5 \end{bmatrix}$ $\begin{bmatrix} 8 \\ 10 \end{bmatrix}$ $\begin{bmatrix} 6 \\ 5 \end{bmatrix}$

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\left\{v_i + P(i-1,j-w_i), P(i-1,j)\right\} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \end{cases}$$

$$j \text{ (total weight)}$$

		_							<i>j</i> (t	otal	weig	ht)									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	5	5	5	152	5	5	5) 	5	5	5	5	5	5	5	5	5	5
2	0 -	0-	0	5	54	5	5	5.4	4	5.	5	110	10	10	10	10	10	10	10	10	10
3	0	0 -	0	5	5	5	5	5	5	15	13-	13-	13.	18	18	18	18	18	18	18	18
4	0	0	0	15	15	5	_ ნ	- б	- 6	-11	43	13	13	18	18	18	-19	-19	-19 -	19	19

Example: The O-1 Knapsack Problem v: $\begin{bmatrix} x = 20 \\ 5 \end{bmatrix}$ $\begin{bmatrix} x = 20 \\ 5 \end{bmatrix}$

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max\{v_i + P(i-1, j-w_i), P(i-1, j)\} & \text{if } 1 \le i \le n, 0 \le j \le x \end{cases}$$

$$j \text{ (total weight)}$$

			m	ax	$\{v_i$	+I	P(i)	– 1	,j	-w	(i), i	P(i		(1, j)	}	if 1	1 <	$i \leq$	n	0 <	$\leq j$:	$\leq x$
-					i			ī		<i>j</i> (t	otal	weig	ht)	1			1					
<u>Ē</u>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s seen)	0	0	0	0	0	0	0	0	0	0	0.	0-	0	0	0	0	0	0	0 -	0	0	0
items	1	0	0		5	5	5	5-	5	5	5	5	5	5	<u>5</u>	5	5	5	5	5	5	5
er of	2	0 •	0-	0	5	- 54		5	5.4		5	5 <	110	10	10	10	10	10	10	10	10	10
number	3	0	0 -	0	5	5	5	5	5	5	5	13-	13-	13.	18.	18	18	18	18	18	18	18
), 	4	0	0	0	15	5	5	- 0	0	0	- 11	43	13	43	18	18	18	19	19	-19 -	19	1 9



3

4

$$P(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ -\infty & \text{if } j < 0 \\ \max{\{v_i + P(i-1, j-w_i), P(i-1, j)\}} & \text{if } 1 \leq i \leq n, 0 \leq j \leq x \\ \hline j \text{(total weight)} \end{cases}$$

ካ3

13-13-13-18-18-18

\$13

\$6