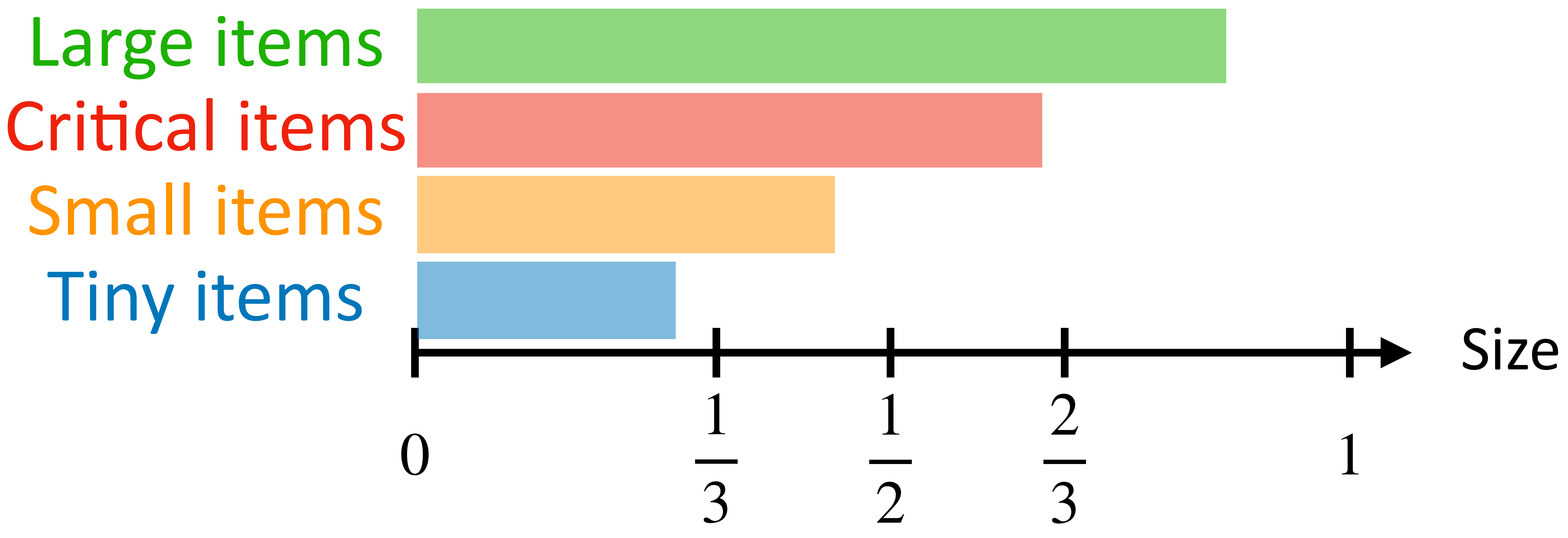
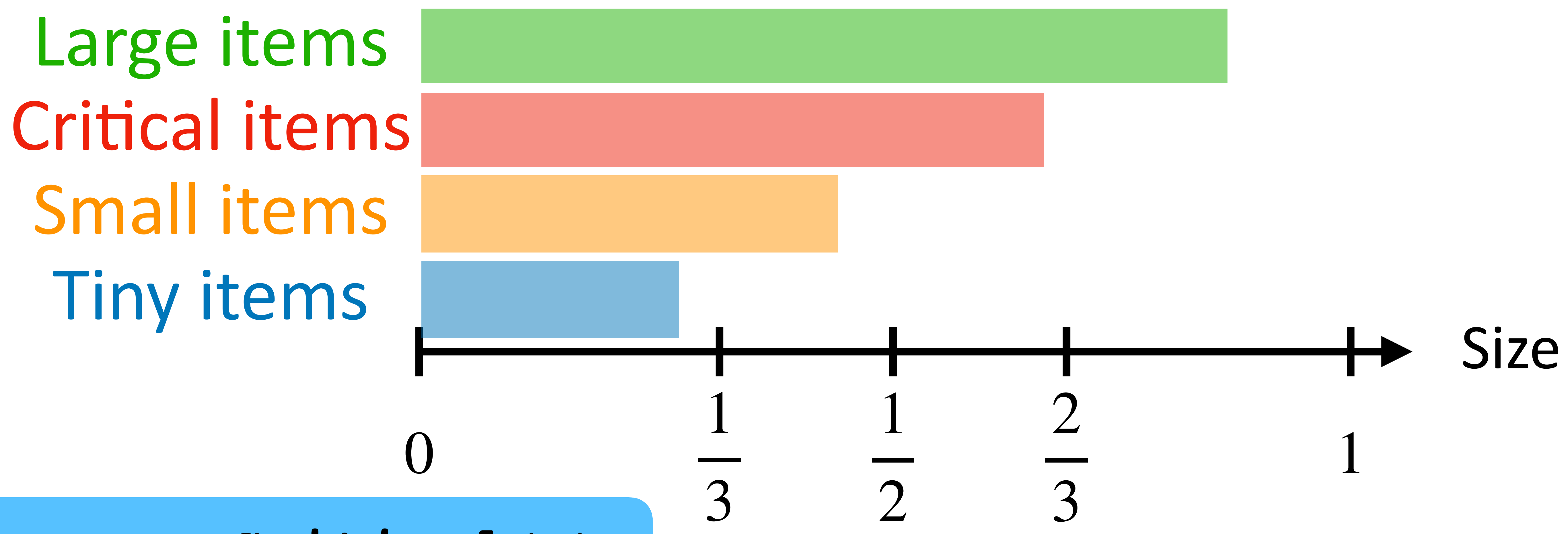
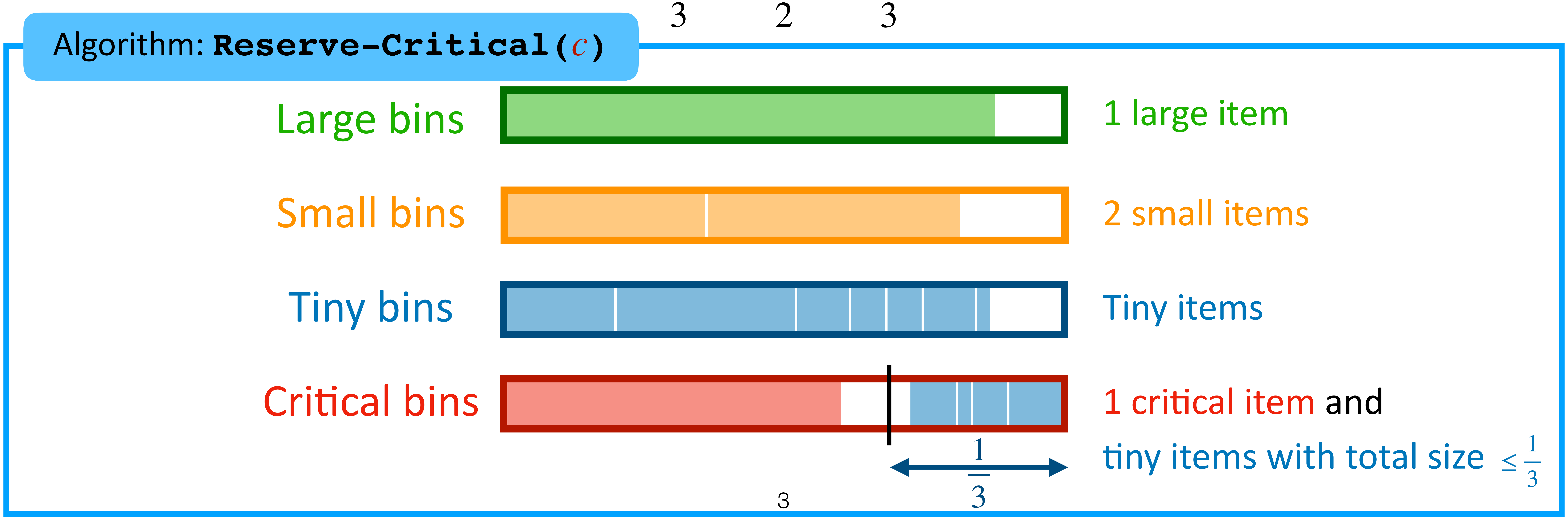
Bin Packing with Advice

Bin Packing with Advice



Bin Packing with Advice





Reserve-Critical Algorithm

Reserve-Critical(c)

- Open c critical bins. In each critical bin, pack one critical item and tiny items with total size $\leq 1/3$
- Pack large items into large bins, small items into small bins
- When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds $1/3$); otherwise, pack it into a tiny bin by First-Fit

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5-competitive

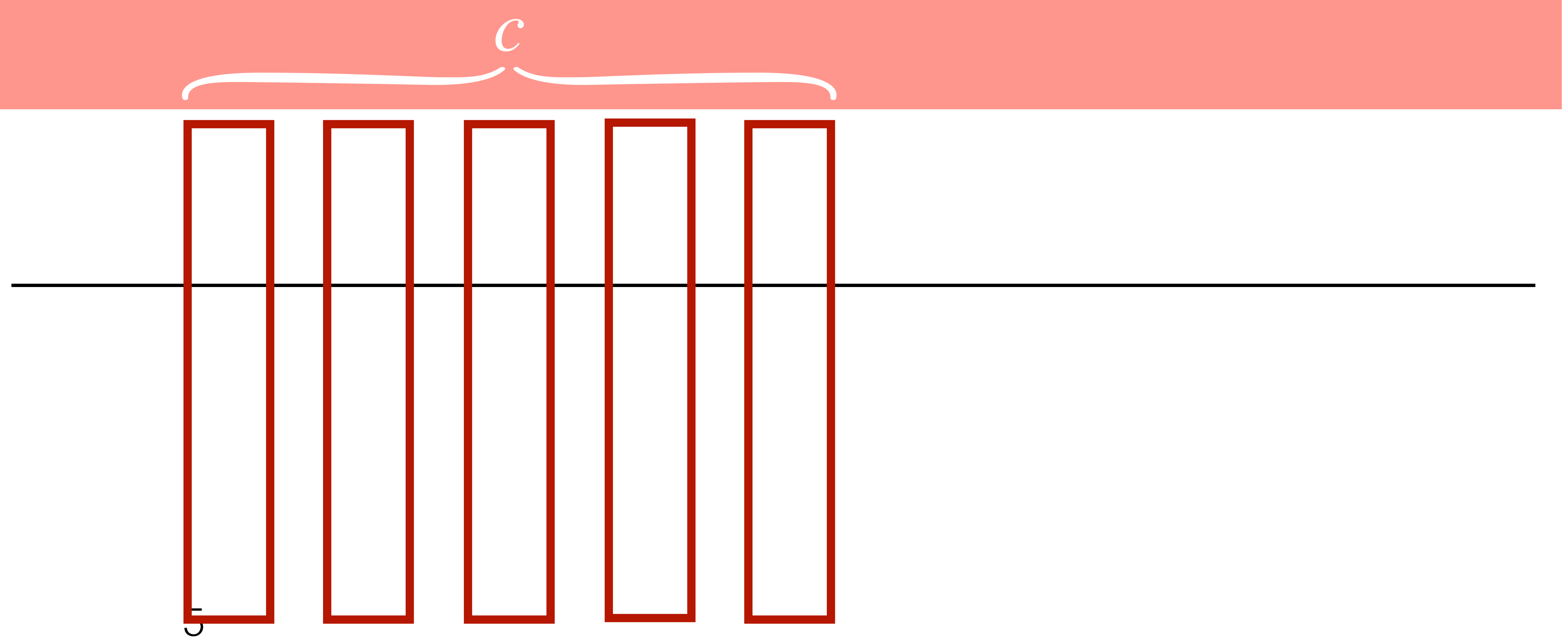
Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full (may with a constant number of exceptions)

`Reserve-Critical` Algorithm

Reserve-Critical(c)

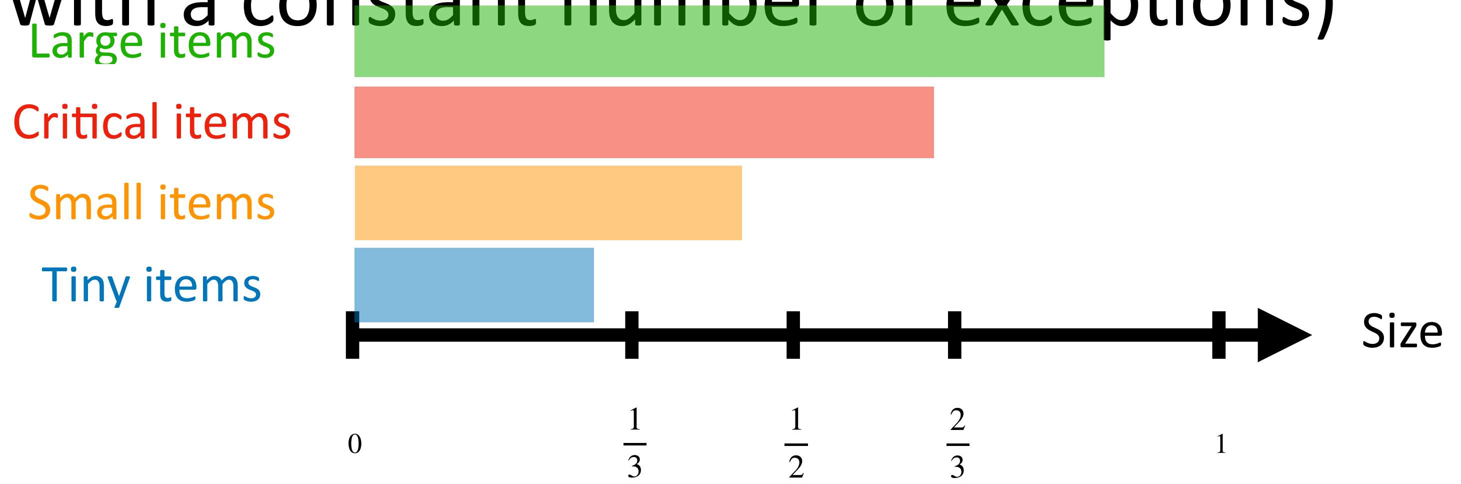
- `Open` c `critical bins. In each critical bin, pack one critical item and tiny items with total size` $\leq 1/3$
- `Pack large items into large bins, small items into small bins`
- `When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds` $1/3$`); otherwise, pack it into a tiny bin by First-Fit`

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5 -competitive



Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)



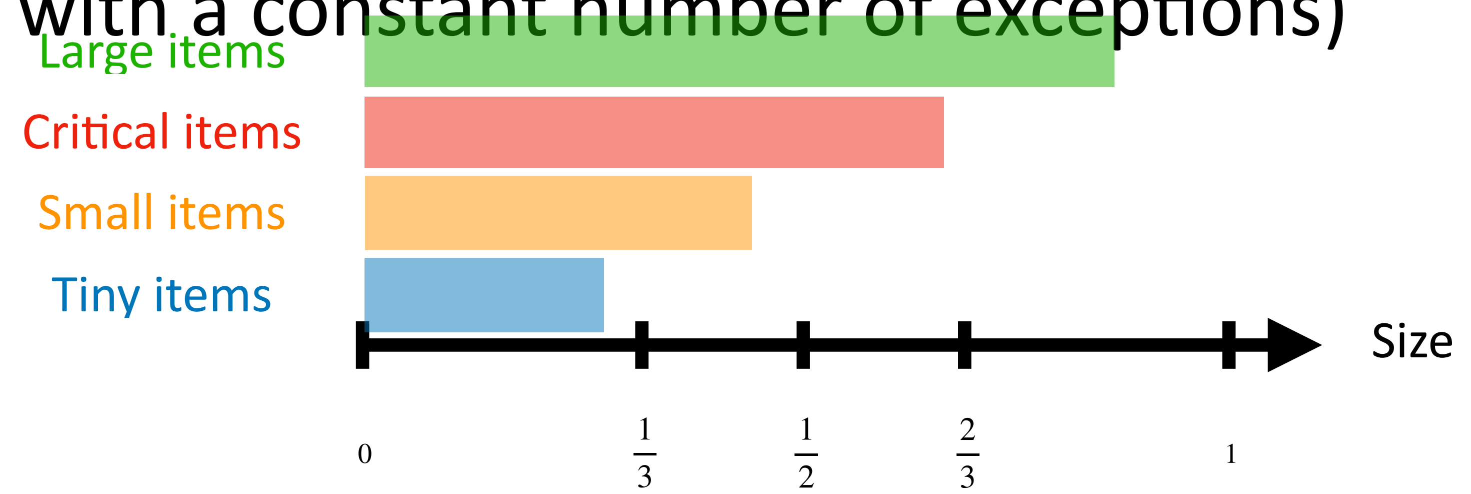
`Reserve-Critical` Algorithm

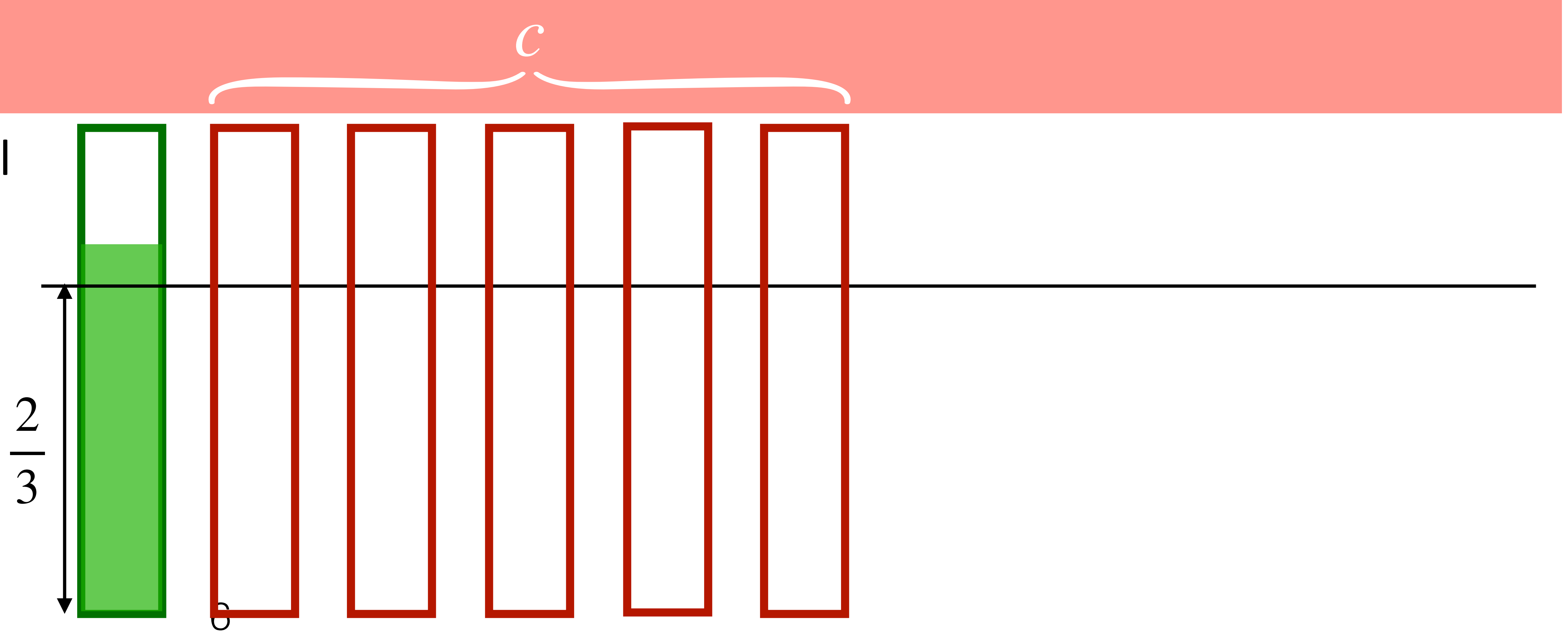
Reserve-Critical(c)

- `Open` c `critical bins. In each critical bin, pack one critical item and tiny items with total size` $\leq 1/3$
- `Pack large items into large bins, small items into small bins`
- `When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds` $1/3$`); otherwise, pack it into a tiny bin by First-Fit`

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5 -competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full (may with a constant number of exceptions)





`Reserve-Critical` Algorithm

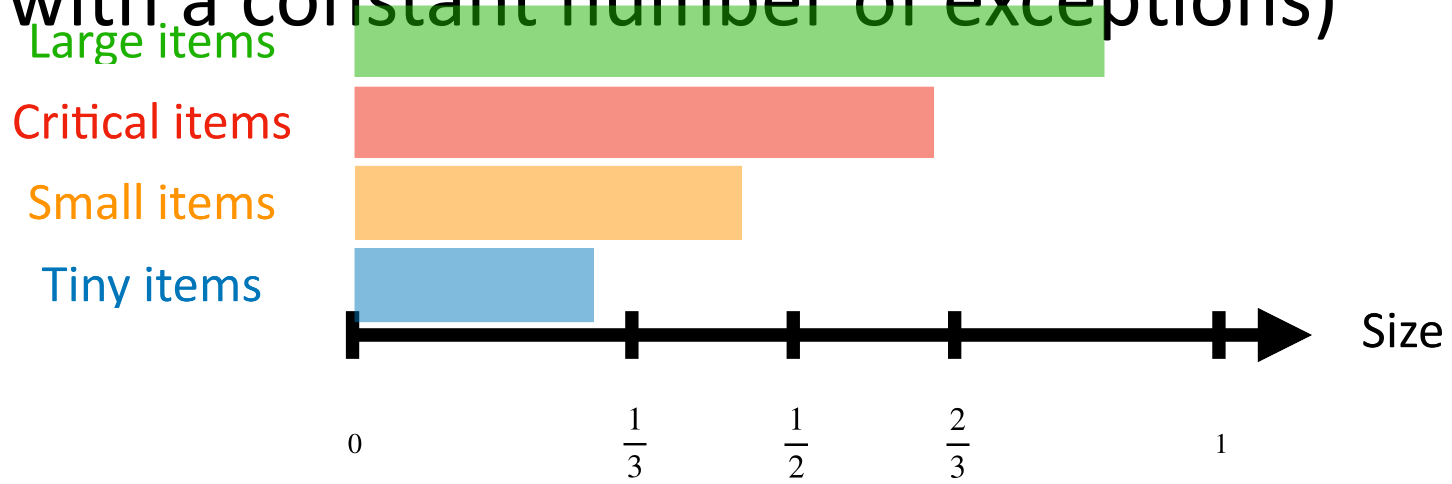
Reserve-Critical(c)

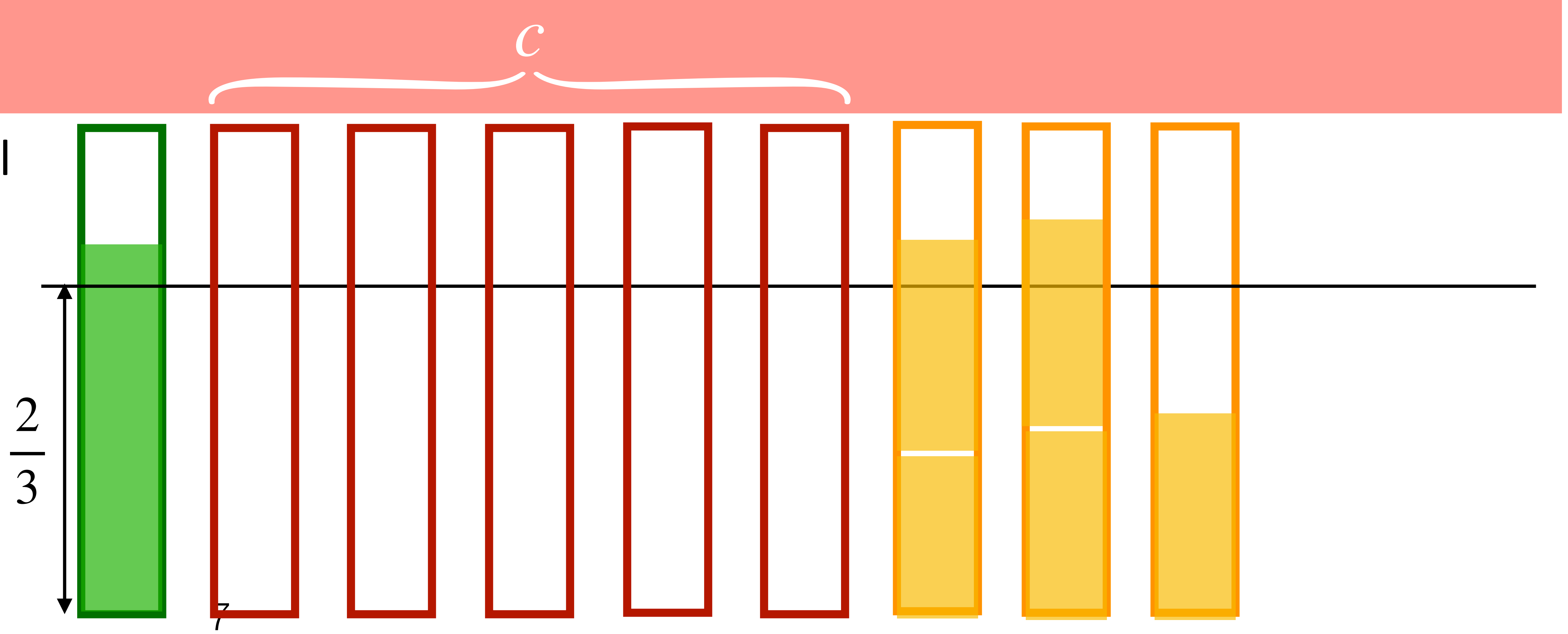
- `Open` c `critical bins. In each critical bin, pack one critical item and tiny items with total size` $\leq 1/3$
- `Pack large items into large bins, small items into small bins`
- `When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds` $1/3$`); otherwise, pack it into a tiny bin by First-Fit`

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5-competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)





Reserve-Critical Algorithm

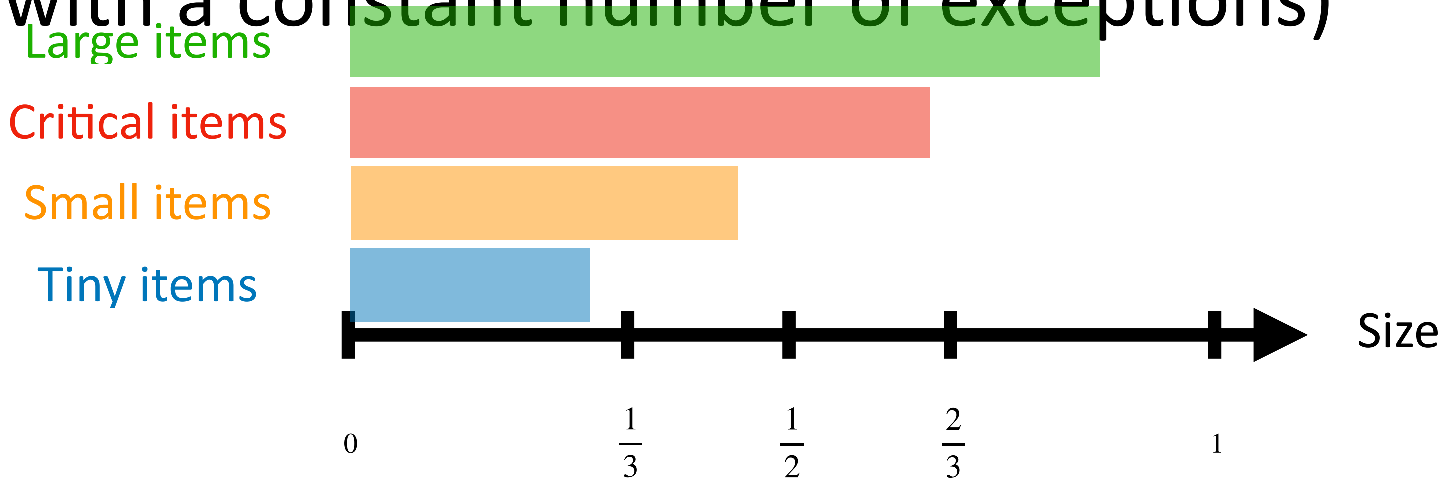
Reserve-Critical(c)

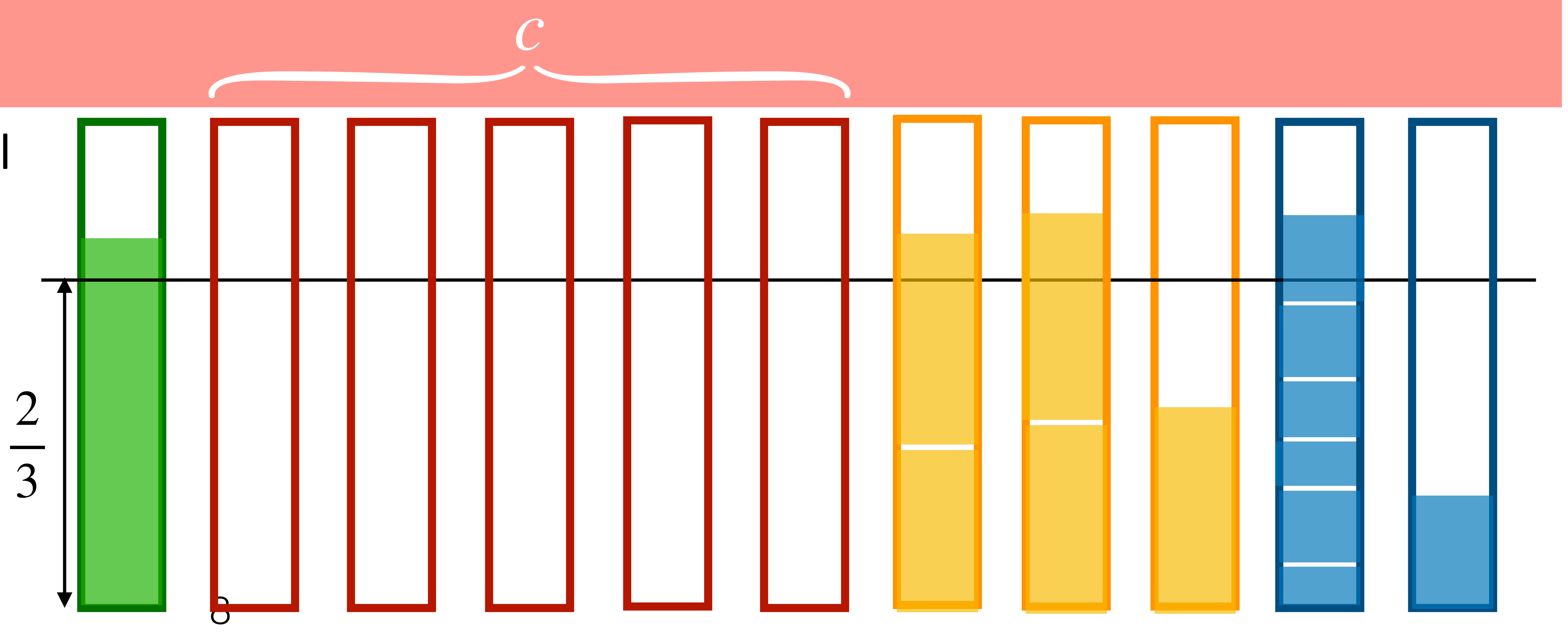
- Open c critical bins. In each critical bin, pack one critical item and tiny items with total size $\leq 1/3$
- Pack large items into large bins, small items into small bins
- When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds $1/3$); otherwise, pack it into a tiny bin by First-Fit

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5-competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)





`Reserve-Critical` Algorithm

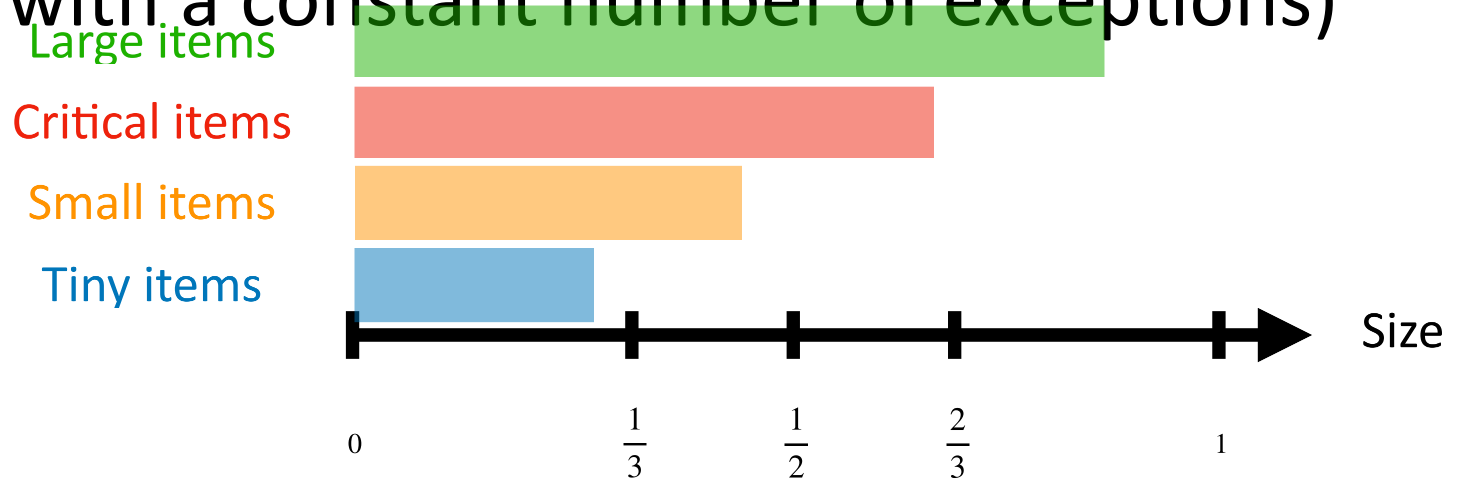
Reserve-Critical(c)

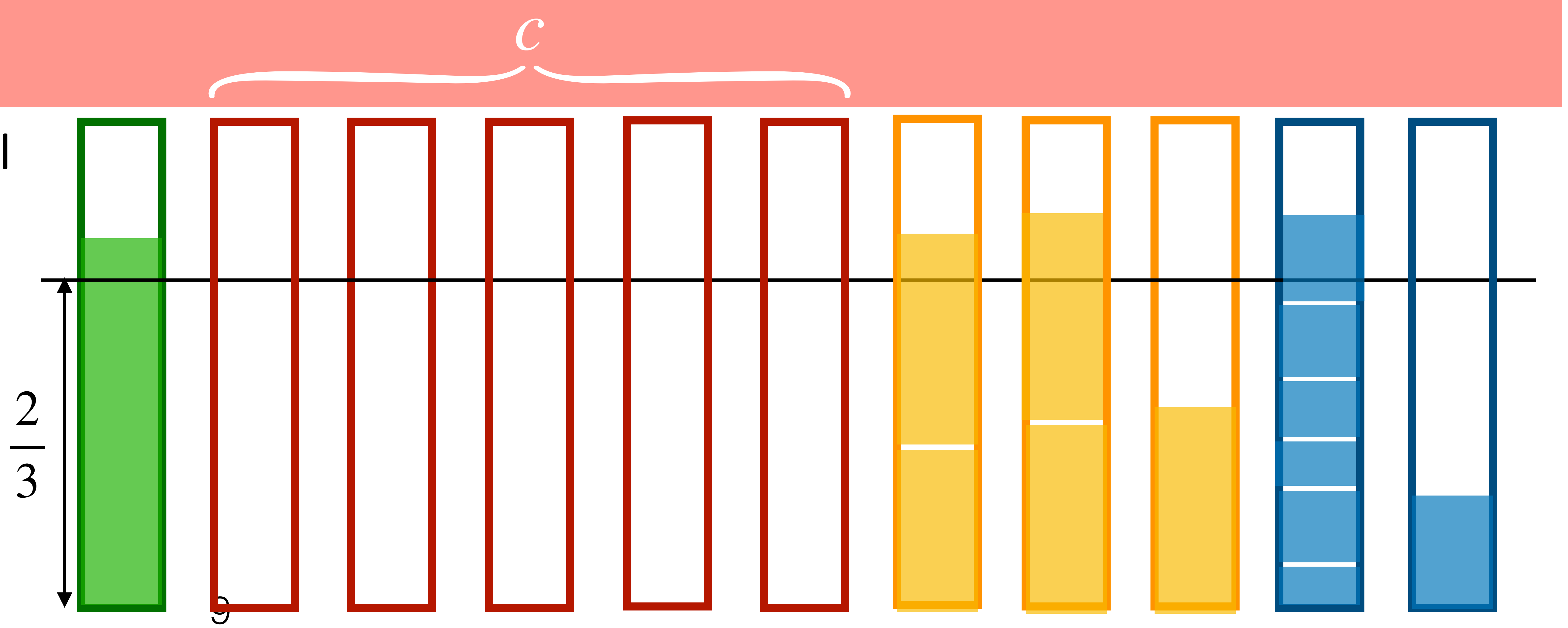
- `Open` c `critical bins. In each critical bin, pack one critical item and tiny items with total size` $\leq 1/3$
- `Pack large items into large bins, small items into small bins`
- `When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds` $1/3$`); otherwise, pack it into a tiny bin by First-Fit`

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5 -competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)





Reserve-Critical Algorithm

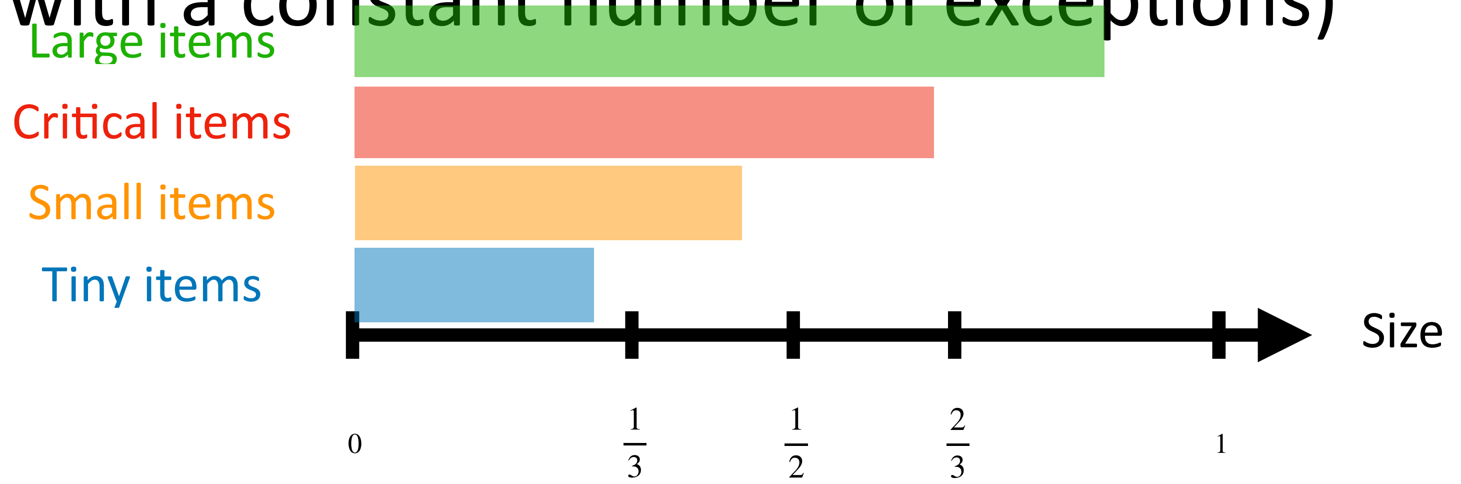
Reserve-Critical(c)

- Open c critical bins. In each critical bin, pack one critical item and tiny items with total size $\leq 1/3$
- Pack large items into large bins, small items into small bins
- When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds $1/3$); otherwise, pack it into a tiny bin by First-Fit

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5 -competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)



`Reserve-Critical` Algorithm

Reserve-Critical(c)

- `Open` c `critical bins. In each critical bin, pack one critical item and tiny items with total size` $\leq 1/3$
- `Pack large items into large bins, small items into small bins`
- `When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds` $1/3$`); otherwise, pack it into a tiny bin by First-Fit`

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5 -competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)

For the critical bins, they are all at least $\frac{2}{3}$-full, or the optimal also need all these critical bins

Reserve-Critical Algorithm

Reserve-Critical(c)

- Open c critical bins. In each critical bin, pack one critical item and tiny items with total size $\leq 1/3$
- Pack large items into large bins, small items into small bins
- When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds $1/3$); otherwise, pack it into a tiny bin by First-Fit

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5 -competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)

For the critical bins, they are all at least $\frac{2}{3}$-full, or the optimal also need all these critical bins

`Reserve-Critical` Algorithm

Reserve-Critical(c)

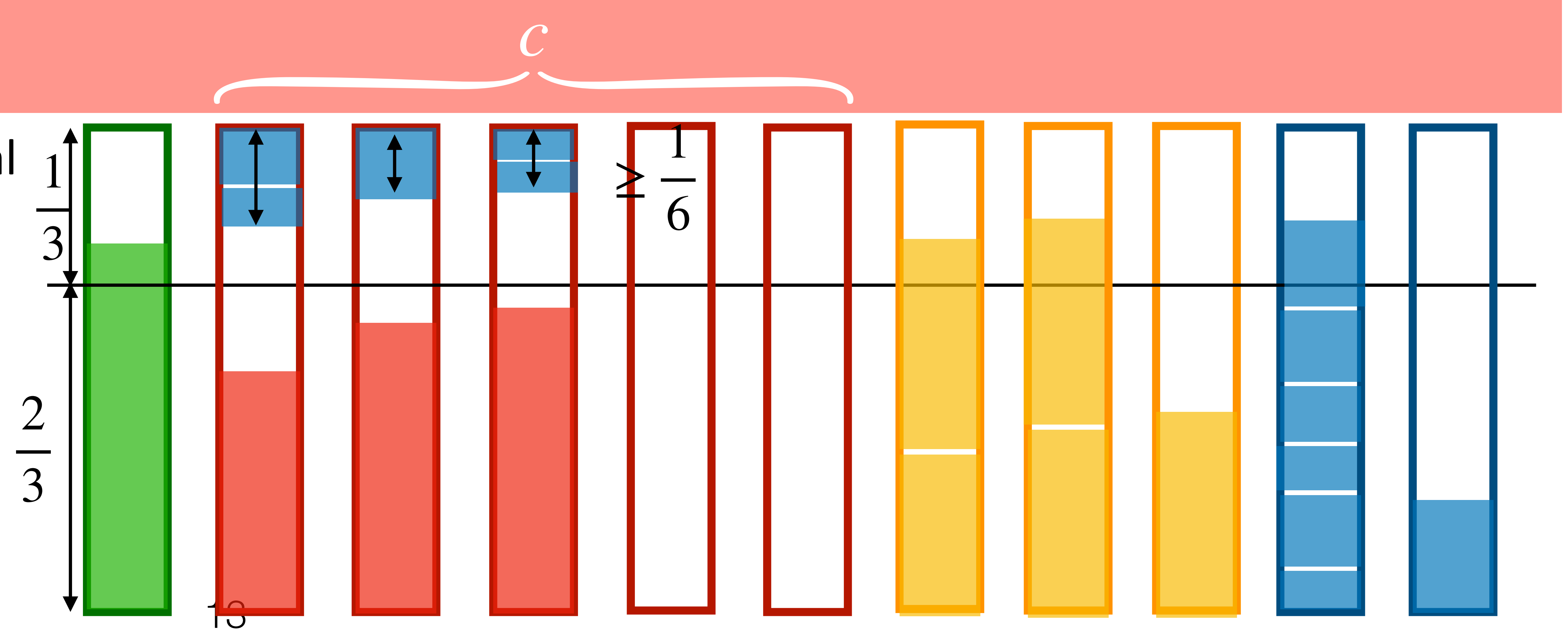
- `Open` c `critical bins. In each critical bin, pack one critical item and tiny items with total size` $\leq 1/3$
- `Pack large items into large bins, small items into small bins`
- `When a tiny item arrives, pack it into a critical bin if possible (without making the total size of tiny items in that bin exceeds` $1/3$`); otherwise, pack it into a tiny bin by First-Fit`

- Theorem: If the number of critical items is known, **Reserve-Critical** is 1.5-competitive

Reserve-Critical algorithm guarantees that every non-critical bin is at least $\frac{2}{3}$-full

(may with a constant number of exceptions)

For the critical bins, they are all at least $\frac{2}{3}$-full, or the optimal also need all these critical bins



What Happened

- `Reserve-Critical` Algorithm is a (semi-online) algorithm that uses the *advice* of critical items to improve the competitive ratio
  - 1.5-competitive

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}$, α **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \frac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\frac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively

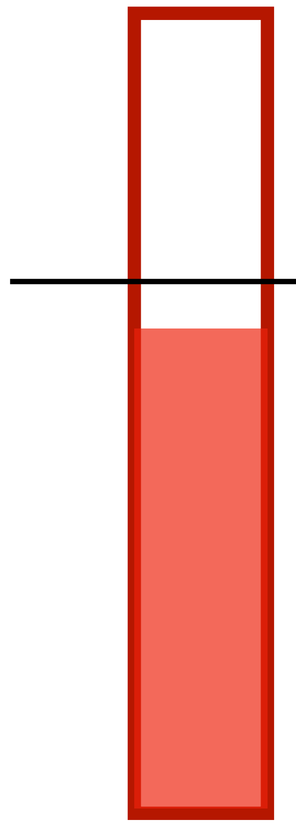
Prediction: the number of critical bins and the number of tiny bins

$$\frac{c}{c+t}$$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}$, α **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- `Let` $\beta = \min\{r, \alpha\}$
- `When a critical item arrives, pack it into a critical bin`
- `When a tiny item arrives, pack it into an available critical bin`
  - `If there is no available critical bin, pack it into a tiny bin`
    - `If there is no tiny bin, open a new bin B`
      - `B is a critical bin if` $\dfrac{c'}{c'+t'} < \beta$, `otherwise, it is a tiny bin`
- `Large items and small items are packed into large bins and small bins respectively`

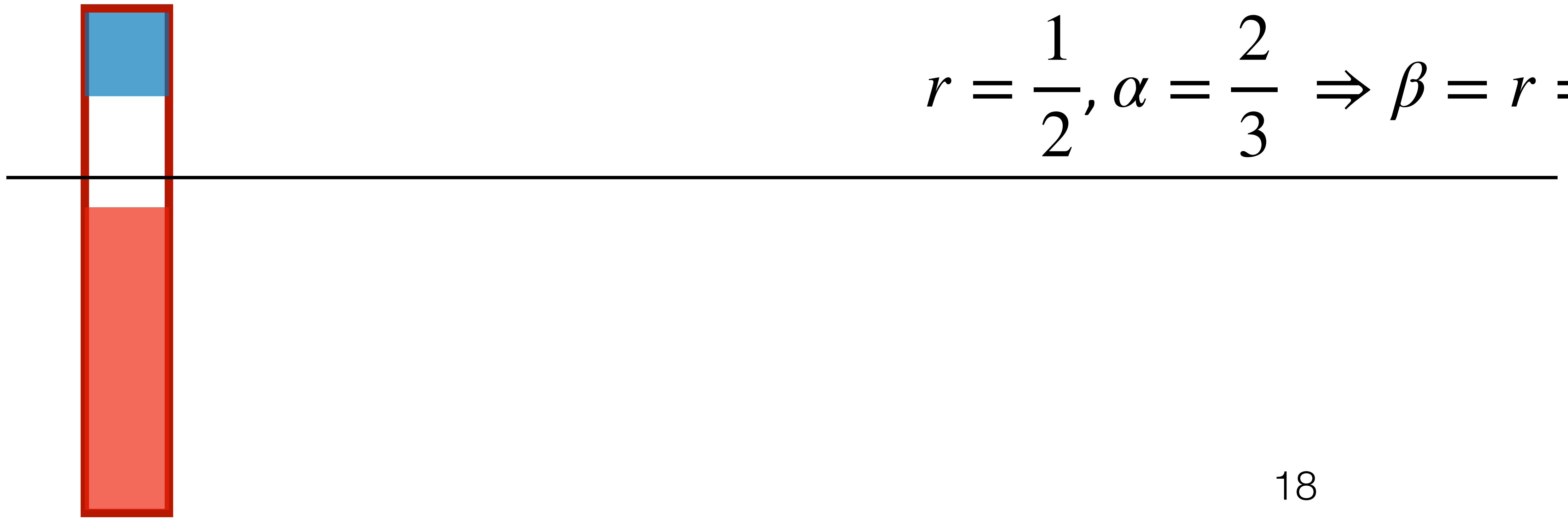


$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively

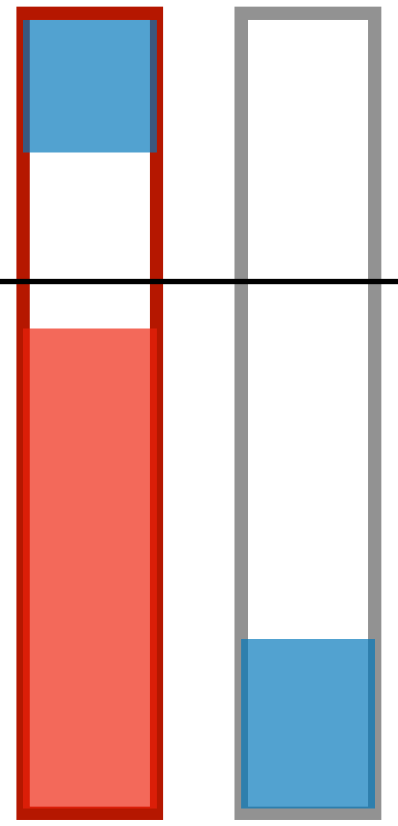


$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



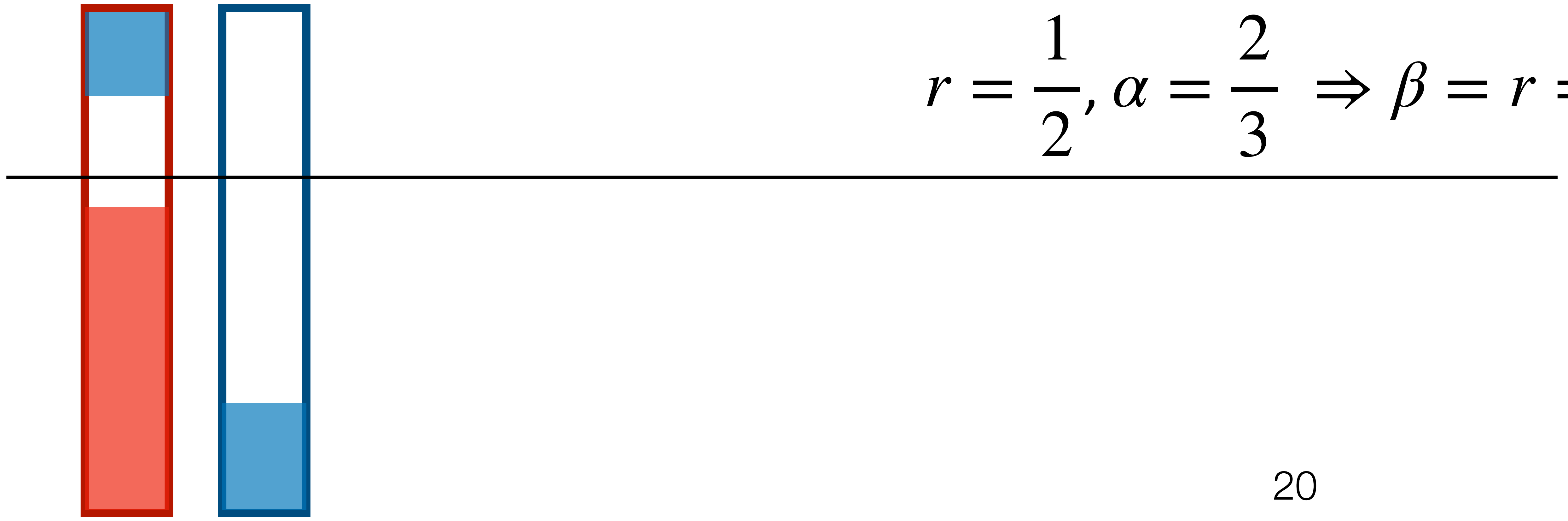
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+0}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively

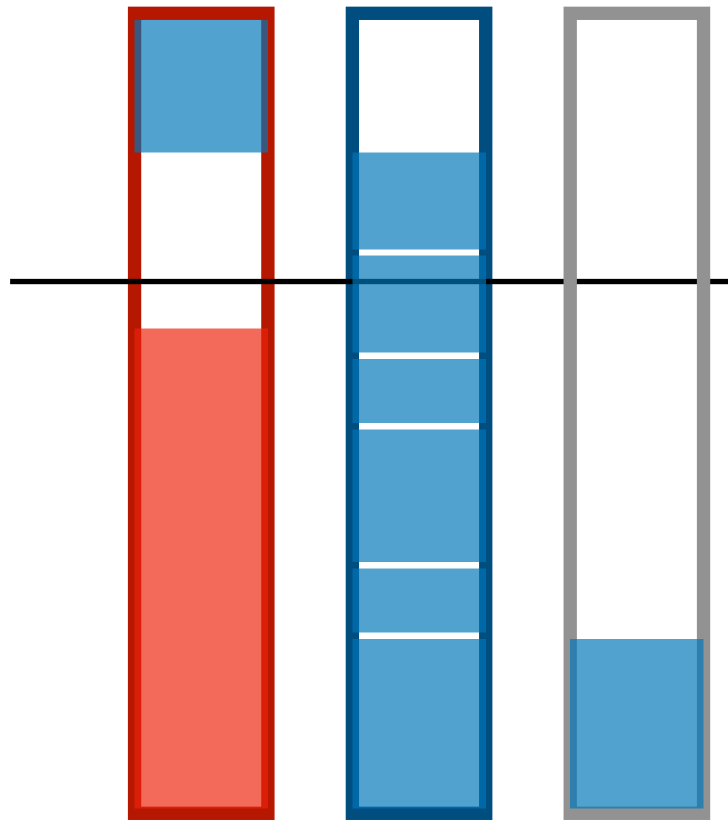


$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



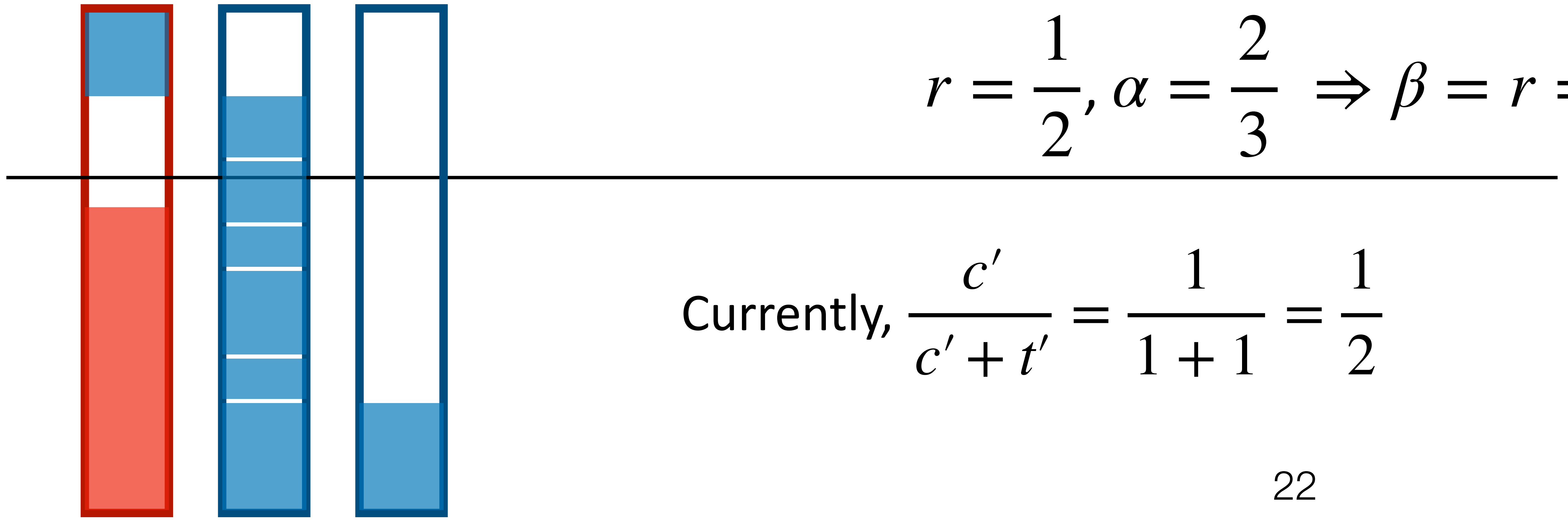
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+1} = \dfrac{1}{2}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



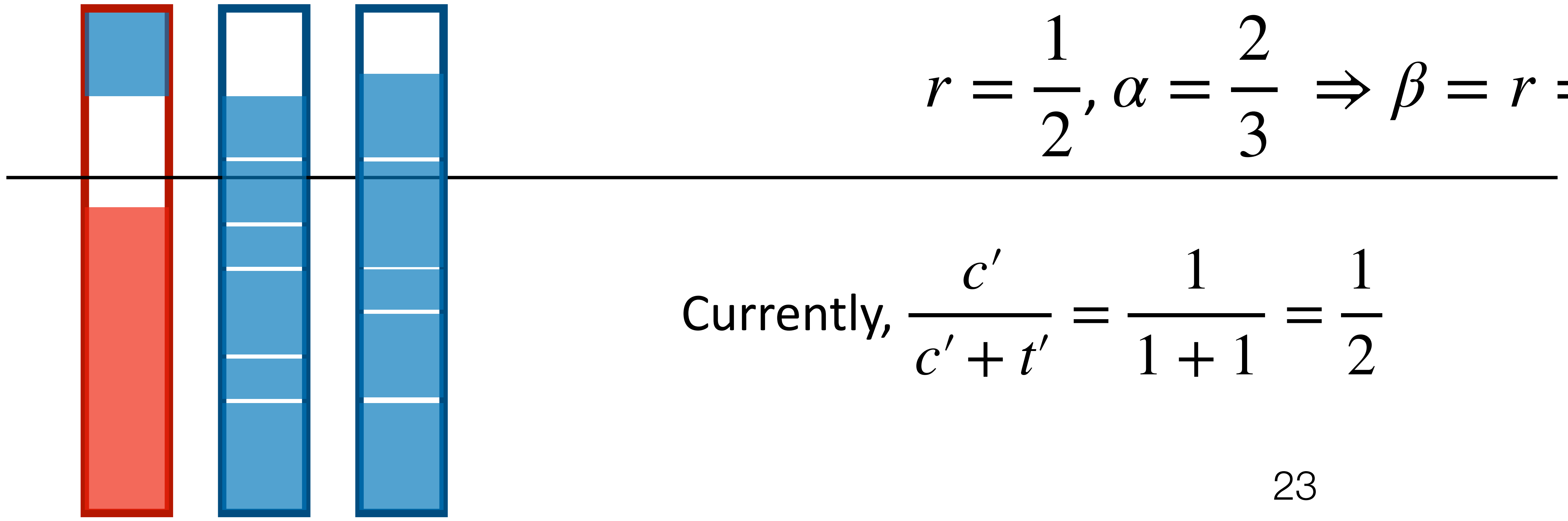
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+1} = \dfrac{1}{2}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



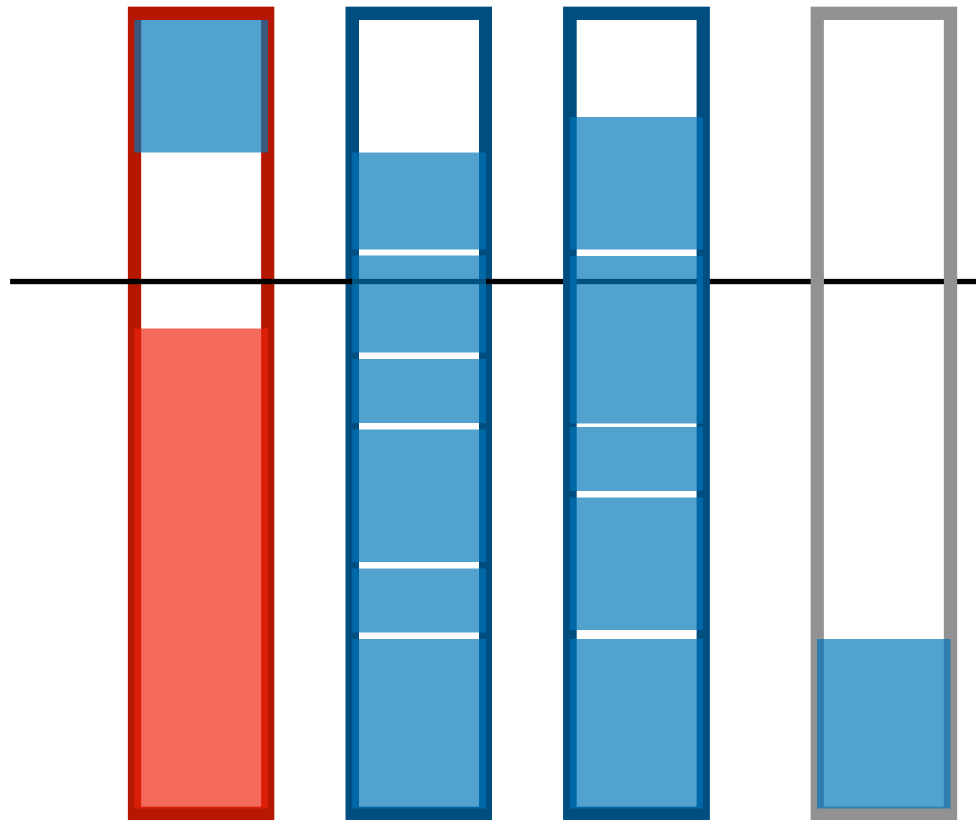
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+1} = \dfrac{1}{2}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



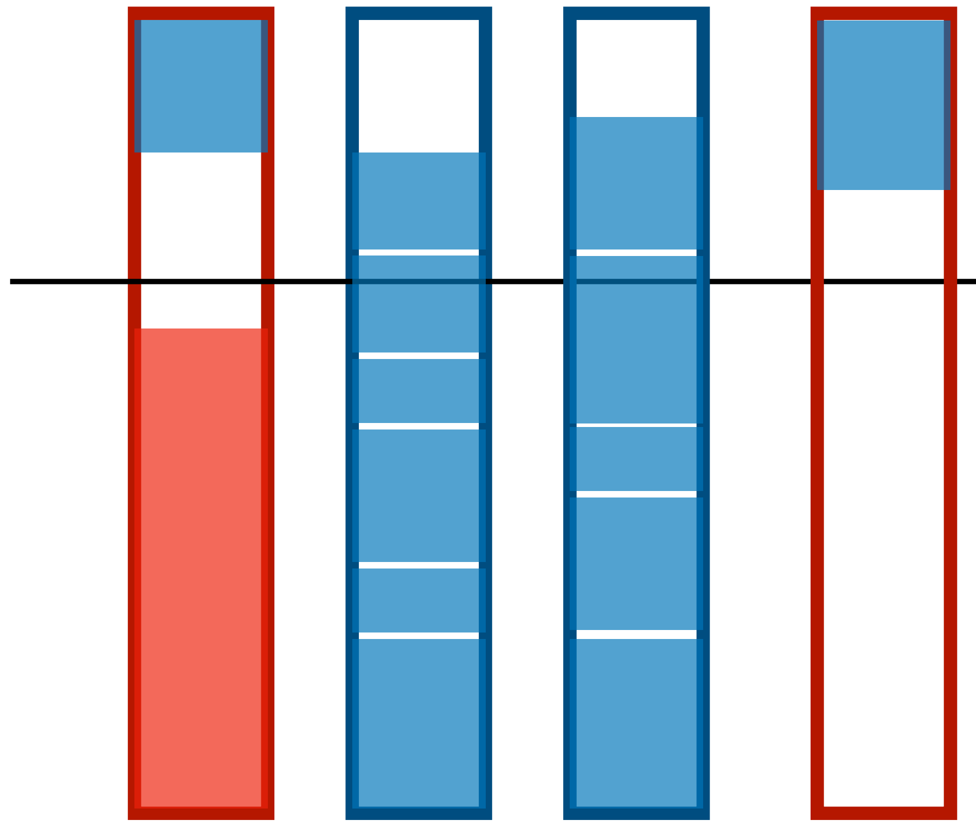
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+2} = \dfrac{1}{3}$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \frac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- `Let` $\beta = \min\{r, \alpha\}$
- `When a critical item arrives, pack it into a critical bin`
- `When a tiny item arrives, pack it into an available critical bin`
  - `If there is no available critical bin, pack it into a tiny bin`
    - `If there is no tiny bin, open a new bin B`
      - `B is a critical bin if` $\frac{c'}{c'+t'} < \beta$, `otherwise, it is a tiny bin`
- `Large items and small items are packed into large bins and small bins respectively`



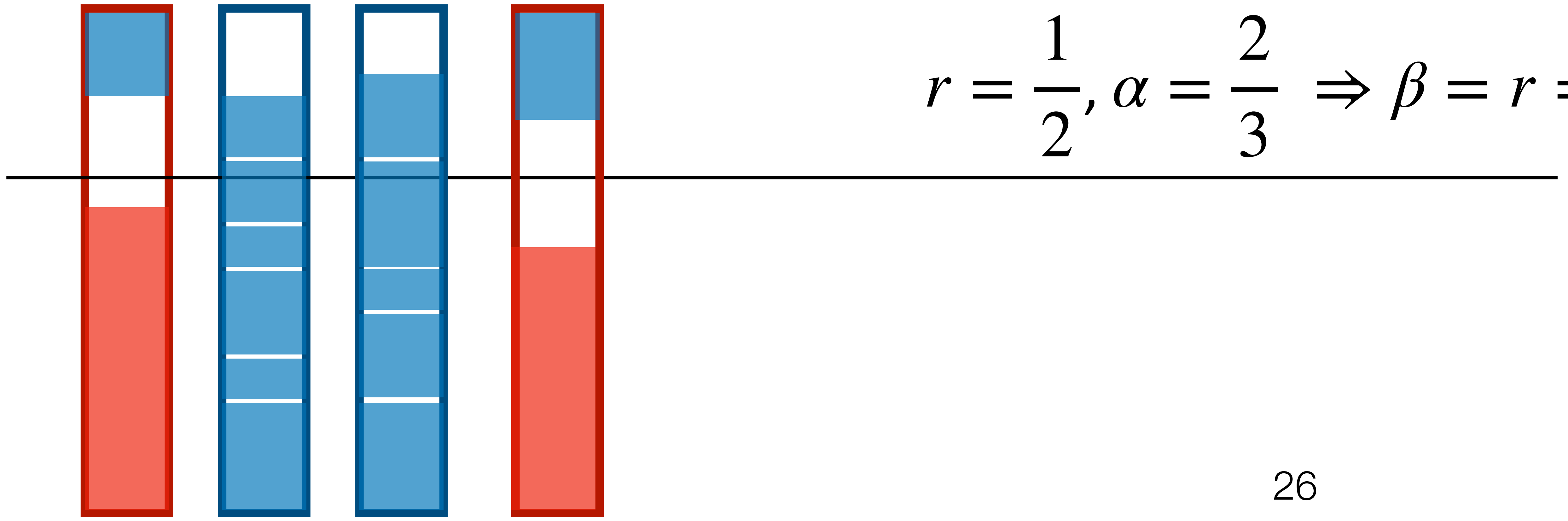
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Currently, $\frac{c'}{c'+t'} = \frac{1}{1+2} = \frac{1}{3}$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- `Let` $\beta = \min\{r, \alpha\}$
- `When a critical item arrives, pack it into a critical bin`
- `When a tiny item arrives, pack it into an available critical bin`
  - `If there is no available critical bin, pack it into a tiny bin`
    - `If there is no tiny bin, open a new bin B`
      - `B is a critical bin if` $\dfrac{c'}{c'+t'} < \beta$, `otherwise, it is a tiny bin`
- `Large items and small items are packed into large bins and small bins respectively`

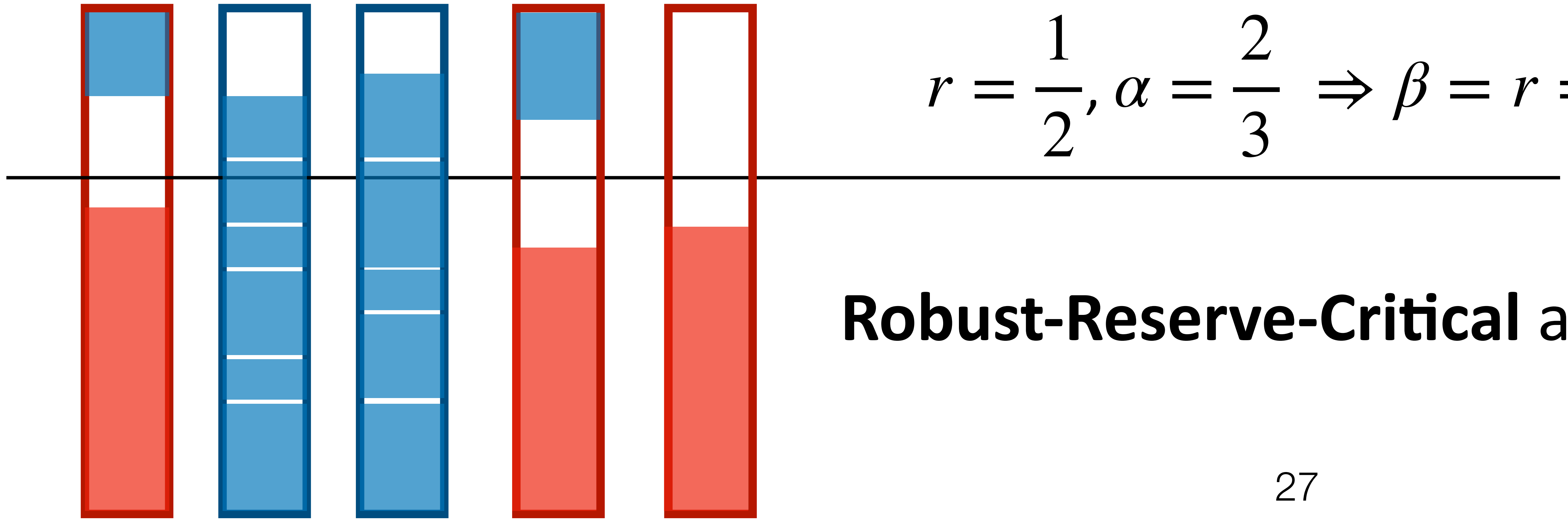


$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \frac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\frac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



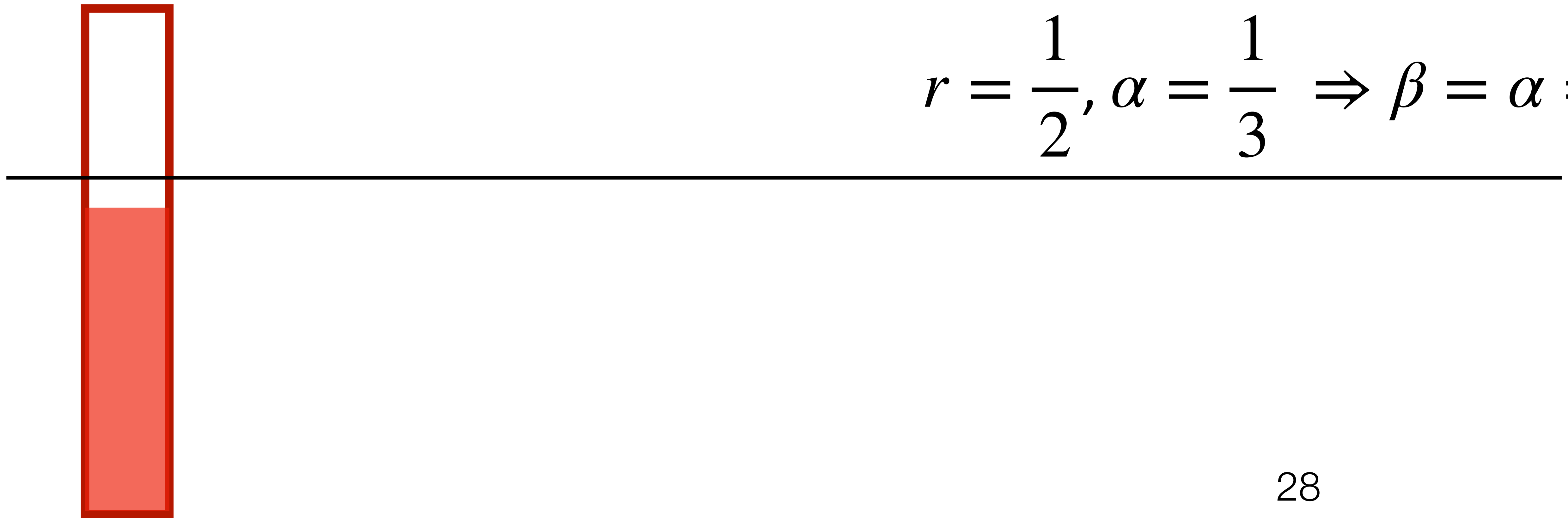
$$r = \frac{1}{2}, \alpha = \frac{2}{3} \Rightarrow \beta = r = \frac{1}{2}$$

Robust-Reserve-Critical always keep the ratio $\frac{c'}{c'+t'}$ close to β

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}$**,** α **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively

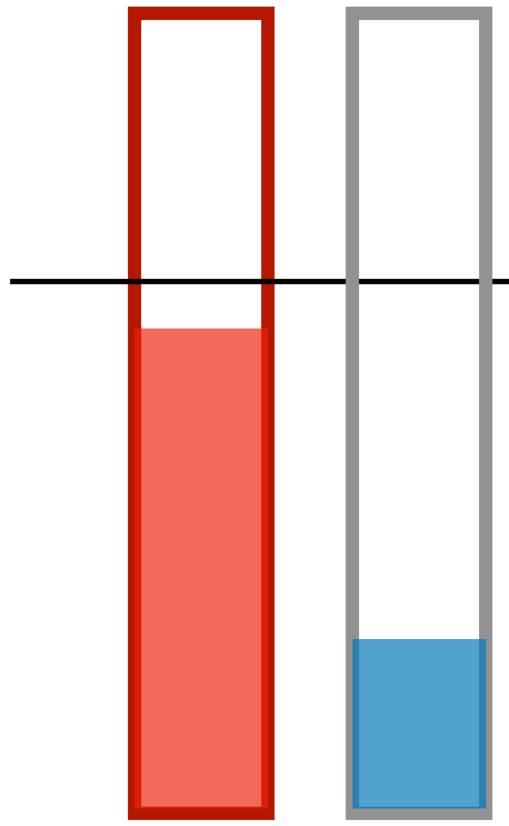


$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



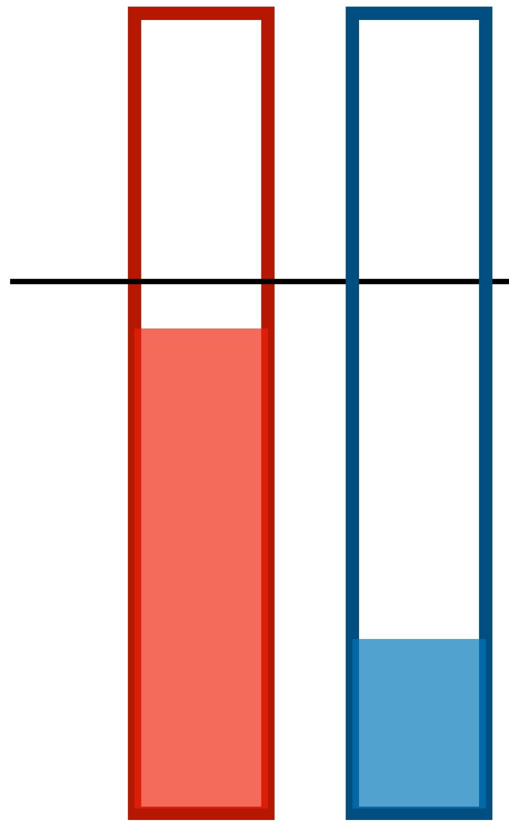
$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+0}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



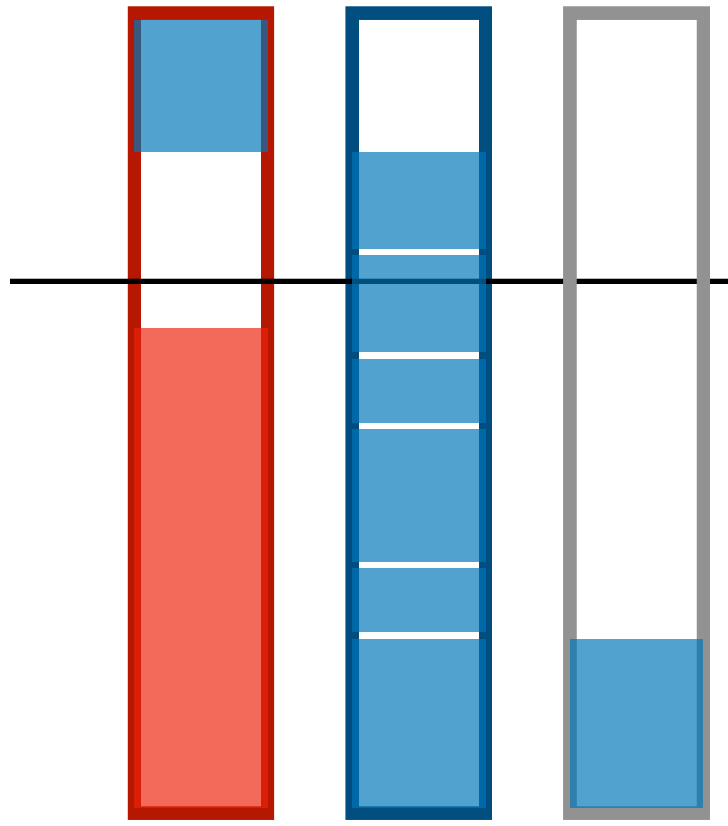
$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+0}$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- `Let` $\beta = \min\{r, \alpha\}$
- `When a critical item arrives, pack it into a critical bin`
- `When a tiny item arrives, pack it into an available critical bin`
  - `If there is no available critical bin, pack it into a tiny bin`
    - `If there is no tiny bin, open a new bin B`
      - `B is a critical bin if` $\dfrac{c'}{c'+t'} < \beta$`, otherwise, it is a tiny bin`
- `Large items and small items are packed into large bins and small bins respectively`



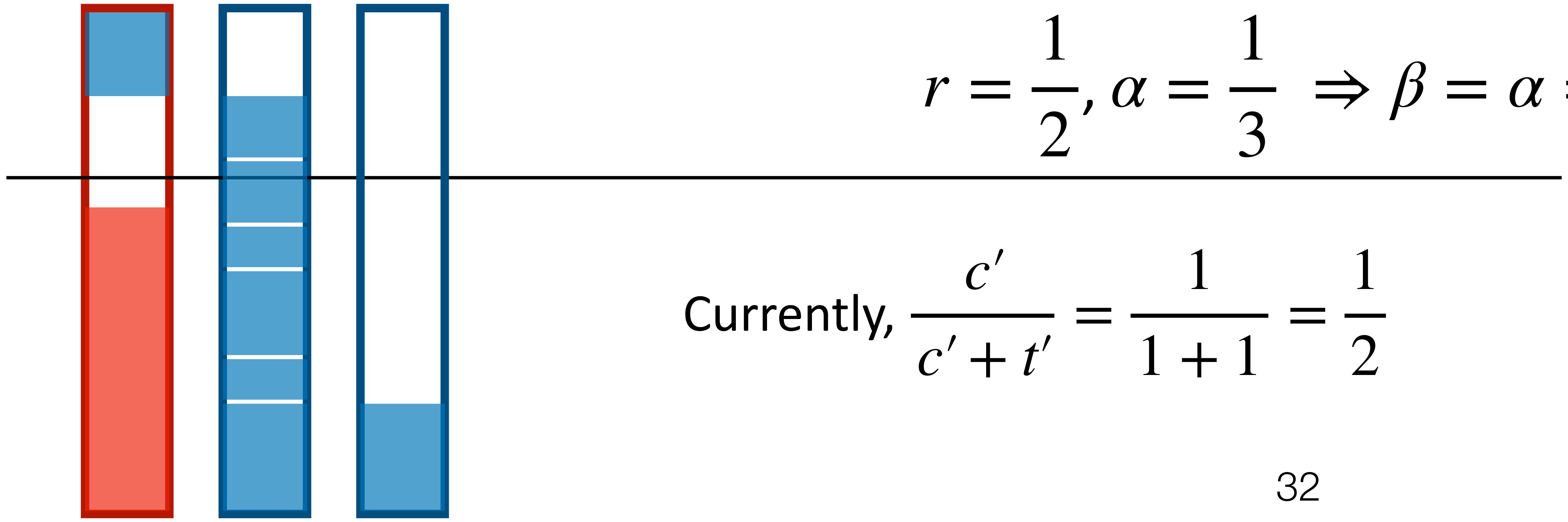
$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+1} = \dfrac{1}{2}$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



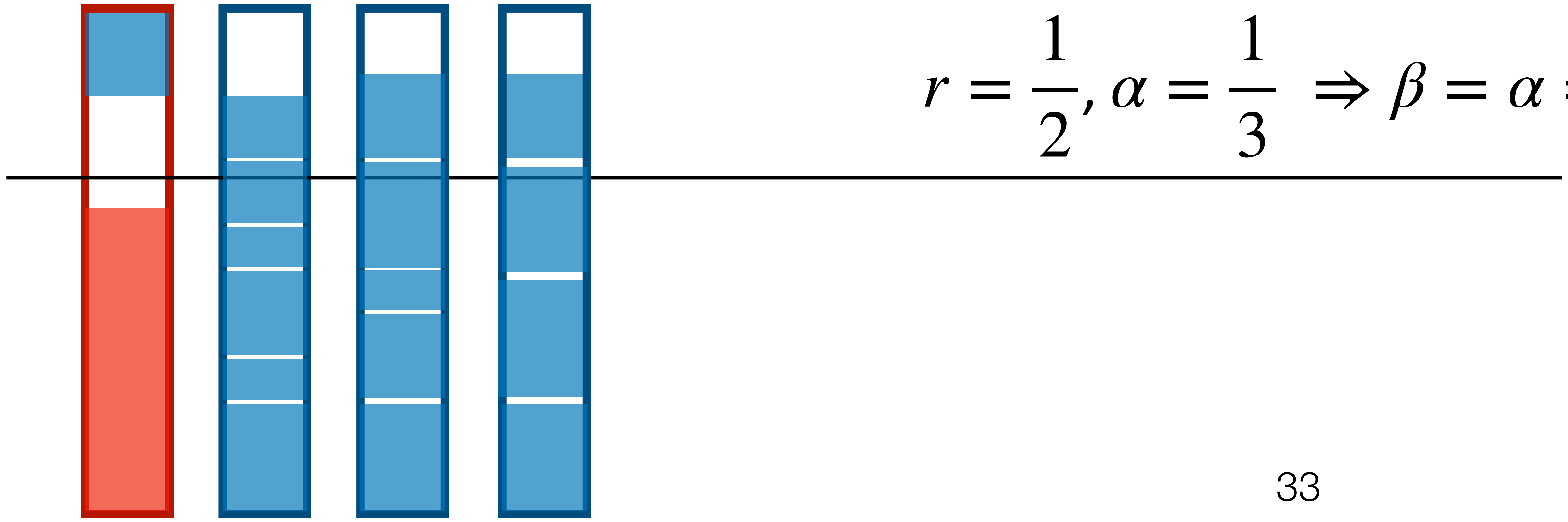
$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+1} = \dfrac{1}{2}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively

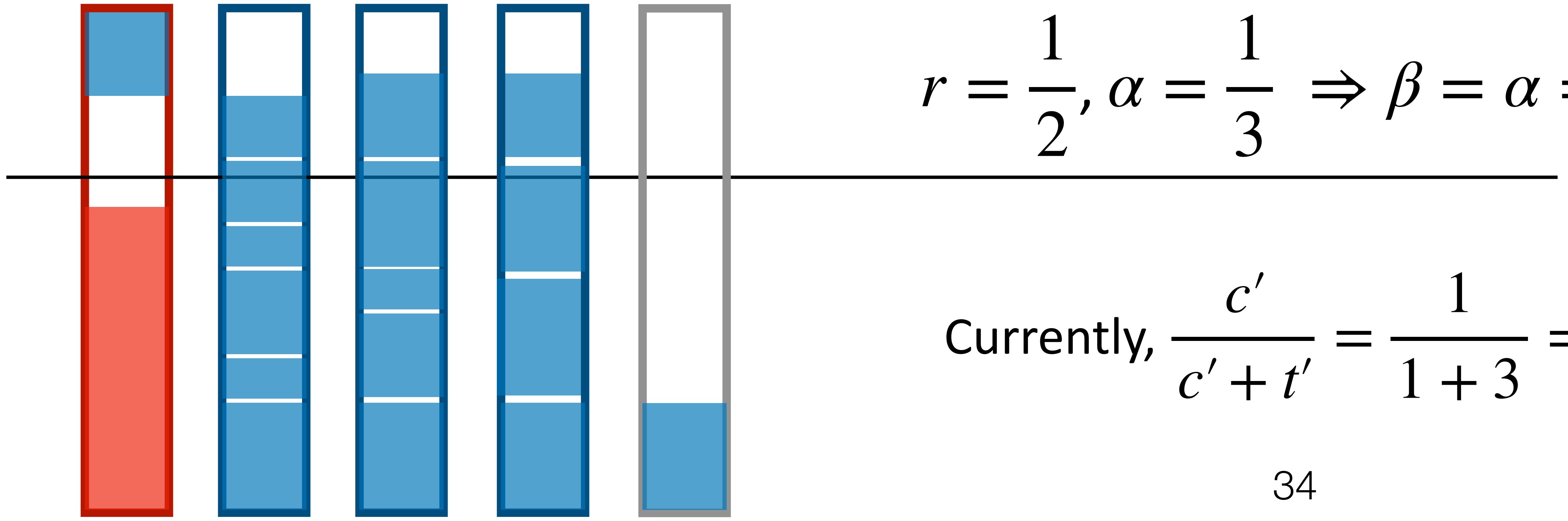


$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \frac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- `Let` $\beta = \min\{r, \alpha\}$
- `When a critical item arrives, pack it into a critical bin`
- `When a tiny item arrives, pack it into an available critical bin`
  - `If there is no available critical bin, pack it into a tiny bin`
    - `If there is no tiny bin, open a new bin B`
      - `B is a critical bin if` $\frac{c'}{c'+t'} < \beta$`, otherwise, it is a tiny bin`
- `Large items and small items are packed into large bins and small bins respectively`



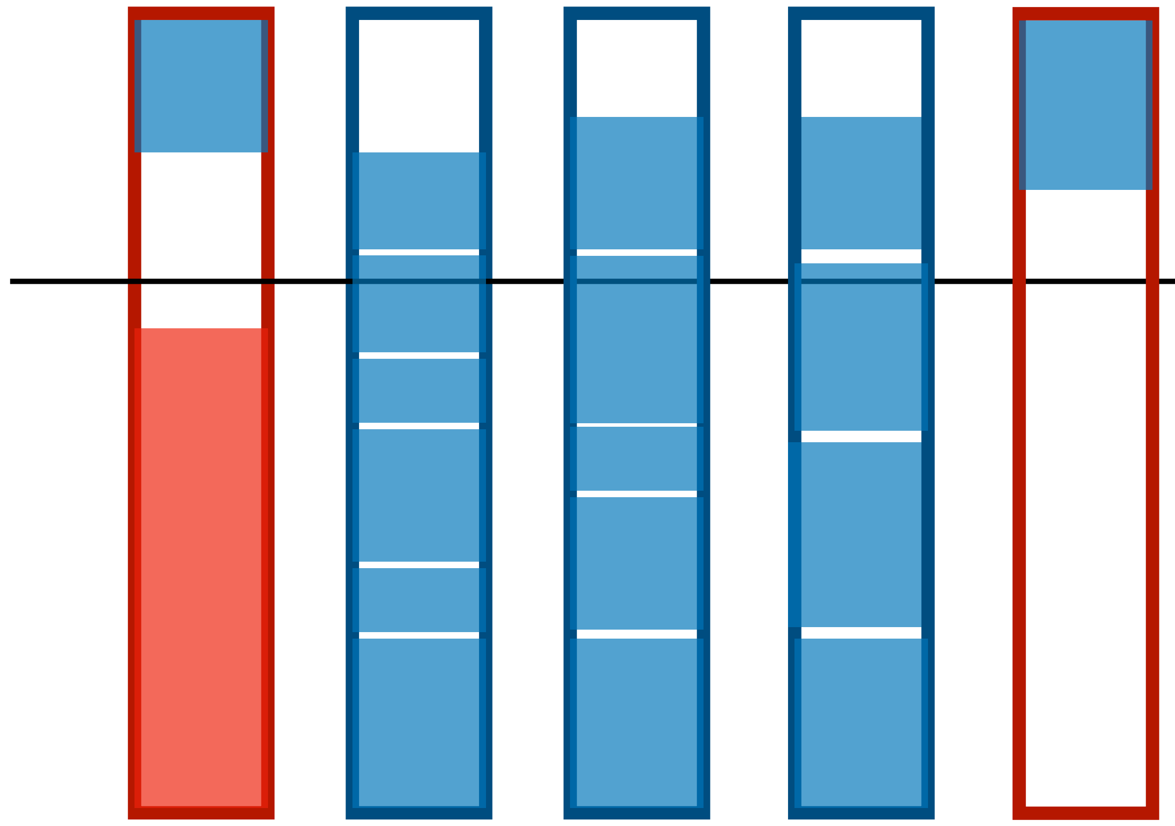
$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

Currently, $\frac{c'}{c'+t'} = \frac{1}{1+3} = \frac{1}{4}$

Robust-Reserve-Critical Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}$**,** α **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- Let $\beta = \min\{r, \alpha\}$
- When a critical item arrives, pack it into a critical bin
- When a tiny item arrives, pack it into an available critical bin
  - If there is no available critical bin, pack it into a tiny bin
    - If there is no tiny bin, open a new bin B
      - B is a critical bin if $\dfrac{c'}{c'+t'} < \beta$, otherwise, it is a tiny bin
- Large items and small items are packed into large bins and small bins respectively



$$r = \frac{1}{2}, \alpha = \frac{1}{3} \Rightarrow \beta = \alpha = \frac{1}{3}$$

Currently, $\dfrac{c'}{c'+t'} = \dfrac{1}{1+3} = \dfrac{1}{4}$

`Robust-Reserve-Critical` Algorithm

Robust-Reserve-Critical($r = \dfrac{c}{c+t}, \alpha$ **)** // c : number of critical bins, t : number of tiny bins, $\alpha \in [0,1]$

- `Let` $\beta = \min\{r, \alpha\}$
- `When a critical item arrives, pack it into a critical bin`
- `When a tiny item arrives, pack it into an available critical bin`
  - `If there is no available critical bin, pack it into a tiny bin`
    - `If there is no tiny bin, open a new bin B`
      - `B is a critical bin if` $\dfrac{c'}{c'+t'} < \beta$, `otherwise, it is a tiny bin`
- `Large items and small items are packed into large bins and small bins respectively`

Prediction: the number of critical bins and the number of tiny bins

$$\frac{c}{c+t}$$

Every critical bin has one critical item

We my open too many critical bins

`Robust-Reserve-Critical` Algorithm

- Lemma: If every critical bin has one critical item, **Robust-Reserve-Critical** is at most $1.5 + \dfrac{1-\beta}{4-3\beta}$-competitive

By the algorithm, we can make sure that each bin is full enough

- Lemma: If there are some critical bins have no critical item, **Robust-Reserve-Critical** is at most $1.5 + \dfrac{9\beta}{8-6\beta}$-competitive

The critical bin without a critical item must be opened because of a tiny item
$\Rightarrow$ at the time the tiny item arrives, the number of critical bins is low

`Robust-Reserve-Critical` Algorithm

- Theorem: If the prediction on $r = \frac{c}{c+t}$ is correct, **Robust-Reserve-Critical** is at most $1.5 + \frac{1-\alpha}{4-3\alpha}$-competitive. Otherwise, **Robust-Reserve-Critical** is at most $1.5 + \max\{\frac{1}{4}, \frac{9\alpha}{8-6\alpha}\}$-competitive.

What Happened

- `Robust-Reserve-Critical` Algorithm uses the `Reserve-Critical` Algorithm
  - Trust parameter $\alpha \in [0,1]$
    - Robustness: $1.5 + \frac{1-\alpha}{4-3\alpha}$
    - Consistency: $1.5 + \max\{\frac{1}{4}, \frac{9\alpha}{8-6\alpha}\}$