Dynamic Programming

Knapsack

Spring 2022

Knapsack Problem

Given n items of

integer weights: $w_1 \quad w_2 \ \ldots \ w_n$

values: $v_1 \quad v_2 \ \ldots \ v_n$

a knapsack of integer capacity W

find most valuable subset of the items that fit into the knapsack

Recursive solution?

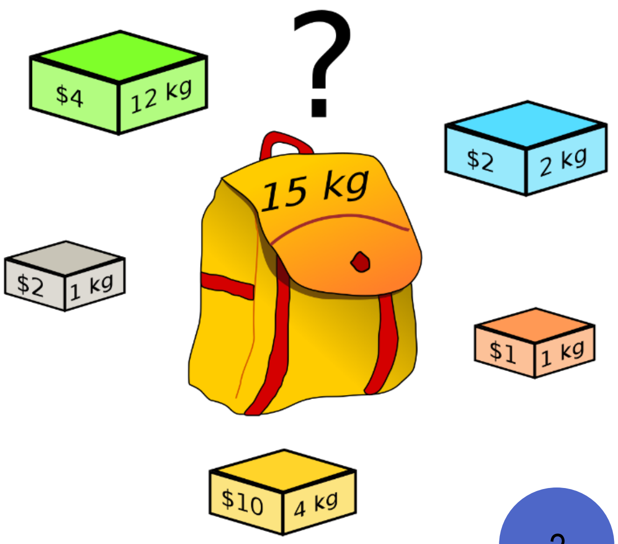
What is smaller problem?

How to use solution to smaller

in solution to larger Table?

Order to solve?

Initial conditions?



Knapsack Problem (Brute Force)

Given n items of

Item x_1 x_2 … x_n

- *Since there are n items , so* 2^n possible combinations.
- We go through all possible combinations and finds the one with the most total value and with total weight less or equal ***W***
- Running time will be 2^n

x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
…			
…			
…			
1	1	1	1

$2^4 = 16$

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
	1	0					
	2	0					
	3	0					
	4	0					

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0		12			
	2	0					
	3	0					
	4	0					

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	**12**	12	12	12
	2	0					
	3	0					
	4	0					

Only Item#1 is picked, having weight=2, value=12

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10				
	3	0					
	4	0					

Item#2 is picked, having weight=1, value=10

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12			
	3	0					
	4	0					

Item#1 is picked, having weight=2, value=12

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22		
	3	0					
	4	0					

Item#1 & 2 both are picked, having weight=2+1, value=12+10

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22	22	22
	3	0					
	4	0					

Item#1 & 2 both are picked, having weight=2+1, value=12+10

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3$, $v_3 = 20$	3	0	10				
	4	0					

Item#2 is picked, having weight=1, value=10

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3$, $v_3 = 20$	3	0	10	12			
	4	0					

Item#1 is picked, having weight=2, value=12

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22		
	4	0					

Item#1 & 2 both having weight=2+1, value=12+10

Item#3 only having weight=3, value=20

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2,\ v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1,\ v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3,\ v_3 = 20$	3	0	10	12	22	30	
	4	0					

Item#2 & 3 both.
having weight=3+1,
value=10+20

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
	4	0					

Item#1 & 3 both.
having weight=2+3,
value=12+20

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3$, $v_3 = 20$	3	0	10	12	22	30	32
	4	0					

Item#1 & 3 both.
having weight=2+3,
value=12+20

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10				

Item#2 is picked, having weight=1, value=10

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10	15			

Item#1 having weight=2, value=12

Item#4 having weight=2, value=15

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3$, $v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2$, $v_4 = 15$	4	0	10	15	25		

Item#1 & 2
weight=2+1,
value=12+10

Item#3
weight=3,
value=20

Item# 2 & 4
weight=1+2,
value=10+15

KNAPSACK PROBLEM BY DP (EXAMPLE)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10	15	25	30	

Item#1 & 4
weight=2+2,
value=12+15

Item# 2 & 3
weight=1+3,
value=10+20

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10	15	25	30	37

Item#1 & 3
weight=2+3,
value=12+20

Item#3 & 4
weight=3+2,
value=20+15

Item# 1 , 2 & 4
weight=2+1+2,
value=12+10+15

Knapsack Problem

Recursive Definition

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases} \quad \textbf{(8.6)}$$

It is convenient to define the initial conditions as follows:

$$F(0, j) = 0 \text{ for } j \geq 0 \quad \text{and} \quad F(i, 0) = 0 \text{ for } i \geq 0. \quad \textbf{(8.7)}$$

Our goal is to find $F(n, W)$, the maximal value of a subset of the n given items that fit into the knapsack of capacity W, and an optimal subset itself.

Among the subsets that do not include the *i*th item, the value of an optimal subset is, by definition, *F(i* − 1*, j)*

Knapsack Problem

Recursive Definition

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases} \quad \textbf{(8.6)}$$

It is convenient to define the initial conditions as follows:

$$F(0, j) = 0 \text{ for } j \geq 0 \quad \text{and} \quad F(i, 0) = 0 \text{ for } i \geq 0. \quad \textbf{(8.7)}$$

Our goal is to find $F(n, W)$, the maximal value of a subset of the n given items that fit into the knapsack of capacity W, and an optimal subset itself.

do include the *i*th item (hence, $j - wi \geq 0$), an optimal subset is made up of this item and an optimal subset of the first $i - 1$ items that fits into the knapsack of capacity $j - wi$. The value of such an optimal subset is $vi + F(i - 1, j - wi)$.

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases}$$

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10				

i=4 j=1

$j - w_i$ **=1-2=-1 <0**

F(3, 1)

Knapsack Problem by DP (Example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases}$$

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10	15			

i=4, j=2

$j - w_i$ **=2-2=0 >=0**

Max { F(3, 2), 15 + F(3, 0) }

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases}$$

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10	15	25		

i=4, j=3

$j - w_i$ **=3-2=1 >=0**

Max { F(3, 3), 15 + F(3, 1) }

Max { 22, 15+10 }

Knapsack Problem by DP (Example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases}$$

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3$, $v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2$, $v_4 = 15$	4	0	10	15	25	30	

i=4, j=4

$j - w_i$ **=4-2=2 >=0**

Max { F(3, 4), 15 + F(3, 2) }

Max { 30, 15+12 }

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

$$F(i, j) = \begin{cases} \max\{F(i-1, j), v_i + F(i-1, j-w_i)\} & \text{if } j - w_i \geq 0, \\ F(i-1, j) & \text{if } j - w_i < 0. \end{cases}$$

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$	3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$	4	0	10	15	25	30	37

i=4, j=5

$j - w_i$ **=5-2=3 >=0**

Max { F(3, 5), 15 + F(3, 3) }

Max { 32, 15+22 }

Knapsack Problem

Recursive Algorithm

ALGORITHM *MFKnapsack*(i, j)

```
//Implements the memory function method for the knapsack problem
//Input: A nonnegative integer i indicating the number of the first
//        items being considered and a nonnegative integer j indicating
//        the knapsack capacity
//Output: The value of an optimal feasible subset of the first i items
//Note: Uses as global variables input arrays Weights[1..n], Values[1..n],
//and table F[0..n, 0..W] whose entries are initialized with −1's except for
//row 0 and column 0 initialized with 0's
if F[i, j] < 0
    if j < Weights[i]
        value ← MFKnapsack(i − 1, j)
    else
        value ← max(MFKnapsack(i − 1, j),
                    Values[i] + MFKnapsack(i − 1, j − Weights[i]))
    F[i, j] ← value
return F[i, j]
```

Knapsack Problem

		0	$j-w_i$	j	W
	0	0	0	0	0
	$i-1$	0	$F(i-1, j-w_i)$	$F(i-1, j)$	
w_i, v_i	i	0		$F(i, j)$	
	n	0			goal

Table for solving the knapsack problem by dynamic programming.

Knapsack Problem by DP (example)

Example: Knapsack of capacity $W = 5$

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2$, $v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1$, $v_2 = 10$	2	0	-	12	22	-	22
$w_3 = 3$, $v_3 = 20$	3	0	-	-	22	-	32
$w_4 = 2$, $v_4 = 15$	4	0	-	-	-	-	37

Knapsack Problem by DP (example)

capacity j

	i	0	1	2	3	4	5
	0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$	1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$	2	0	-	12	22	-	22
$w_3 = 3, v_3 = 20$	3	0	-	-	22	-	32
$w_4 = 2, v_4 = 15$	4	0	-	-	-	-	37

We can find the composition of an optimal subset by backtracking the computations of this entry in the table.
Since ***F(4, 5) > F(3, 5),*** item 4 has to be included in an optimal solution along with an optimal subset for filling **5 – 2 = 3** remaining units of the knapsack capacity.
The value of the latter is *F*(3, 3).
Since ***F(3, 3) = F(2, 3),*** item 3 need not be in an optimal subset.
Since ***F(2, 3) > F(1, 3),*** item 2 is a part of an optimal selection,
which leaves element *F*(1, 3 – 1) to specify its remaining composition.
Similarly,
since ***F(1, 2) > F(0, 2),*** item 1 is the final part of the optimal solution

{item 1, item 2, item 4}

Knapsack Problem

The time efficiency and space efficiency of this algorithm are both in $\Theta(nW)$.

The time needed to find the composition of an optimal solution is in $O(n)$.

KNAPSACK PROBLEM

- Chapter #8 (Dynamic Programming)
- Knapsack Problem (8.2)
- Page # 292 – 297
- Introduction to the Design and Analysis of Algorithms 3rd Edition,

by Anany Levitin