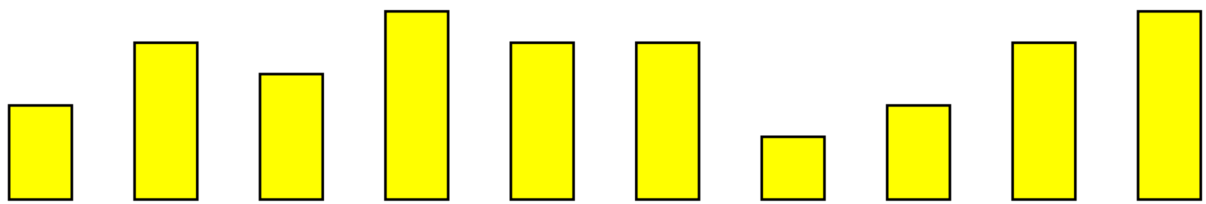
Algoritmos de Aproximação

Segundo Semestre de 2012

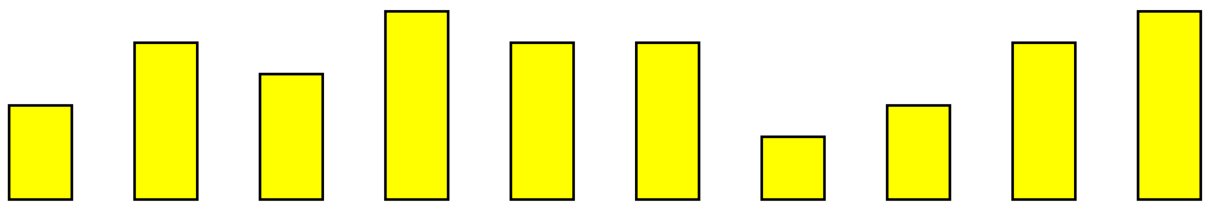
Bin Packing

Dados: n itens $\quad$ ($[n] = \{1, \ldots, n\}$)
comprimento $a[i]$ do item i $\quad$ ($i = 1, \ldots, n$)



Bin Packing

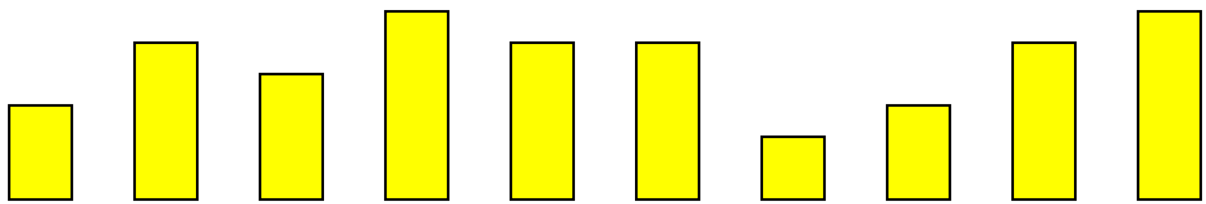
Dados: n itens   ($[n] = \{1, \ldots, n\}$)
comprimento $a[i]$ do item i ($i = 1, \ldots, n$)



Um empacotamento é uma partição $\{B[1], \ldots, B[k]\}$ de $[n]$ tal que $\sum_{j \in B[i]} a_j \leq 1$, para cada i em $[k]$.

Bin Packing

Dados: n itens   ($[n] = \{1, \ldots, n\}$)
comprimento $a[i]$ do item i ($i = 1, \ldots, n$)

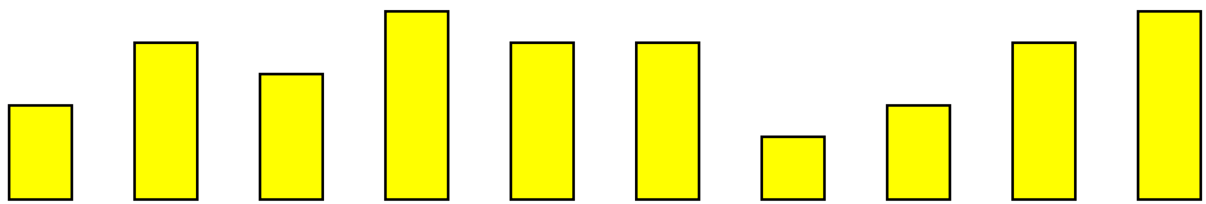


Um empacotamento é uma partição $\{B[1], \ldots, B[k]\}$ de $[n]$ tal que $\sum_{j \in B[i]} a_j \leq 1$, para cada i em $[k]$.

k é o número de bins usados no empacotamento.

Bin Packing

Dados: n itens   ($[n] = \{1, \ldots, n\}$)
comprimento $a[i]$ do item i ($i = 1, \ldots, n$)



Um empacotamento é uma partição $\{B[1], \ldots, B[k]\}$ de $[n]$ tal que $\sum_{j \in B[i]} a_j \leq 1$, para cada i em $[k]$.

k é o número de bins usados no empacotamento.

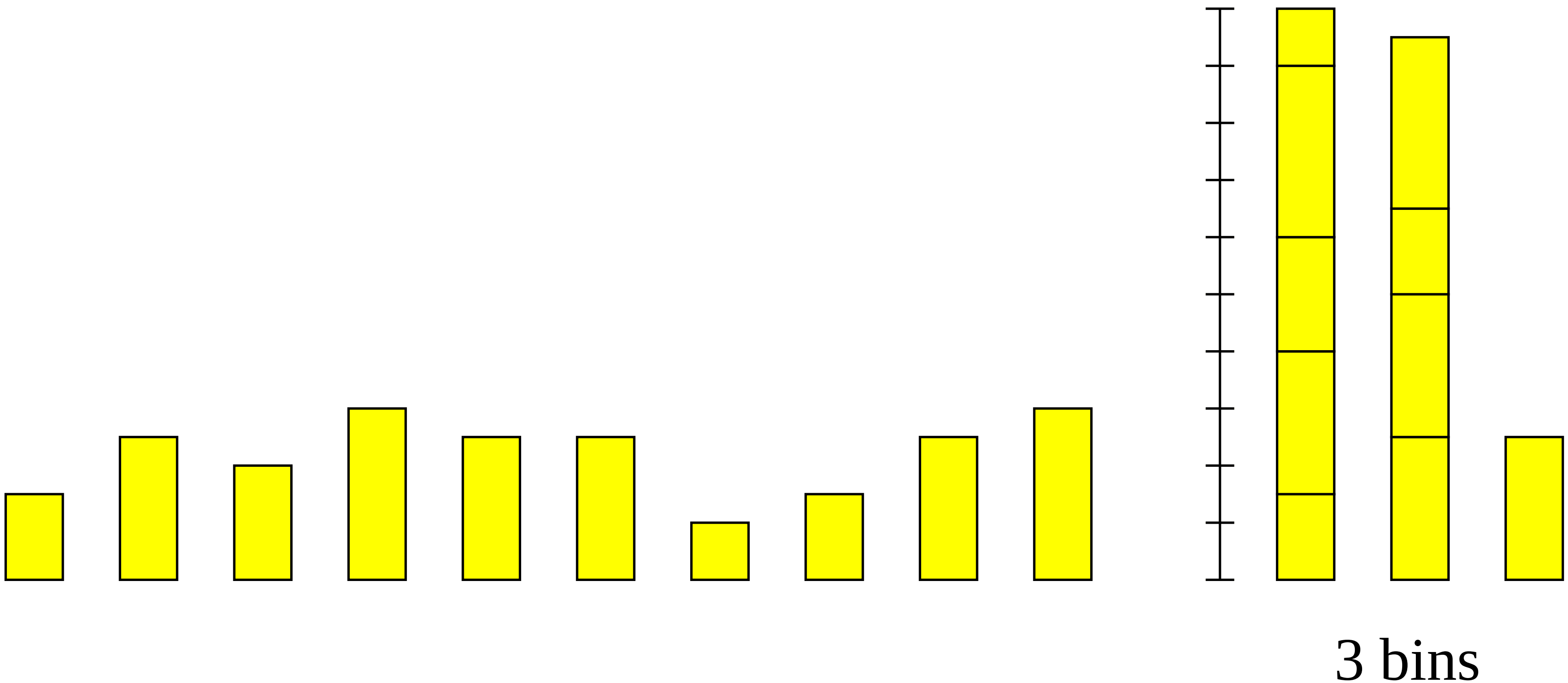
Encontrar empacotamento no menor número de bins.

Bin Packing

Problema: Dados n e $a[1 \,.\,.\, n]$, encontrar empacotamento com o menor número de bins.

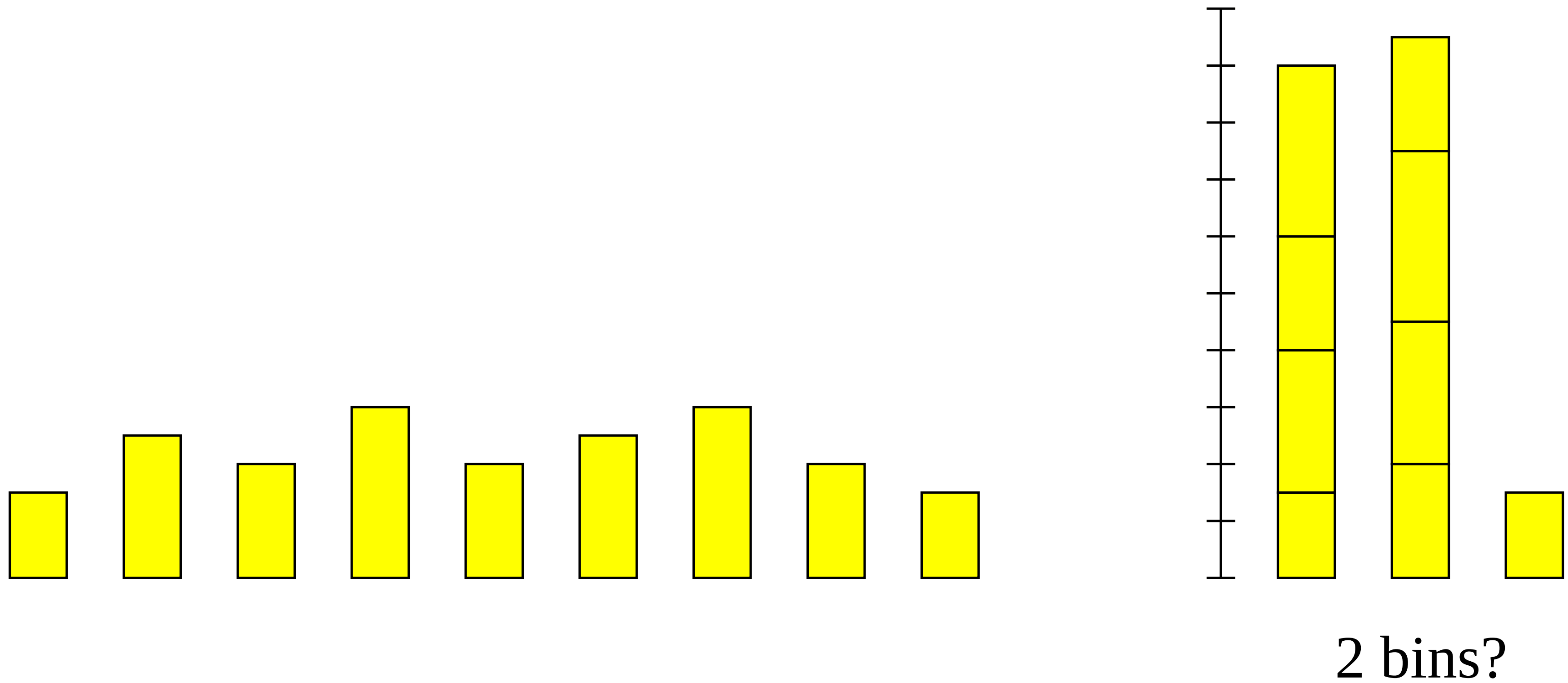
Bin Packing

Problema: Dados n e $a[1 \,.\,.\, n]$, encontrar empacotamento com o menor número de bins.



Bin Packing

Problema: Dados n e $a[1\,..\,n]$, encontrar empacotamento com o menor número de bins.



Teorema: A menos que P $=$ NP, não existe α-aproximação para o bin packing com $\alpha < 3/2$.

Redução do problema da partição.

Algoritmo FFD

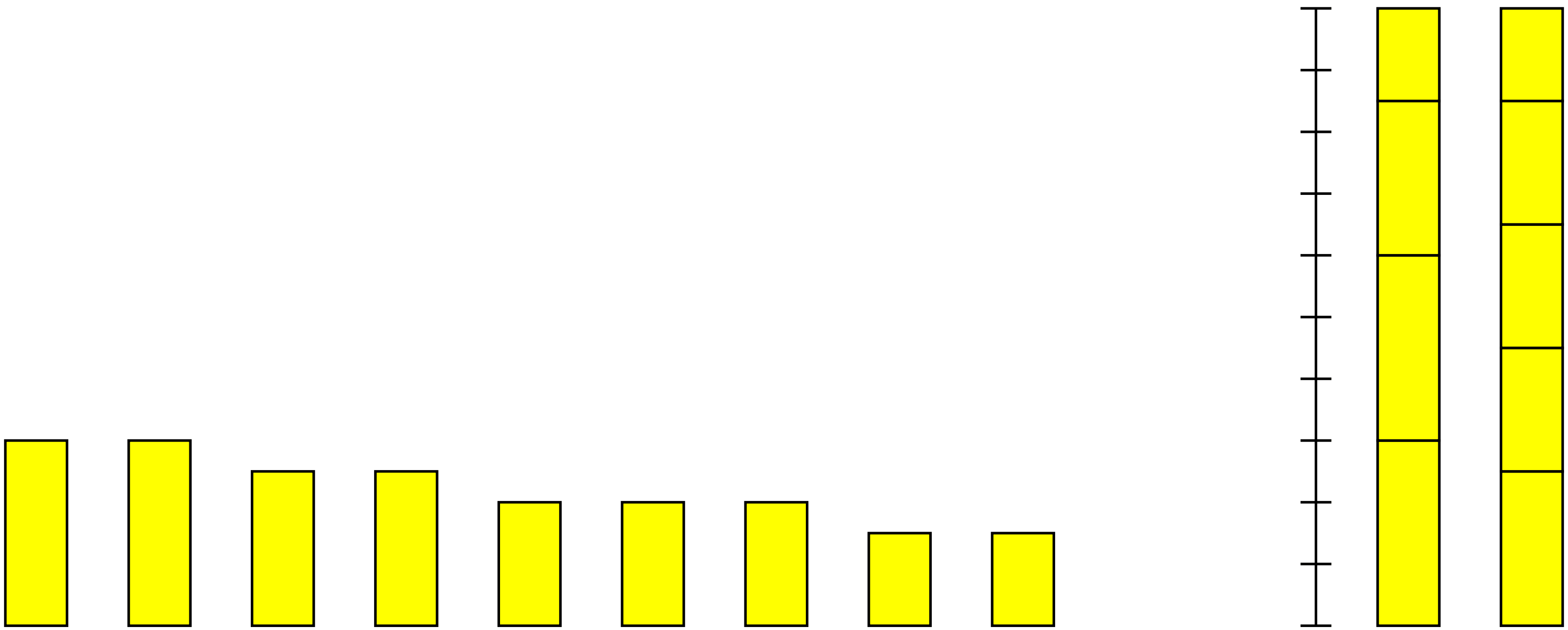
FFD: first fit decreasing

Ordene os itens por tamanho.
Coloque cada item no primeiro bin onde ele cabe.

Algoritmo FFD

FFD: first fit decreasing

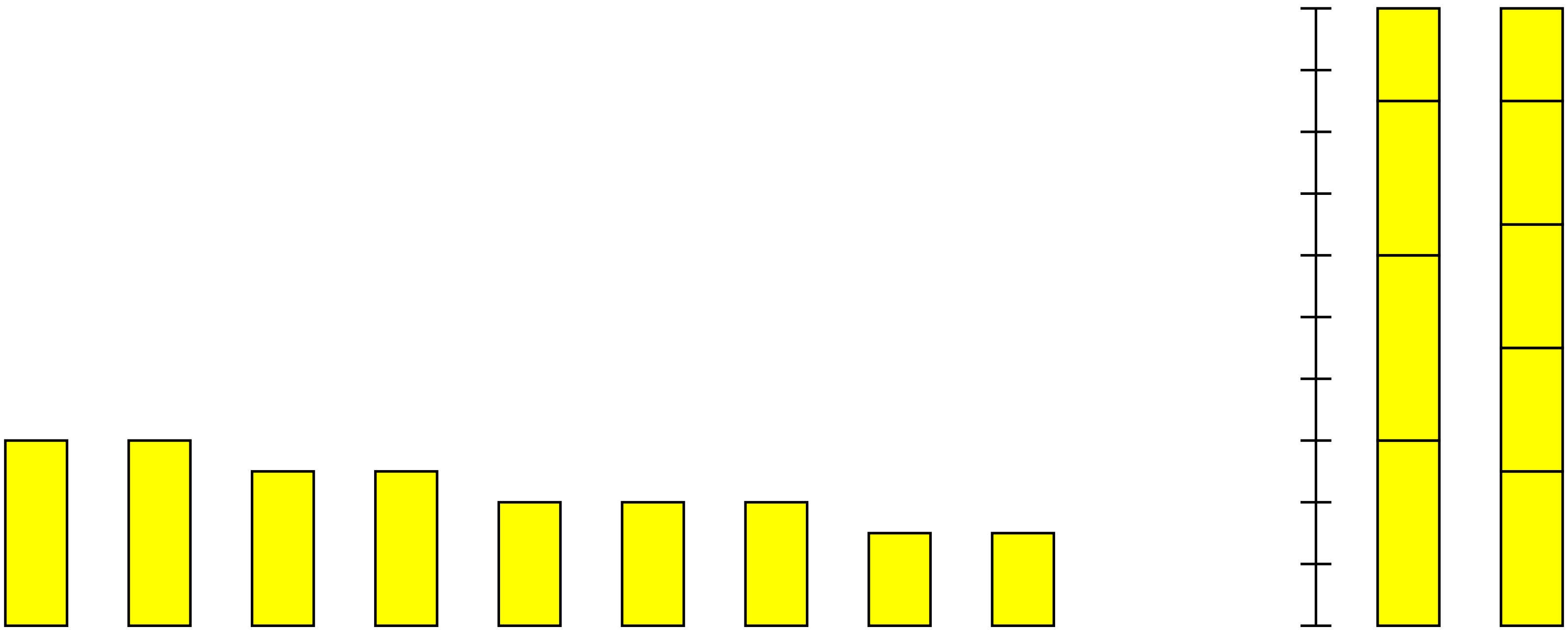
Ordene os itens por tamanho.
Coloque cada item no primeiro bin onde ele cabe.



Algoritmo FFD

FFD: first fit decreasing

Ordene os itens por tamanho.
Coloque cada item no primeiro bin onde ele cabe.



Para uma instância I,
seja FFD(I) o número de bins usados pelo FFD.

Vale que FFD$(I) \leq \frac{11}{9}\mathrm{OPT}(I) + 1$ para toda instância I.

Algoritmo FFD

Teorema: A menos que P $=$ NP, não existe α-aproximação para o bin packing com $\alpha < 3/2$.

Algoritmo FFD

Teorema: A menos que P $=$ NP, não existe α-aproximação para o bin packing com $\alpha < 3/2$.

FFD: first fit decreasing

Para uma instância I,
vale que FFD$(I) \leq \frac{11}{9}\mathrm{OPT}(I) + 1$ para toda instância I.

Algoritmo FFD

Teorema: A menos que P $=$ NP, não existe α-aproximação para o bin packing com $\alpha < 3/2$.

FFD: first fit decreasing

Para uma instância I,
vale que FFD$(I) \leq \frac{11}{9}\mathrm{OPT}(I) + 1$ para toda instância I.

Isso não é contraditório?

Algoritmo FFD

Teorema: A menos que P $=$ NP, não existe α-aproximação para o bin packing com $\alpha < 3/2$.

FFD: first fit decreasing

Para uma instância I,
vale que FFD$(I) \leq \frac{11}{9}\mathrm{OPT}(I) + 1$ para toda instância I.

Isso não é contraditório?

Não. Inclusive existe um algoritmo polinomial que sempre usa $\mathrm{OPT}(I) + 1$ bins no máximo!

Algoritmo FFD

Teorema: A menos que P $=$ NP, não existe α-aproximação para o bin packing com $\alpha < 3/2$.

FFD: first fit decreasing

Para uma instância I,
vale que FFD$(I) \leq \frac{11}{9}\mathrm{OPT}(I) + 1$ para toda instância I.

Isso não é contraditório?

Não. Inclusive existe um algoritmo polinomial que sempre usa $\mathrm{OPT}(I) + 1$ bins no máximo!

Vamos mostrar um APTAS para o bin packing.

APTAS para o bin packing

APTAS: asymptotic PTAS

Família de algoritmos parametrizada por $\epsilon > 0$ onde cada algoritmo usa no máximo $(1+\epsilon)\mathrm{OPT}(I) + c$ bins para a instância I, e c é uma constante.

APTAS para o bin packing

APTAS: asymptotic PTAS

Família de algoritmos parametrizada por $\epsilon > 0$ onde cada algoritmo usa no máximo $(1+\epsilon)\mathrm{OPT}(I) + c$ bins para a instância I, e c é uma constante.

Usaremos no APTAS a PD vista na aula passada.

APTAS para o bin packing

APTAS: asymptotic PTAS

Família de algoritmos parametrizada por $\epsilon > 0$ onde cada algoritmo usa no máximo $(1+\epsilon)\mathrm{OPT}(I) + c$ bins para a instância I, e c é uma constante.

Usaremos no APTAS a PD vista na aula passada.

k: número de durações distintas das tarefas

APTAS para o bin packing

APTAS: asymptotic PTAS

Família de algoritmos parametrizada por $\epsilon > 0$ onde cada algoritmo usa no máximo $(1+\epsilon)\mathrm{OPT}(I) + c$ bins para a instância I, e c é uma constante.

Usaremos no APTAS a PD vista na aula passada.

k: número de durações distintas das tarefas

A PD calculava o valor de $\mathrm{OPT}(n_1, \ldots, n_k)$, onde n_i é o número de tarefas com duração d_i.

$\mathrm{OPT}(n_1, \ldots, n_k)$ é o menor número de máquinas necessárias para escalonar as $n_1 + \cdots + n_k$ tarefas, com cada máquina tendo tempo de conclusão no máximo T.

Adaptação da PD

k: número de durações distintas das tarefas

n_i: número de tarefas com duração d_i

$\mathrm{OPT}(n_1, \ldots, n_k)$: menor número de máquinas necessárias para escalonar as $n_1 + \cdots + n_k$ tarefas.

Cada máquina com tempo de conclusão no máximo T.

Adaptação da PD

k: número de durações distintas das tarefas

n_i: número de tarefas com duração d_i

$\mathrm{OPT}(n_1, \ldots, n_k)$: menor número de máquinas necessárias para escalonar as $n_1 + \cdots + n_k$ tarefas.

Cada máquina com tempo de conclusão no máximo T.

Divida a duração de toda tarefa por T.

Resultado: instância do bin packing

OPT: exatamente o ótimo do bin packing.

Adaptação da PD

k: número de durações distintas das tarefas

n_i: número de tarefas com duração d_i

$\mathrm{OPT}(n_1, \ldots, n_k)$: menor número de máquinas necessárias para escalonar as $n_1 + \cdots + n_k$ tarefas.

Cada máquina com tempo de conclusão no máximo T.

Divida a duração de toda tarefa por T.

Resultado: instância do bin packing

OPT: exatamente o ótimo do bin packing.

Logo a PD resolve o bin packing com
número limitado de tamanhos de itens distintos e
um número limitado de itens cabem em um bin.

Resolução do bin packing

Ideia: Gerarmos uma instância do bin packing onde possamos aplicar a PD.

Resolução do bin packing

Ideia: Gerarmos uma instância do bin packing
onde possamos aplicar a PD.

De solução obtida, construímos uma solução para
a instância original usando apenas mais uns poucos bins.

Resolução do bin packing

Ideia: Gerarmos uma instância do bin packing
onde possamos aplicar a PD.

De solução obtida, construímos uma solução para
a instância original usando apenas mais uns poucos bins.

Primeiro: mostrar que podemos ignorar itens pequenos.

Resolução do bin packing

Ideia: Gerarmos uma instância do bin packing
onde possamos aplicar a PD.

De solução obtida, construímos uma solução para
a instância original usando apenas mais uns poucos bins.

Primeiro: mostrar que podemos ignorar itens pequenos.

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$
em ℓ bins pode ser extendidos para um empacotamento
da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Resolução do bin packing

Ideia: Gerarmos uma instância do bin packing
onde possamos aplicar a PD.

De solução obtida, construímos uma solução para
a instância original usando apenas mais uns poucos bins.

Primeiro: mostrar que podemos ignorar itens pequenos.

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$
em ℓ bins pode ser extendidos para um empacotamento
da entrada toda com no máximo $\max\{\ell, \frac{\mathrm{size}(I)}{1-\gamma} + 1\}$ bins.

FF: First Fit

Itens pequenos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\mathrm{size}(I)}{1-\gamma} + 1\}$ bins.

Itens pequenos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\mathrm{size}(I)}{1-\gamma} + 1\}$ bins.

Usamos o FF (first fit) nos pequenos a partir do empacotamento dos graúdos.

Itens pequenos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\mathrm{size}(I)}{1-\gamma} + 1\}$ bins.

Usamos o FF (first fit) nos pequenos a partir do empacotamento dos graúdos.

Se FF não usa nenhum bin adicional, termina com ℓ bins.

Itens pequenos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Usamos o FF (first fit) nos pequenos a partir do empacotamento dos graúdos.

Se FF não usa nenhum bin adicional, termina com ℓ bins.

Se usa, então todos, menos possivelmente o último bin, estão com pelo menos $1 - \gamma$ de ocupação.

Itens pequenos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\mathrm{size}(I)}{1-\gamma} + 1\}$ bins.

Usamos o FF (first fit) nos pequenos a partir do empacotamento dos graúdos.

Se FF não usa nenhum bin adicional, termina com ℓ bins.

Se usa, então todos, menos possivelmente o último bin, estão com pelo menos $1 - \gamma$ de ocupação.

Se termina com $k + 1$ bins, então $\mathrm{size}(I) \geq k(1 - \gamma)$.
Ou seja, $k \leq \frac{\mathrm{size}(I)}{1-\gamma}$.

Instância dos itens graúdos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\mathrm{size}(I)}{1-\gamma} + 1\}$ bins.

Instância dos itens graúdos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Para $\gamma = \epsilon/2$, vale que $1/(1-\epsilon/2) \leq 1+\epsilon$.

Instância dos itens graúdos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Para $\gamma = \epsilon/2$, vale que $1/(1 - \epsilon/2) \leq 1 + \epsilon$.

No máximo $\max\{\ell, (1 + \epsilon)\text{OPT}(I) + 1\}$ bins!

Instância dos itens graúdos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Para $\gamma = \epsilon/2$, vale que $1/(1 - \epsilon/2) \leq 1 + \epsilon$.

No máximo $\max\{\ell, (1 + \epsilon)\text{OPT}(I) + 1\}$ bins!

Primeira propriedade: no máximo $2/\epsilon$ itens por bin!

Instância dos itens graúdos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Para $\gamma = \epsilon/2$, vale que $1/(1-\epsilon/2) \leq 1+\epsilon$.

No máximo $\max\{\ell, (1+\epsilon)\text{OPT}(I) + 1\}$ bins!

Primeira propriedade: no máximo $2/\epsilon$ itens por bin!

Para poder usar a PD,

falta limitar o número de tamanhos distintos.

Instância dos itens graúdos

Lema: Qualquer empacotamento dos itens com $a_i \geq \gamma$ em ℓ bins pode ser extendidos para um empacotamento da entrada toda com no máximo $\max\{\ell, \frac{\text{size}(I)}{1-\gamma} + 1\}$ bins.

Para $\gamma = \epsilon/2$, vale que $1/(1-\epsilon/2) \leq 1+\epsilon$.

No máximo $\max\{\ell, (1+\epsilon)\text{OPT}(I)+1\}$ bins!

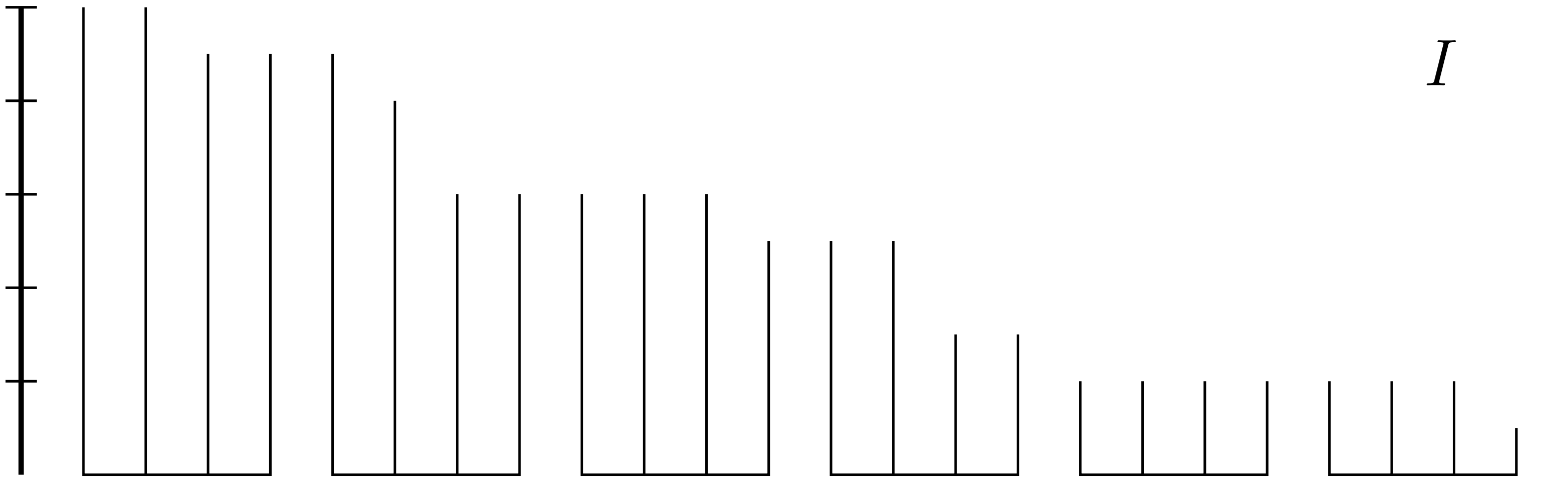
Primeira propriedade: no máximo $2/\epsilon$ itens por bin!

Para poder usar a PD,

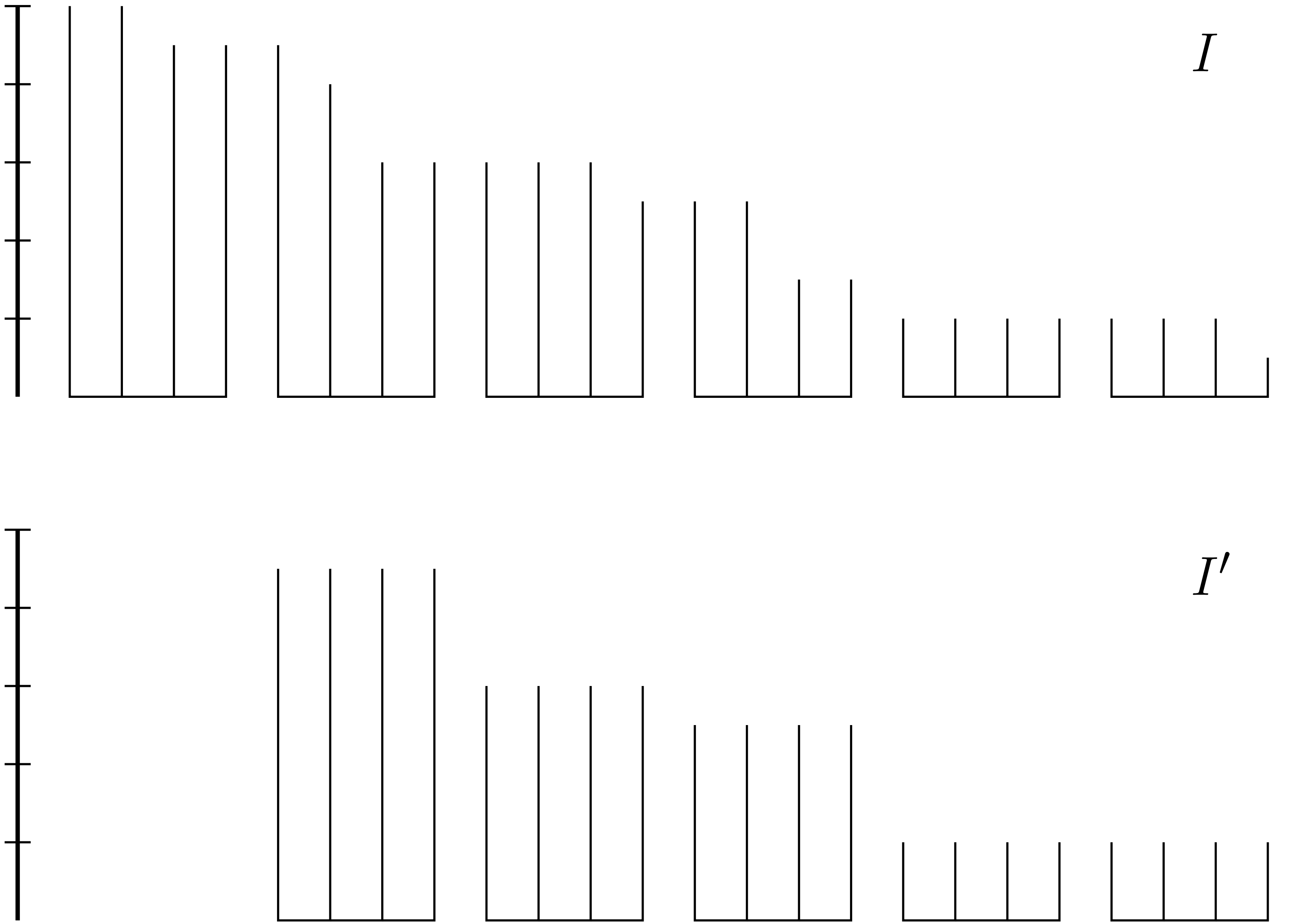
falta limitar o número de tamanhos distintos.

Agrupamento linear!

Agrupamento linear



Agrupamento linear



Agrupamento linear

Considere apenas itens com $a_i \geq \gamma$.

k: parâmetro a ser definido à frente

Agrupamento linear

Considere apenas itens com $a_i \geq \gamma$.

k: parâmetro a ser definido à frente

Ordene os itens por tamanho.

Agrupamento linear

Considere apenas itens com $a_i \geq \gamma$.

k: parâmetro a ser definido à frente

Ordene os itens por tamanho.

Agrupe os itens de k em k. (Último grupo: $h \leq k$ itens.)

Agrupamento linear

Considere apenas itens com $a_i \geq \gamma$.

k: parâmetro a ser definido à frente

Ordene os itens por tamanho.

Agrupe os itens de k em k. (Último grupo: $h \leq k$ itens.)

Instância arredondada I':

Jogue fora o primeiro grupo.

Arredonde cada item para
o tamanho do maior item do seu grupo.

Agrupamento linear

Itens com $a_i \geq \gamma$ e k a ser definido à frente

Ordene os itens por tamanho e agrupe os itens de k em k.

Instância arredondada I':

Jogue fora o primeiro grupo.

Arredonde cada item para
o tamanho do maior item do seu grupo.

Agrupamento linear

Itens com $a_i \geq \gamma$ e k a ser definido à frente

Ordene os itens por tamanho e agrupe os itens de k em k.

Instância arredondada I':

Jogue fora o primeiro grupo.

Arredonde cada item para
o tamanho do maior item do seu grupo.

Lema: $\mathrm{OPT}(I') \leq \mathrm{OPT}(I) \leq \mathrm{OPT}(I') + k$.

Agrupamento linear

Itens com $a_i \geq \gamma$ e k a ser definido à frente

Ordene os itens por tamanho e agrupe os itens de k em k.

Instância arredondada I':

Jogue fora o primeiro grupo.

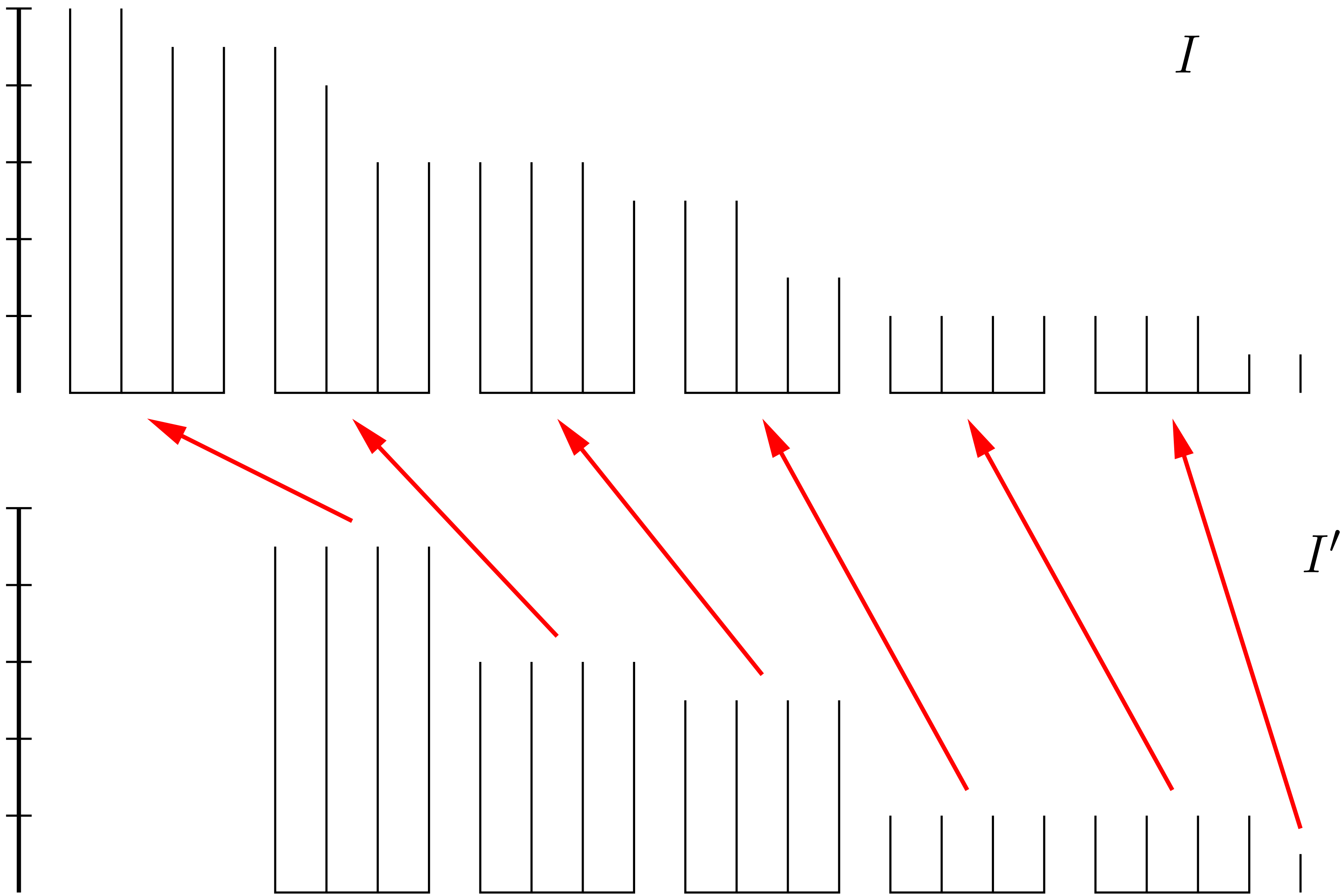
Arredonde cada item para
o tamanho do maior item do seu grupo.

Lema: $\mathrm{OPT}(I') \leq \mathrm{OPT}(I) \leq \mathrm{OPT}(I') + k$.

Prova feita em aula.

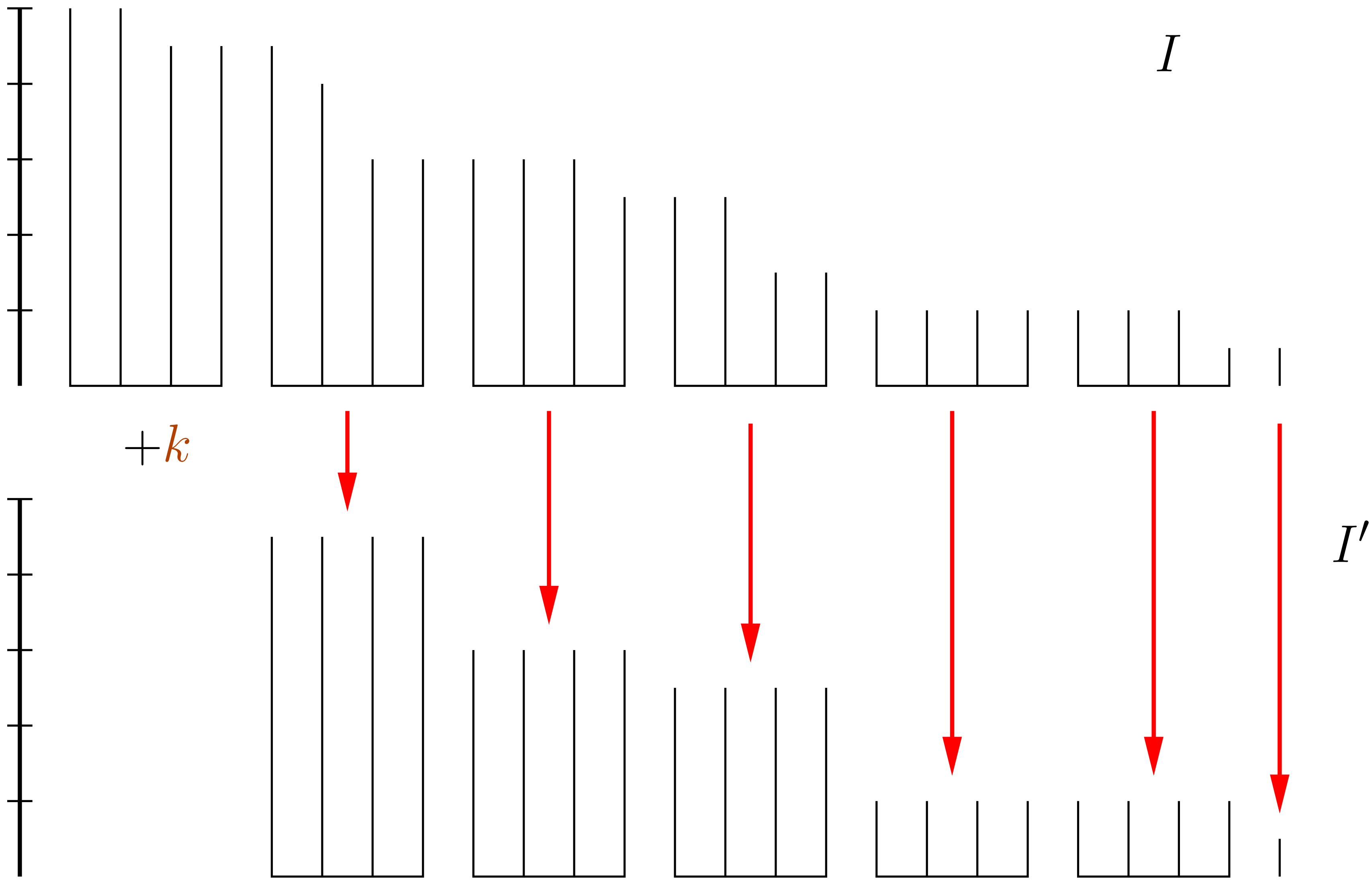
Agrupamento linear

Lema: $\mathrm{OPT}(I') \leq \mathrm{OPT}(I)$.



Agrupamento linear

Lema: $\mathrm{OPT}(I) \leq \mathrm{OPT}(I') + k$.



Instância arredondada

I: instância só com os graúdos (com $a_i \geq \epsilon/2$)
n: número de itens em I

Claro que $\mathrm{size}(I) \geq n\,\epsilon/2$.

Instância arredondada

I: instância só com os graúdos (com $a_i \geq \epsilon/2$)
n: número de itens em I

Claro que $\mathrm{size}(I) \geq n\,\epsilon/2$.

I' tem no máximo n/k tamanhos distintos.

Instância arredondada

I: instância só com os graúdos (com $a_i \geq \epsilon/2$)
n: número de itens em I

Claro que $\mathrm{size}(I) \geq n\,\epsilon/2$.

I' tem no máximo n/k tamanhos distintos.

Se $k = \lfloor \epsilon\,\mathrm{size}(I) \rfloor$, então

$$\frac{n}{k} \;\leq\; \frac{2n}{\epsilon\,\mathrm{size}(I)} \;\leq\; \frac{4}{\epsilon^2},$$

onde usamos que $\lfloor \alpha \rfloor \geq \alpha/2$ para $\alpha \geq 1$.

Instância arredondada

I: instância só com os graúdos (com $a_i \geq \epsilon/2$)
n: número de itens em I

Claro que $\mathrm{size}(I) \geq n\,\epsilon/2$.

I' tem no máximo n/k tamanhos distintos.

Se $k = \lfloor \epsilon\,\mathrm{size}(I) \rfloor$, então

$$\frac{n}{k} \;\leq\; \frac{2n}{\epsilon\,\mathrm{size}(I)} \;\leq\; \frac{4}{\epsilon^2},$$

onde usamos que $\lfloor \alpha \rfloor \geq \alpha/2$ para $\alpha \geq 1$.

Se $\epsilon\,\mathrm{size}(I) < 1$, então
$n < (1/\epsilon)/(\epsilon/2) = 2/\epsilon^2$ e aplicamos a PD direto em I.

Instância arredondada

I: instância só com os graúdos (com $a_i \geq \epsilon/2$)
n: número de itens em I

Claro que $\mathrm{size}(I) \geq n\,\epsilon/2$.

I' tem no máximo n/k tamanhos distintos.

Se $\epsilon\,\mathrm{size}(I) < 1$, então $n < (1/\epsilon)/(\epsilon/2) = 2/\epsilon^2$
e aplicamos a PD direto em I.

Instância arredondada

I: instância só com os graúdos (com $a_i \geq \epsilon/2$)
n: número de itens em I

Claro que $\mathrm{size}(I) \geq n\,\epsilon/2$.

I' tem no máximo n/k tamanhos distintos.

Se $\epsilon\,\mathrm{size}(I) < 1$, então $n < (1/\epsilon)/(\epsilon/2) = 2/\epsilon^2$
e aplicamos a PD direto em I.

Senão, para $k = \lfloor \epsilon\,\mathrm{size}(I) \rfloor$,

$$\frac{n}{k} \;\leq\; \frac{2n}{\epsilon\,\mathrm{size}(I)} \;\leq\; \frac{4}{\epsilon^2},$$

e aplicamos a PD em I'
(I' tem um número limitado de tamanhos distintos).

Adicionando os itens pequenos

Do empacotamento para I', obtemos um para I.

Depois adicionamos os itens pequenos com o FF.

Adicionando os itens pequenos

Do empacotamento para I', obtemos um para I.

Depois adicionamos os itens pequenos com o FF.

Usaremos no máximo $(1+\epsilon)\mathrm{OPT}(I)+1$ bins.

Adicionando os itens pequenos

Do empacotamento para I', obtemos um para I.

Depois adicionamos os itens pequenos com o FF.

Usaremos no máximo $(1+\epsilon)\mathrm{OPT}(I)+1$ bins.

Teorema: Para todo $\epsilon > 0$, existe algoritmo polinomial para o bin packing que produz empacotamento em no máximo $(1+\epsilon)\mathrm{OPT}(I)+1$ bins, para a instância I.

Ou seja, existe um APTAS para o bin packing.

Adicionando os itens pequenos

Do empacotamento para I', obtemos um para I.

Depois adicionamos os itens pequenos com o FF.

Usaremos no máximo $(1+\epsilon)\mathrm{OPT}(I)+1$ bins.

Teorema: Para todo $\epsilon > 0$, existe algoritmo polinomial para o bin packing que produz empacotamento em no máximo $(1+\epsilon)\mathrm{OPT}(I)+1$ bins, para a instância I.

Ou seja, existe um APTAS para o bin packing.

Prova feita em aula.

Adicionando os itens pequenos

ℓ: número de bins usados com os itens graúdos.

Sabemos que, já com os pequenos, usamos no máximo

$$\max\{\ell, (1+\epsilon)\mathrm{OPT}(I)\} + 1\} \text{ bins.}$$

Adicionando os itens pequenos

ℓ: número de bins usados com os itens graúdos.

Sabemos que, já com os pequenos, usamos no máximo

$$\max\{\ell, (1+\epsilon)\mathrm{OPT}(I)\} + 1\} \text{ bins.}$$

Pelo lema anterior, usamos com os graúdos no máximo

$$\mathrm{OPT}(I') + k \leq \mathrm{OPT}(I) + k \text{ bins,}$$

onde $k = \lfloor \epsilon\, \mathrm{size}(I) \rfloor$.

Adicionando os itens pequenos

ℓ: número de bins usados com os itens graúdos.

Sabemos que, já com os pequenos, usamos no máximo

$$\max\{\ell, (1+\epsilon)\mathrm{OPT}(I)\} + 1\} \text{ bins.}$$

Pelo lema anterior, usamos com os graúdos no máximo

$$\mathrm{OPT}(I') + k \leq \mathrm{OPT}(I) + k \text{ bins,}$$

onde $k = \lfloor \epsilon\,\mathrm{size}(I) \rfloor$.

Assim $\ell \leq \mathrm{OPT}(I) + k \leq \mathrm{OPT}(I) + \epsilon\,\mathrm{size}(I) \leq (1+\epsilon)\mathrm{OPT}(I)$.