Section 3.1: The Knapsack Problem

Knapsack

Input:

Knapsack with a capacity $B \in \mathbb{Z}^+$

Items $I = \{1, 2, \dots, n\}$

Item i has size $s_i \in \mathbb{Z}^+$ and value $v_i \in \mathbb{Z}^+$

Objective:

Find a set of items with total size $\leq B$ and largest possible total value

Greedy alg.:

Consider items in order of decreasing v/s-ratio

Does not have any constant approx. ratio:

Ex:

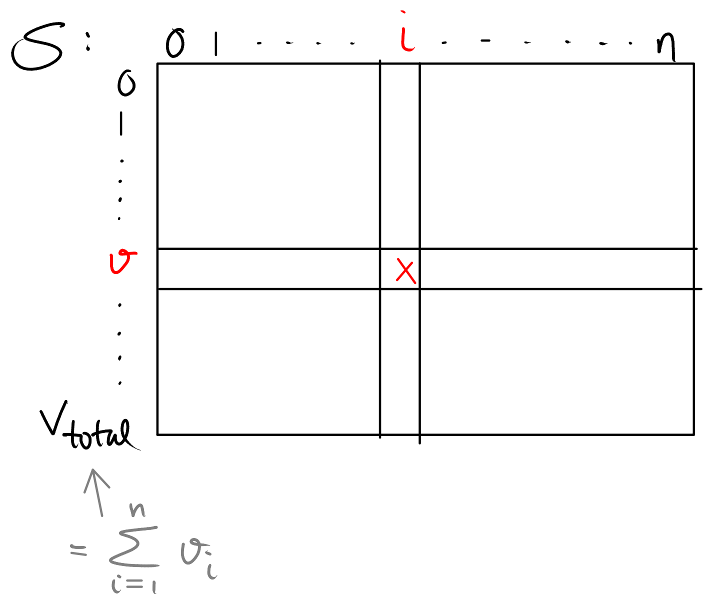
$s_1 = v_1 = 1 \longrightarrow v_1/s_1 = 1$

$s_2 = B, \quad v_2 = B-1 \longrightarrow v_2/s_2 = 1 - \frac{1}{B}$

Greedy $= 1$

OPT $= B-1$

Dynamic prg alg: (Alg. 3.1)



$S_{v,i}$: smallest possible total size of subset of items $1, \ldots, i$ with total value v.

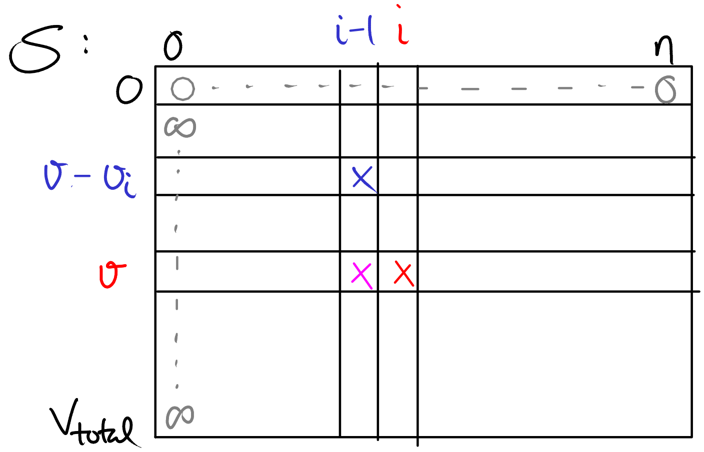
Ex: $B = 5$

	1	2	3
size	2	4	2
value	3	2	1

	0	1	2	3	
0	0	0	0	0	
1	∞	∞	∞	2	
2	∞	∞	4	4	
3	∞	2	2	2	
4	∞	∞	∞	4	$\leq B$
5	∞	∞	6	6	$> B$
6	∞	∞	∞	8	$> B$

largest possible total value $\rightarrow$ 4

What are the rules for filling the table?



$S_{v,i}$: smallest possible total size of subset of items $1, \dots, i$ with total value v.

If $i=0$ and $1 \le v \le V_{total}$

$S_{v,i} = \infty$

Otherwise,

$$S_{v,i} = \begin{cases} 0, & \text{if } v = 0 \\ S_{v,i-1}, & \text{if } 0 < v < v_i \\ \min\{\underbrace{S_{v,i-1}}_{\text{best solution \underline{without} item } i},\ \underbrace{S_{v-v_i,i-1} + s_i}_{\text{best solution \underline{with} item } i}\}, & \text{if } v \ge v_i \end{cases}$$

How do we determine which items to select to obtain the optimal value?

$i-1 \quad i$

$\nwarrow$ include item i in the solution

$i-1 \quad i$

$\leftarrow$ leave out item i

Analysis of the alg:

Running time $O(nV_{total})$

Input size $O(\log B + n(\log M + \log S))$, where

$$M = \max_{1 \le i \le n} \{v_i\} \quad \text{and} \quad S = \max_{1 \le i \le n} \{s_i\}$$

Thus, the running time is not poly., since there could be instances with, e.g.,

$$V_{total} = 2^n \quad \text{and}$$
$$\log M + \log S = n$$

But if the numeric part of the input (i.e., the capacity, the item sizes, and the item values) were written in unary, the input size would be $\Theta(B + V_{total} + S_{total})$, and the running time would be poly. in the input size.
Hence, the running time is pseudo-polynomial.

Note: if V_{total} is poly. in n for all instances, the dyn prg. alg. is poly.
Leading to the following idea.

$\boxed{\text{Idea for approx. alg.}}$: Round values s.t. there are only a poly. number of different (equidistant) values:

- Choose a value μ.
- Round down each item value to the nearest multiple of μ.
- Do dyn. prg. on the rounded values

How to choose μ?

- When rounding, each value looses less than μ. Hence, the value of any solution is changed by less than $n\mu$.
- $OPT \geq M = \max_{1 \leq i \leq n} \{v_i\}$.

Thus, if we want a precision of ε, $\mu = \frac{\varepsilon M}{n}$ will do, since then

$$n\mu = \varepsilon M \leq \varepsilon \cdot OPT$$

(We will add more detail to this argument in the proof of Thm 3.5)

Since each rounded value is a multiple of μ, we may as well scale each value by a factor of $1/\mu$:

Alg 3.2

$M \leftarrow \max_{1 \le i \le n} v_i$

$\mu \leftarrow \frac{\varepsilon M}{n}$

for $i \leftarrow 1$ to n

$\quad v_i' \leftarrow \lfloor \frac{v_i}{\mu} \rfloor$

Do dyn. prg. with values v_i' (and sizes s_i)

Theorem 3.5

Alg. 3.2 is a $(1-\varepsilon)$-approx. alg. with a running time poly. in both input size and ε

<u>Proof</u>:

Approximation ratio:

For each item i, $\mu v_i'$ equals v_i rounded down to the nearest multiple of μ.

Let S be the set of items selected by Alg. 3.2. This is an optimal solution for the instance with values v_i' and hence also for the instance with values $\mu v_i'$.

Let O be the set of items in an optimal solution to the original instance with values v_i.

The total value produced by Alg. 3.2 is

$$\sum_{i \in S} v_i \geq \sum_{i \in S} \mu \underbrace{v_i'}_{= v_i \text{ rounded down to nearest multiple of } \mu}$$

$$\geq \sum_{i \in O} \mu v_i', \quad \text{since } S \text{ is optimal for rounded values}$$

$$> \sum_{i \in O} (\mu v_i - \mu), \quad \text{since each item loses less than } \mu \text{ in the rounding}$$

$$\geq \sum_{i \in O} \mu v_i - n\mu, \quad \text{since } |O| \leq n$$

$$= OPT - M\varepsilon$$

$$\geq (1-\varepsilon)\, OPT, \quad \text{since } OPT \geq M$$

Running time:

$$\# \text{values} = n \left\lfloor \frac{M}{\mu} \right\rfloor = n \left\lfloor \frac{n}{\varepsilon} \right\rfloor \leq n^2 \frac{1}{\varepsilon}$$

$\Downarrow$

$$\# \text{ table entries} \leq n^3 \cdot \frac{1}{\varepsilon}$$

$\Downarrow$

Running time $O(n^3 \cdot \frac{1}{\varepsilon})$ $\square$

According to Thm 3.5, Alg. 3.2 is a

fully polynomial time approximation scheme (FPTAS)

also poly. in $1/\varepsilon$

poly. in input size

Family $\{A_\varepsilon\}$ of alg., where A_ε has precision ε.

($(1-\varepsilon)$-approx. alg for max. problems,
$(1+\varepsilon)$-approx. alg for min. problems)

Def. 3.4

Def. 3.3

Thus, Thm 3.5 could also be stated like this:

$\boxed{\text{Theorem 3.5: Alg 3.2 is a FPTAS}}$

In the Multiple Knapsack problem, there are a fixed number of knapsacks.

Bin Packing can be seen as a dual problem of Multiple Knapsack:

In the $\boxed{\text{Bin Packing}}$ problem, there is an unlimited supply of bins, all of size 1. The aim is to pack all items in as few bins as possible.

Simple approx. alg.s:	Asymptotic approx. ratio
Next-Fit (NF)	2
First-Fit (FF)	1.7
Best-Fit (BF)	1.7
Next-Fit-Decreasing (NFD)	≈ 1.69
First-Fit-Decreasing (FFD)	1.222...
Best-Fit-Decreasing (BFD)	1.222...

Approx. scheme?

Can we do the same kind of rounding for Bin Packing as we did for Knapsack?