

Knapsack and rounding



Knapsack special special case

**All items have size = value
 $\in \{1, 2, \dots, B\}$
and in addition
 B is a "small" integer**

Dynamic programming

add
stuff
here



interface

Given partial solution
for first i items,

what to remember

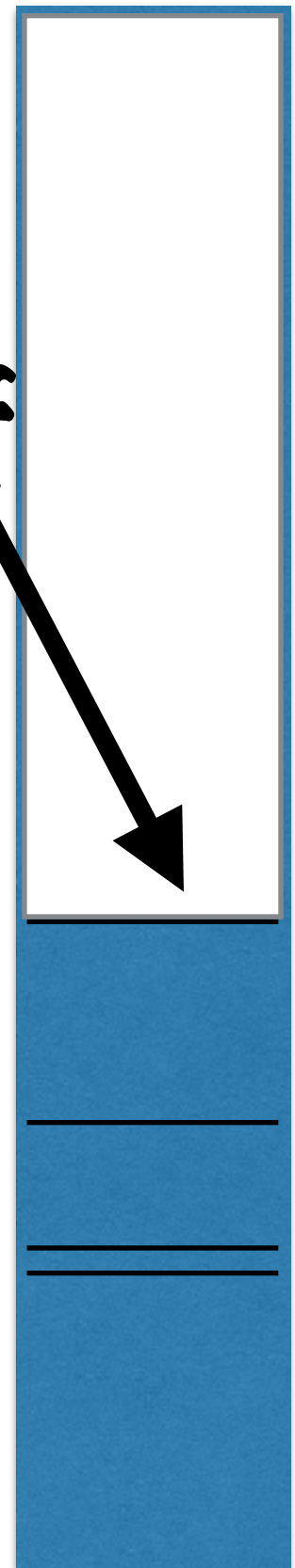
to complete solution optimally?

Dynamic programming

Q: What to remember?

**A: remember
total value so far**

add
stuff
here



**$A[i, v]$ = whether
 v achievable with
subset of first i items**

Q: v achievable with
subset of first i items iff...

A: ...it depends on
whether subset contains i

If not:

v must be reached
with first $i-1$ items

If yes:

$v - v_i$ must be reached
with first $i-1$ items

**$A[i, v]$ = whether
 v achievable with
subset of first i items**

$$A[i, v] = A[i - 1, v] \text{ or } ((v \geq v_i) \text{ and } A[i - 1, v - v_i])$$

Algorithm

For $v = 0 \dots B : A[1, v] \leftarrow \text{false}$

$A[1, v_1] \leftarrow \text{true}, A[1, 0] \leftarrow \text{true}$

For $i = 2 \dots n,$

For $v = 0 \dots v_i - 1 : A[i, v] \leftarrow A[i - 1, v]$

For $v = v_i \dots B :$

$A[i, v] \leftarrow A[i - 1, v] \text{ or } A[i - 1, v - v_i]$

Output $\max\{v : A[n, v] \text{ is true} \}$

Knapsack and rounding

