Ex. 3.1:

New variant of Greedy:

Sort items s.t $v_1/s_1 \geq v_2/s_2 \geq \dots \geq v_n/s_n$

Choose k s.t. $\sum_{i=1}^{k} s_i \leq B$, but $\sum_{i=1}^{k+1} s_i > B$

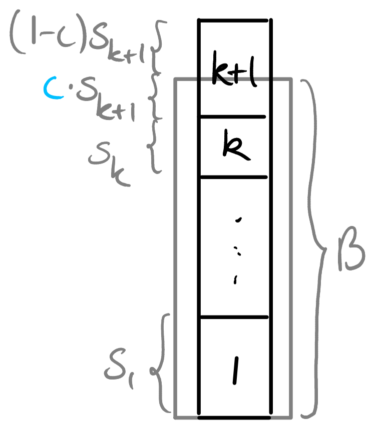
Choose i^* s.t. $v_{i^*} = \max_{1 \leq i \leq n} v_i$

If $\sum_{i=1}^{k} v_i > v_{i^*}$

  Return $\{1, \dots, k\}$

Else

  Return $\{i^*\}$



Since no solution has more value per size unit than the „solution" containing items $1, \dots, k$ and a c-fraction of item $k+1$:

$$
\begin{aligned}
OPT &\leq \sum_{i=1}^{k} v_i + c \cdot v_{k+1} \\
&< \sum_{i=1}^{k} v_i + v_{k+1}, \quad \text{since } c < 1 \\
&\leq \sum_{i=1}^{k} v_i + v_{i^*}, \quad \text{since } v_{i^*} \geq v_{k+1}
\end{aligned}
$$

$$
\max\left\{ \sum_{i=1}^{k} v_i, \; v_{i^*} \right\} \geq \frac{1}{2}\left(\sum_{i=k}^{k} v_i + v_{i^*} \right) \geq \frac{1}{2} \cdot OPT
$$

Section 3.3: Bin Packing

Bin Packing

Input: n items with sizes between 0 and 1.
Objective: Pack items in bins of size 1, using as few bins as possible.

Last time we discussed simple approximation algorithms.
Today we will develop an approximation scheme:

$A_\varepsilon(I)$

Split input I into
- I_s : items smaller than $\varepsilon/2$ (small items)
- I_ℓ : remaining items (large items)

1. Pack large items:
   a. Round up item sizes ($I_\ell \to I_\ell'$)
      $\Rightarrow O\left(\frac{1}{\varepsilon^2}\right)$ different sizes
   b. Do dyn. prg. on I_ℓ'
      $\Rightarrow A_\varepsilon(I_\ell') = OPT(I_\ell')$

2. Add small items to the packing using First-Fit (or any other Any-Fit alg.)

The rounding scheme (1.a.) will be described later.

Adding small items to the packing (2.)

Lemma 3.10

$$A_\varepsilon(I) \leq \max\left\{ A_\varepsilon(I_\ell),\ \underbrace{\frac{2}{2-\varepsilon}}_{\leq 1+\varepsilon,\ \text{for } \varepsilon\leq 1} \cdot \underbrace{\mathrm{size}(I)}_{\leq OPT(I)} + 1 \right\}$$

Proof:

If no extra bin is needed for adding the small items, $A_\varepsilon(I) = A_\varepsilon(I_\ell)$.

Otherwise, all bins, except possibly the last one, are filled to more than $1-\varepsilon/2$.
In this case,

$$A_\varepsilon(I) \leq \left\lceil \frac{\mathrm{size}(I)}{1-\varepsilon/2} \right\rceil < \frac{\mathrm{size}(I)}{1-\varepsilon/2} + 1$$

$$= \frac{2}{2-\varepsilon}\,\mathrm{size}(I) + 1$$

□

Thus, we just need to ensure that
$A_\varepsilon(I_\ell) \leq (1+\varepsilon)\,OPT$.

Rounding scheme (1.a.)

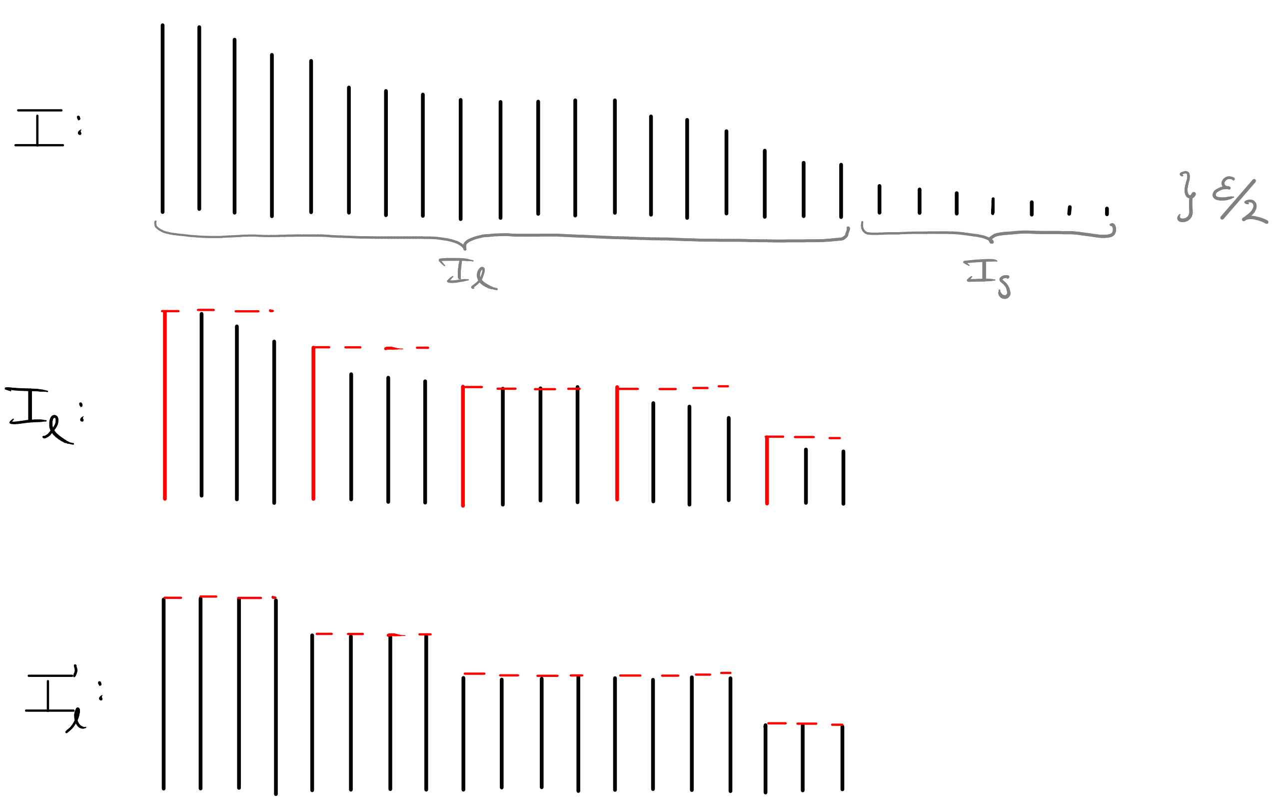
Last time we saw that a rounding scheme similar to the one we used for Knapsack would at best yield an approx. factor of 1.5. Instead, we will use:

Linear grouping:

- Sort items in I_ℓ by decreasing sizes.
- Partition sorted I_ℓ in groups of k consecutive items. (k will be determined later.)
- For each group, round up item sizes to largest size in the group.

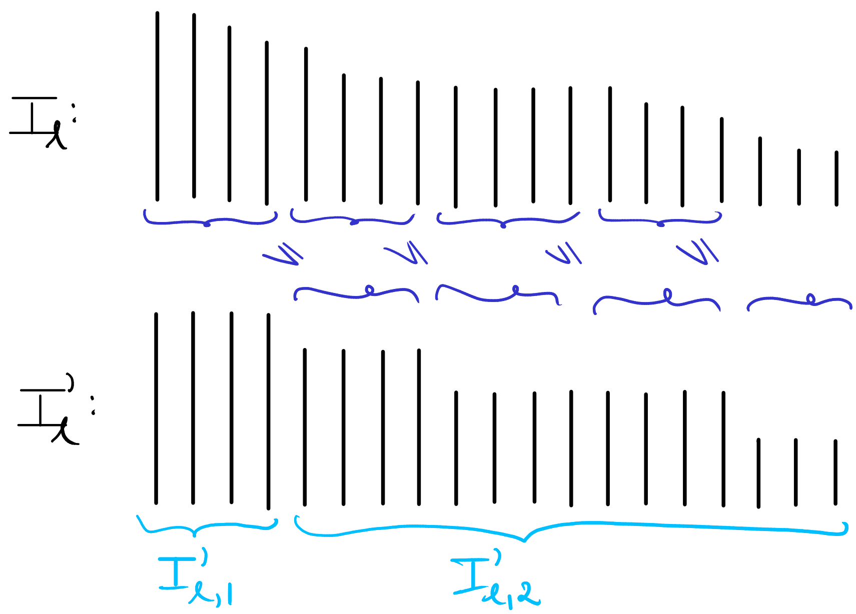
The result is called I'_ℓ.

Ex: $(k=4)$



Approximation

Each item in the i'th group of I_ℓ is at least as large as any item in the $(i+1)$st group of I'_ℓ:



$\Longrightarrow \mathrm{OPT}(I'_{\ell,2}) \leq \mathrm{OPT}(I_\ell)$

$$\mathrm{OPT}(I'_\ell) \leq \underbrace{\mathrm{OPT}(I'_{\ell,1})}_{\leq k,} + \underbrace{\mathrm{OPT}(I'_{\ell,2})}_{\leq \mathrm{OPT}(I_\ell)}$$

since $|I'_{\ell,1}| = k$

This proves:

$\boxed{\textbf{Lemma 3.11: } \mathrm{OPT}(I'_\ell) \leq \mathrm{OPT}(I_\ell) + k}$

$$\underbrace{\mathrm{OPT}(I'_\ell)}_{= A_\varepsilon(I'_\ell)}$$

Thus, letting $k = \lfloor \varepsilon \cdot \text{size}(I) \rfloor \overset{(*)}{\leq} \varepsilon \cdot OPT(I)$ will ensure that

$$
\begin{aligned}
A_\varepsilon(I_\ell) &= A_\varepsilon(I_\ell') \\
&= OPT(I_\ell') \\
&\leq OPT(I_\ell) + k, && \text{by Lemma 3.11} \\
&\leq OPT(I_\ell) + \varepsilon \cdot OPT(I), && \text{by } (*) \\
&\leq (1+\varepsilon) \cdot OPT(I), && \text{since } I_\ell \subseteq I
\end{aligned}
$$

Now, by Lemma 3.10,

$$A_\varepsilon(I) \leq \max\{ \underbrace{A_\varepsilon(I_\ell)}_{\substack{\leq (1+\varepsilon) OPT(I), \\ \text{as just shown}}}, \underbrace{\tfrac{2}{2-\varepsilon} \cdot \text{size}(I) + 1}_{\leq (1+\varepsilon) OPT + 1 \,:} \}$$

$$
\begin{aligned}
\frac{2}{2-\varepsilon} \leq 1+\varepsilon &\iff \\
2 \leq (1+\varepsilon)(2-\varepsilon) &\iff \\
2 \leq 2 + \varepsilon - \varepsilon^2 &\iff \\
\varepsilon \leq 1
\end{aligned}
$$

Thus, $A_\varepsilon(I) \leq (1+\varepsilon) \cdot OPT(I) + 1$

asymptotic approximation scheme

Packing I_ε' using dynamic programming (1.b.)

At most $2/\varepsilon$ items fit into one bin,
since all items in I_ε have size at least $\varepsilon/2$.

There are $N \leq \lceil n/k \rceil$ different sizes $s_1, \ldots, s_N$ in I_ε'.

Hence, any packing of a bin can be represented by a vector $(m_1, \ldots, m_N)$, where m_i, $1 \leq i \leq N$, is the #items of size s_i in the bin and $0 \leq m_i \leq 2/\varepsilon$.
A vector representing the contents of a bin is called a configuration.
Let $\mathcal{C}$ denote the set of possible bin configurations.
Note that $|\mathcal{C}| < \left(\frac{2}{\varepsilon}\right)^N$

Let n_i be the #items of size s_i in I_ε'

For the dyn. prg. we use an N-dimensional table B with $n_i + 1$ rows in the i'th dimension.
$B[m_1, \ldots, m_N]$ will be the minimum #bins required to pack m_i items of size s_i, $1 \leq i \leq N$.

Ex:

$I = \langle 0.6, 0.5, 0.5, 0.4, 0.4, 0.4, 0.3, \underbrace{0.1, 0.1}_{< \varepsilon/2} \rangle$

$\varepsilon = 0.4, \quad k = 4$

$I_\ell = \langle 0.6, 0.5, 0.5, 0.4 \mid 0.4, 0.4, 0.3 \rangle$

$I_\ell' = \langle 0.6, 0.6, 0.6, 0.6, 0.4, 0.4, 0.4 \rangle$

$s_1 = 0.6 \qquad s_2 = 0.4$

$n_1 = 4 \qquad n_2 = 3$

$\mathcal{C} = \{ (0,1), (0,2), (1,0), (1,1) \}$

0.6 \ 0.4	0	1	2	3
0	0	1		
1	1	1		
2	2	2	≤ 3	≤ 3
3	3	≤ 3	≤ 3	
4	4			

0.6 \ 0.4	0	1	2	3
0	0	1	1	
1	1	1	2	
2	2	2	2	
3	3	3	3	
4	4	4		

0.6 \ 0.4	0	1	2	3
0	0	1	1	2
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3
4	4	4	4	4

$$B[4,3] = 1 + \min_{(m_1,m_2)\in\mathcal{C}} \{ B[4-m_1, 3-m_2] \}$$

$$= 1 + \min\{ B[4,2], B[4,1], B[3,3], B[3,2] \}$$

$$= 1 + B[3,2] = 4$$

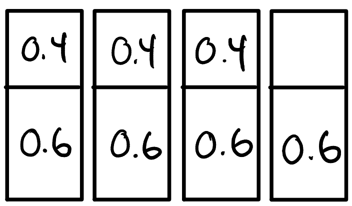
In general:

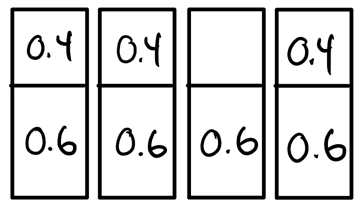
$$B[m_1, \dots, m_N] = 1 + \min_{(c_1,\dots,c_N)\in\mathcal{C}} \{ m_1 - c_1, \dots, m_N - c_N \}$$

0.6 \ 0.4	0	1	2	3
0	0	1	1	2
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3
4	4	4	4	4

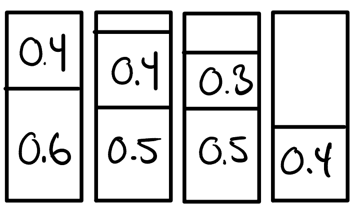
0.6 \ 0.4	0	1	2	3
0	0	1	1	2
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3
4	4	4	4	4

I_e':

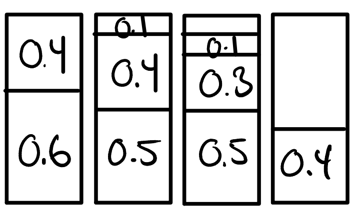




I_e:



I:



Running time

Let $n_\ell = |I_\ell|$. Then,

$$\text{size}(I) \geq \text{size}(I_\ell) \geq n_\ell \cdot \frac{\varepsilon}{2}, \qquad (*)$$

since I_ℓ contains only large items.

Thus,

$$k = \lfloor \varepsilon \cdot \text{size}(I) \rfloor \underset{\text{by } (*)}{\geq} \lfloor n_\ell \cdot \frac{\varepsilon^2}{2} \rfloor \geq n_\ell \cdot \frac{\varepsilon^2}{4} \qquad (**)$$

Hence,

$$N \leq \left\lceil \frac{n_\ell}{k} \right\rceil \underset{\text{by } (**)}{\leq} \left\lceil \frac{4}{\varepsilon^2} \right\rceil \qquad (***)$$

Table size $\leq n_\ell^N \leq n^N$

Time per entry $O(|\mathcal{C}|) \leq O((2/\varepsilon)^N)$

Running time $O(n^N \cdot (\frac{2}{\varepsilon})^N) = O((\frac{2n}{\varepsilon})^N) \underset{\text{by } (***)}{\leq} O((\frac{2n}{\varepsilon})^{\lceil \frac{4}{\varepsilon^2} \rceil})$

not ~~fully~~ poly. time

Hence, $\{A_\varepsilon\}$ is an Asymptotic Poly. Time Approx. Scheme (APTAS)

This proves:

$\boxed{\text{Thm 3.12: There is an APTAS for Bin Packing}}$

There is no PTAS for Bin Packing:

Theorem 3.8

No alg. for Bin Packing has an absolute approx. ratio better than $3/2$, unless $P = NP$

Proof:

Reduction from Partition Problem:

Given a set S of integers,
can S be partitioned into two sets S_1 and S_2,
such that $\sum_{s \in S_1} s = \sum_{s \in S_2} s$?

For a given instance S of the partition problem, let
$B = \sum_{s \in S} s$ and $I = \{ s \cdot \frac{2}{B} \mid s \in S \}$.

Then, $\sum_{i \in I} i = B \cdot \frac{2}{B} = 2$

If we use I as input for the bin packing problem,

- at least 2 bins are needed, and
- 2 bins suffice, iff S is a yes-instance for the partition problem.

If we had a bin packing alg. with an approx. factor $< 3/2$, it would always use only 2 bins, whenever 2 bins suffice

Thus, the alg. could be used to decide any instance of the partition problem. $\square$