## Assignment Project Exam Help Add WeChat powcoder

Companyaming (2)

Add WeChat powcoder Jérôme Waldispühl

School of Computer Science McGill University

Based on (Kleinberg & Tardos, 2005) & Slides by K. Wayne

# Assignment Project Exam Help Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

#### SINGLE SOURCE SHORTEST PATHS

#### Assignment Project Exam Help

#### Modelingas, graphs

#### Input:

- Directed graph G = (V, E)
- Weight function  $w : E \rightarrow R$

Weight of path 
$$p = \langle v_0, v_1, v_2 \rangle$$
  
Assignment Project Exam Help

$$= \sum_{k=1}^{n} w(v_{k-1}, v_{k-1}) = \sum_$$

= sum of edge weightspeechat powcoder

#### **Shortest-path weight** *u* to *v*:

$$\delta(u,v) = \begin{cases} \min \left\{ w(p) : u \mapsto^p v \right\} & \text{if there exists a path } u \rightsquigarrow v. \\ \infty & \text{Otherwise.} \end{cases}$$

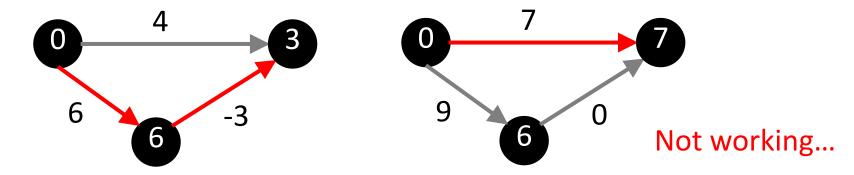
Shortest path u to v is any path p such that  $w(p) = \delta(u, v)$ . Generalization of breadth-first search to weighted graphs.

#### Assignment Project Exam Help Dijkstra's algorithm

- No negative-weight edges.
- Weighted version of BFS:
  - Instead of a FIFO queue, uses a priority queue.
    Assignment Project Exam Help
  - Keys are shortest-path weights (d[v]).
- Greedy choice: Attento step we choose the light edge.

#### How to deal with negative weight edgesoder

- Allow re-insertion in queue? ⇒ Exponential running time...
- Add constant to each edge?



#### Assignment Project Exam Help Bellman-Fard Algorithm

- Allows negative-weight edges.
- Computes a [v] and π[v] for all v ∈ v.
- Returns TRUE TROWN Megative Weight cycles reachable from the fatherwiseer

If Bellman-Ford has not converged after V(G) - 1 iterations, then there cannot be a shortest path tree, so there must be a negative weight cycle.

#### Assignment Project Exam Help Bellman-Ford Algorithm

- Can have negative-weight edges.
- Will "detect" reachable negative-weight cycles.

```
Assignment Project Exam Help
```

```
Initialize(G, s);

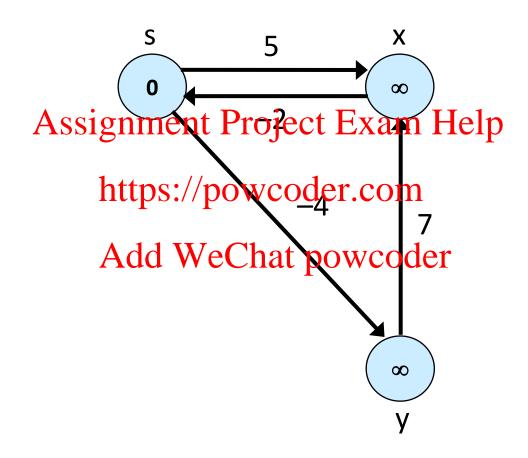
fohttps://ppv/gpder.gom

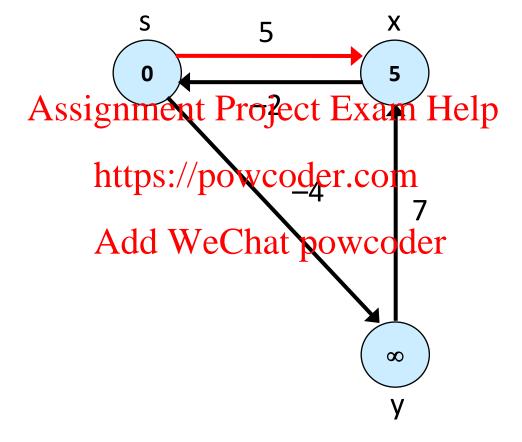
for each (u, v) in E[G] do
Add WeChat powcoder
Relax(u, v, w)

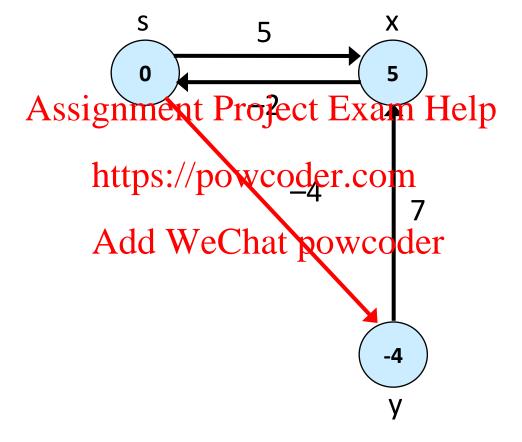
for each (u, v) in E[G] do
if d[v] > d[u] + w(u, v) then
return false

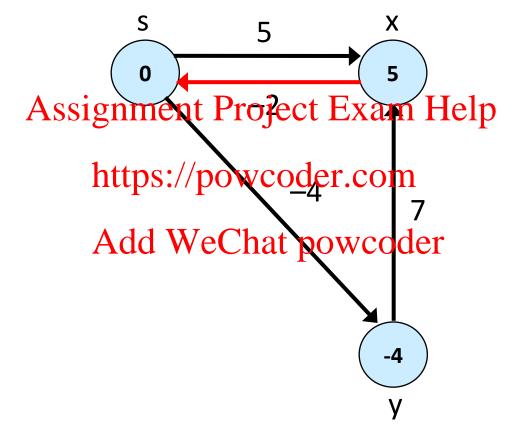
return true
```

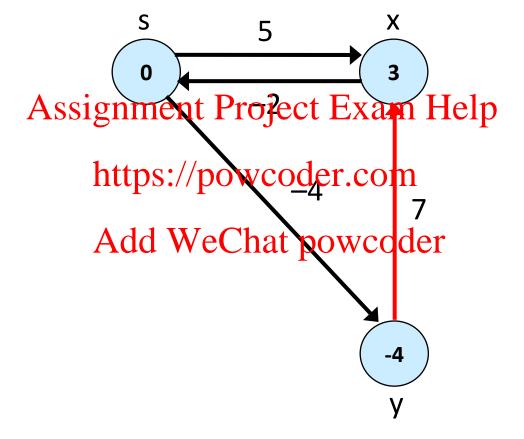
Time Complexity is O(VE).

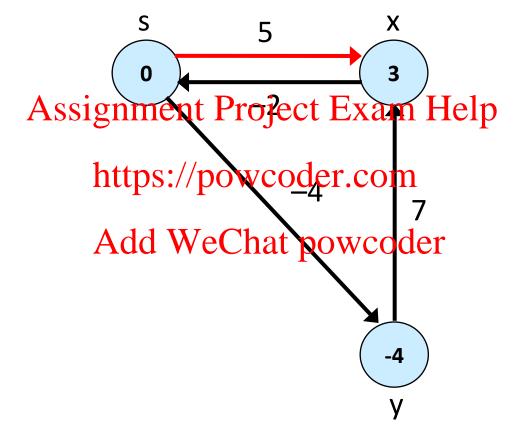


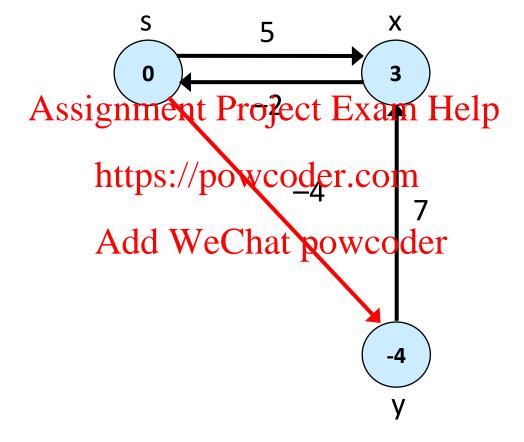


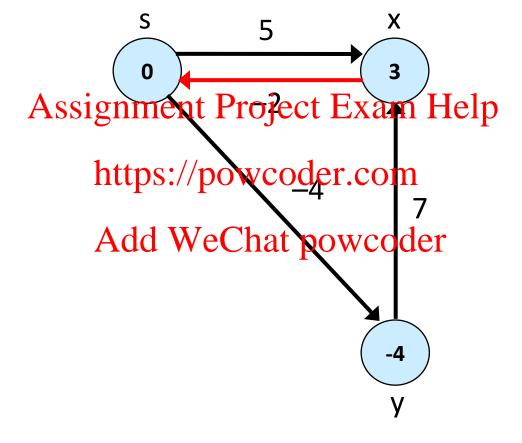


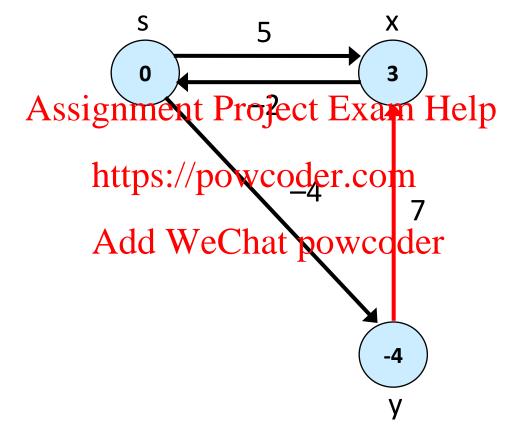




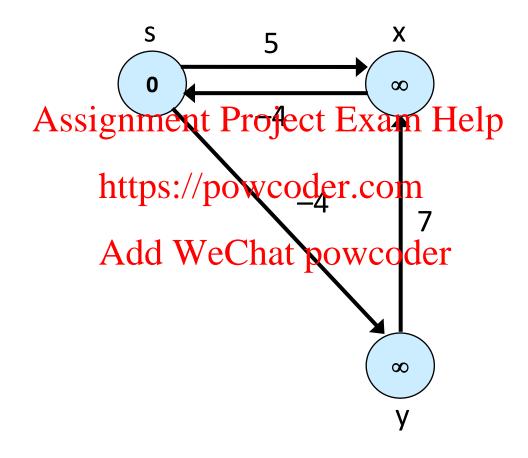


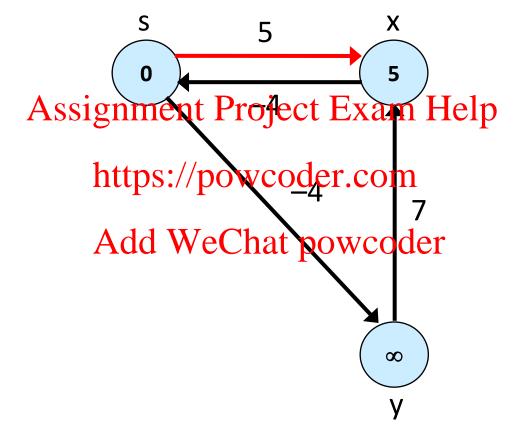


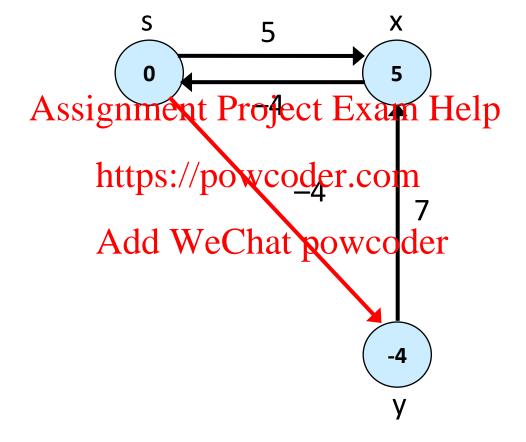


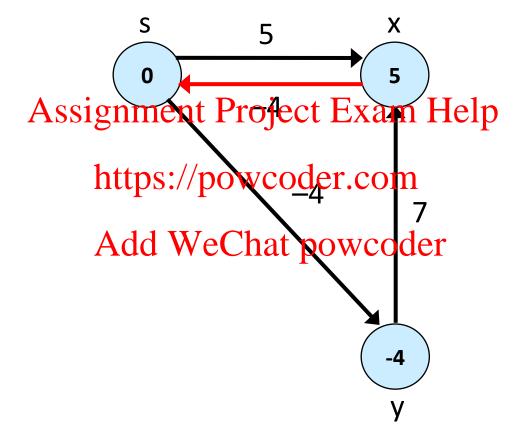


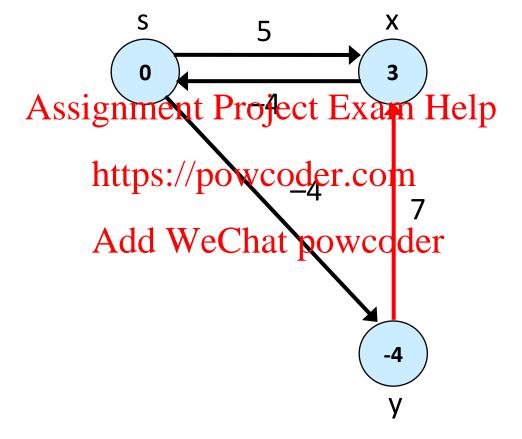
## Assignment Project Exam Help Add wexnample 2 Add wexnample 2

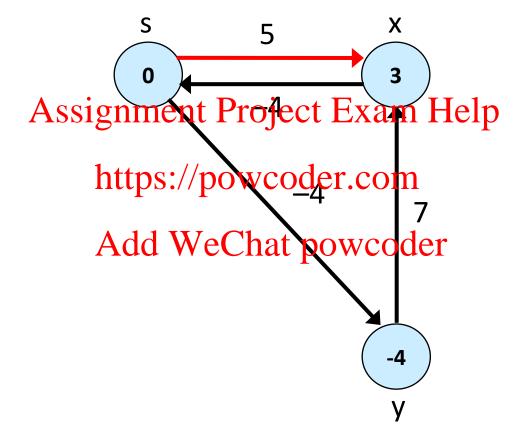


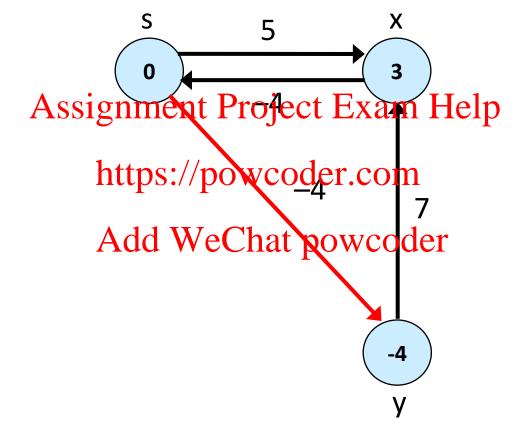


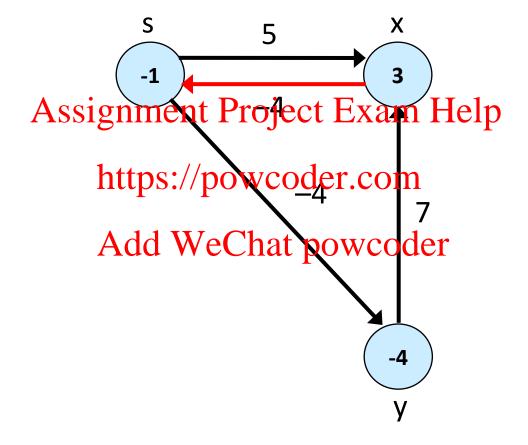


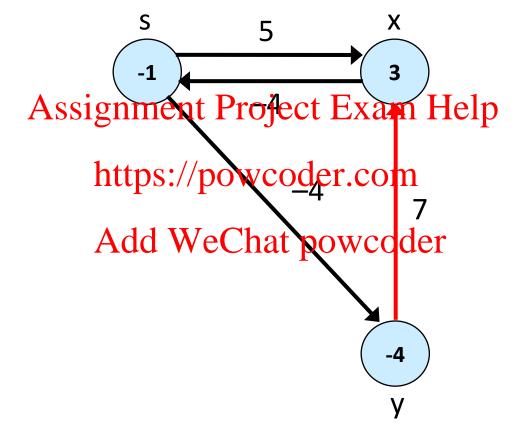




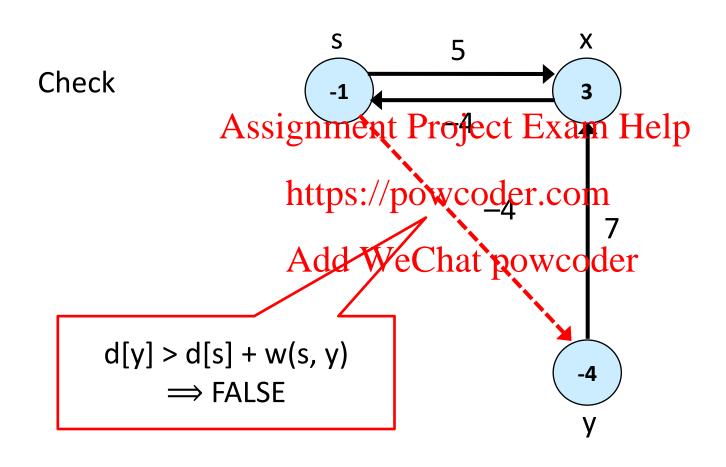








#### Assignment Project Exam Help Add wexnar Power and Add wexnar Power and Add wexnar Power and Add wexnar Add wexnar Power and Add wexnar Power Add wexnar



#### Assignment Project Exam Help

#### Another Look at Bellman-Ford

**Note:** This is essentially **dynamic programming**.

Let d(i, j) = cost of the shortest path from s to i that is at most j hops.

$$d(i,j) = \begin{cases} 0 & \text{Assignment Project Exam Help} & \text{if } i = s \land j = 0 \\ \infty & \text{if } i \neq s \land j = 0 \\ \min(\{d(k,j-h)\text{ tpsi(k,poweraller)}\}\text{ cond } (i,j-1)\}) & \text{if } j > 0 \end{cases}$$

#### Add WeChat powcoder

# Assignment Project Exam Help Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

#### KNAPSACK PROBLEM

## Assignment Project Exam Help Knapsack problem

- Given n objects and a "knapsack."
- Item i weighs  $w_i > 0$  and has value  $v_i > 0$ .
- Knapsack has capacity of W.
- Goal: fill knapsack so as to maximize total value.

  Assignment Project Exam Help

  1

  Ex. {1,2,5} has value 35.

  Ex. {3,4} has value 4ttps://powcoder.com

  3

  Ex. {3,5} has value 46 (but exceeds weight limit).

  Add WeChat powcoder

  5

  6

  7

knapsack instance (weight limit W = 11)

Greedy by value. Repeatedly add item with maximum  $v_i$ . Greedy by weight. Repeatedly add item with minimum  $w_i$ . Greedy by ratio. Repeatedly add item with maximum ratio  $v_i / w_i$ .

Observation. None of greedy algorithms is optimal.

## Assignment Project Exam Help Add Walse Starter

Def.  $OPT(i) = \max \text{ profit subset of items } 1, ..., i$ .

Case 1. OPT does signment Project Exam Help

• *OPT* selects best of { 1, 2, ..., *i* - 1 }.

https://powcoder.com
optimal substructure property
(proof via exchange argument)

Case 2. OPT selects itemd WeChat powcoder

- Selecting item i does not immediately imply that we will have to reject other items.
- Without knowing what other items were selected before i, we don't even know if we have enough room for i.

Conclusion. Need more subproblems!

#### Assignment Project Exam Help

#### Add Wew variable

Def.  $OPT(i, w) = \max \text{ profit subset of items } 1, ..., i \text{ with weight limit } w$ .

Case 1. *OPT* does not select item i.
Assignment Project Exam Help
• *OPT* selects best of  $\{1, 2, ..., i-1\}$  using weight limit w.

https://powcoder.com \ optimal substructure property

- New weight limit = Add. WeChat powcoder (proof via exchange argument)
  - *OPT* selects best of  $\{1, 2, ..., i-1\}$  using this new weight limit.

$$OPT(i, w) = \begin{cases} 0 & \text{if } i = 0 \\ OPT(i-1, w) & \text{if } w_i > w \\ \max \left\{ OPT(i-1, w), v_i + OPT(i-1, w-w_i) \right\} & \text{otherwise} \end{cases}$$

#### Assignment Project Exam Help Dynamicaprogramming algorithm

```
KNAPSACK (n, W, w_1, ..., w_n, v_1, ..., v_n)
```

```
FOR w = 0 TO W Assignment Project Exam Help M[0, w] \leftarrow 0.

https://powcoder.com

FOR i = 1 TO w = 1
```

IF  $(w_i > w)$   $M[i, w] \leftarrow M[i-1, w]$ . ELSE  $M[i, w] \leftarrow \max \{ M[i-1, w], v_i + M[i-1, w-w_i] \}$ .

RETURN M[n, W].

## Assignment Project Exam Help Add Weenat poleter

i	v <sub>i</sub>	<b>w</b> i
Assignmer	nt Project E	Exam Help
Pttps:/	/powcoder	.com <sup>2</sup>
3	18	5
Add V	VeChat pov	vcoder
5	28	7

Max weight W = 11

v <sub>i</sub>	w <sub>i</sub>
1	1
6	2
18	5
22	6
28	7
	1 6 18 22

# Assignment Project Exam Help Add Weenat powelder

Assignment Project Exam Help M 3 6 0 9 10 11 <del>//bowcoder.com</del> {} 0 0 0 0 0 0 0 0 Add WeChat powcoder **{1}** 0 {1,2} {1,2,3} 0 {1,2,3,4} 0 {1,2,3,4,5} 0

W

i	Vi	W: A	Assi
1			
1	1	1	
2	6	2	
3	18	5	
4	22	6	
5	28	7	

# ignment Project Exam Help Add Weenat Project Exam Help Add Weenat Project Exam Help

M	0	1	2 http	3	4	5	6	7	8	9	10	11
{}	0	0	0	0	0	0	<b>f.CO</b> 0	0	0	0	0	0
{1}	0	1	Add	1 W (	eCha 1	at po	WCC 1	oder 1	1	1	1	1
{1,2}	0											
{1,2,3}	0											
{1,2,3,4}	0											
{1,2,3,4,5}	0											

Assign	w <sub>i</sub> 4	v <sub>i</sub>	i
٨	1	1	1
Ac	2	6	2
	5	18	3
	6	22	4
	7	28	5

{1,2}

{1,2,3}

{1,2,3,4}

{1,2,3,4,5}

# gnment Project Exam Help Add Weenat Popleder

Assignment Project Exam Help M https://bowcoder.com {} WeChat powcoder **{1**} 

i	v <sub>i</sub>	w <sub>i</sub> 4	Assignment Project Exam Help
1	1	1	Add weekampleder
2	6	2	Add weenat poweoder
3	18	5	
4	22	6	
5	28	7	

Assignment Project Exam Help M bowcoder.com {} **{1}** M(i-1,w)  $V_2+M(i-1,w-w_2)$  1 {1,2,3} {1,2,3,4} {1,2,3,4,5} 

i	v <sub>i</sub>	w <sub>i</sub> 4	Assignment Project Exam Help
1	1	1	Add weenat populeder
2	6	2	Add Weenat poweoder
3	18	5	
4	22	6	
5	28	7	

Assignment Project Exam Help M <del>bowcoder.com</del> {} eChat powcoder **{1}** M(i-1,w) {1,2}  $V_2+M(i-1,w-w_2)$ {1,2,3} {1,2,3,4} {1,2,3,4,5} 

Assign	w <sub>i</sub> 4	v <sub>i</sub>	i
٨	1	1	1
Ac	2	6	2
	5	18	3
	6	22	4
	7	28	5

# gnment Project Exam Help Add Weenat Project Exam Help

М	0	1	2 httr	3	4	5	6	7	8	9	10	11
{}	0	0	0	0	0	code 0	0	0	0	0	0	0
{1}	0	1	Add	1 W (	eCha 1	at po	WCC 1	oder 1	1	1	1	1
{1,2}	0	1	6	7	7	7	7	7	7	7	7	7
{1,2,3}	0											
{1,2,3,4}	0											
{1,2,3,4,5}	0											

i	v <sub>i</sub>	w <sub>i</sub> 4	Assignment Project Exam Help
1	1	_	Add weenat populeder
2	6	2	Add Weenat poweoder
3	18	5	
4	22	6	
5	28	7	

M	0	1	2	3	4	5	6	7	8	9	10	11
{}	0	0	http 0	0	0	ode 0	0	0	0	0	0	0
{1}	0	1	Add	1 Wo	eCha 1	at po	WCC 1	oder 1	1	1	1	1
{1,2}	0	1	6	7	7	7	7	7	7	7	7	7
{1,2,3}	0	1	6	7	7	18	19	24	25	25	25	25
{1,2,3,4}	0											
{1,2,3,4,5}	0											

i	v <sub>i</sub>	w <sub>i</sub> /	Assignment Project Exam Help
1	1	1	Add weekampleder
2	6	2	Add weenat poweoder
3	18	5	
4	22	6	
5	20	7	

M	0	1	2	3	4	5	6	7	8	9	10	11
{}	0	0	0	0	0 0	0	<b>r.co</b> : 0	0	0	0	0	0
{1}	0	1	Add	1 W (	eCha 1	at po	WCC 1	oder 1	1	1	1	1
{1,2}	0	1	6	7	7	7	7	7	7	7	7	7
{1,2,3}	0	1	6	7	7	18	19	24	25	25	25	25
{1,2,3,4}	0	1	6	7	7	18	22	24	28	29	29	40
{1,2,3,4,5}	0											

i	v <sub>i</sub>	w <sub>i</sub> 4	Assignment Project Exam Help
1	1	1	Add weenat populeder
2	6	2	Add Weenat poweoder
3	18	5	
4	22	6	
5	28	7	

Assignment Project Exam Help M <del>nowcode</del>r {} WeChat po wcoder **{1**} Item 3 in solution {1,2} Item 4 in solution {1,2,3} {1,2,3,4} {1,2,3,4,5} 

#### Assignment Project Exam Help

#### Add Wechar bysisder

Theorem. There exists an algorithm to solve the knapsack problem with nitems and maximum weight W in  $\Theta(n \ W)$  time and  $\Theta(n \ W)$  space.

Pf.

- Takes O(1) time per table entry. Assignment Project Exam Help There are  $\Theta(n|W)$  table entries.
- After computing optimal values wan trace back to find solution: take item i in OPT(i, w) iff M[i, w] < M[i-1, w].

Add WeChat powcoder

#### Remarks.

- Not polynomial in input size! ← "pseudo-polynomial"
- Decision version of knapsack problem is NP-Complete. [ Chapter 8 ]
- There exists a poly-time algorithm that produces a feasible solution that has value within 1% of optimum. [ Section 11.8]