# Cookies 解説

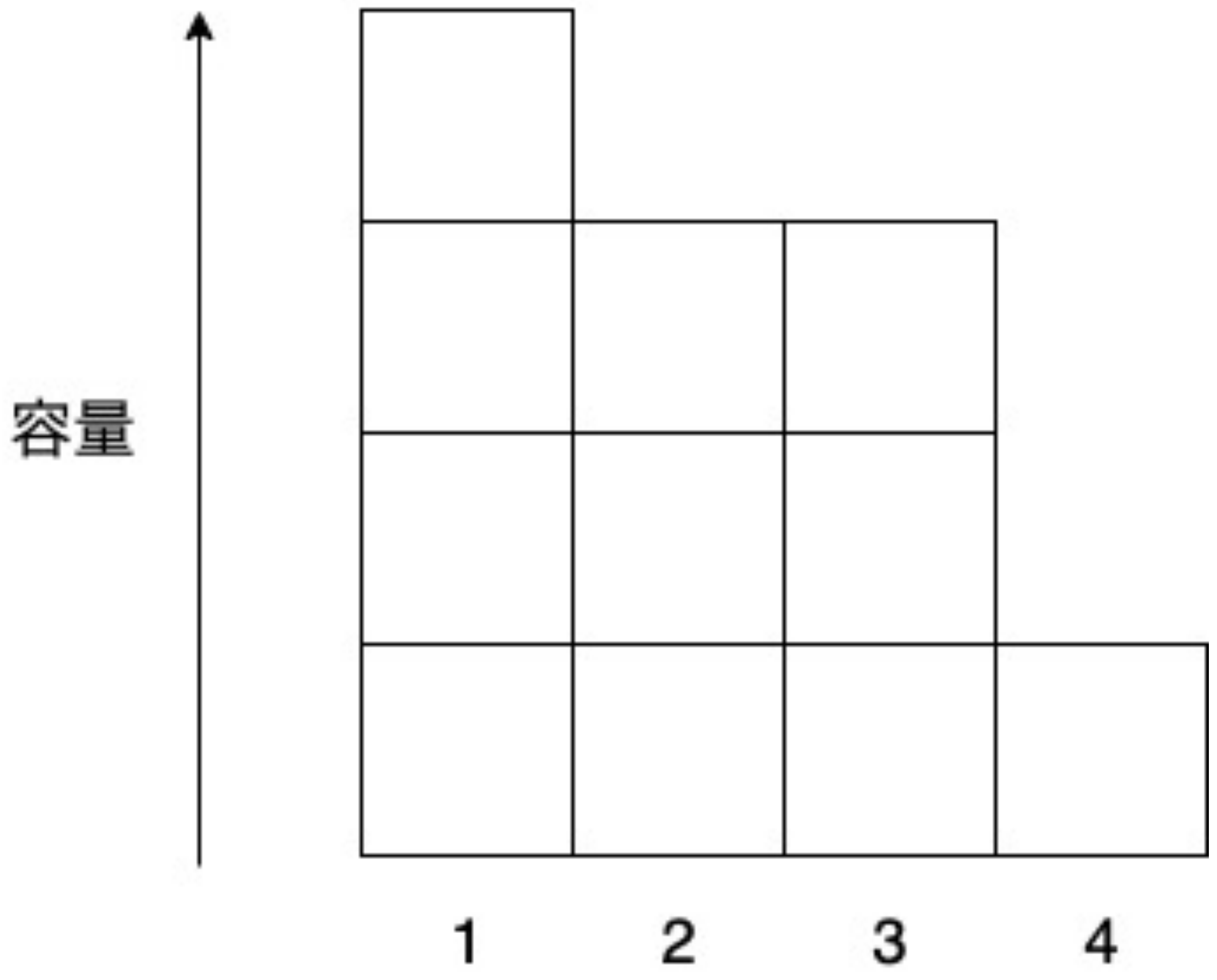
解説 : yuto1115

# 問題概要

- $N$ 種類のクッキーがあり、$i$ 種類目は $A_i$ 枚ある
- 今から全てのクッキーを何箱かに分けて箱詰めするが、
  - 同じ箱の中に同じ種類のクッキーがあってはいけない
  - それぞれの箱に入れるクッキーの枚数は $B_1, B_2, \ldots, B_M$ のいずれか
- 箱詰めは可能か？可能な場合、使う箱の数が最小になる割り当てを $1$ つ構築

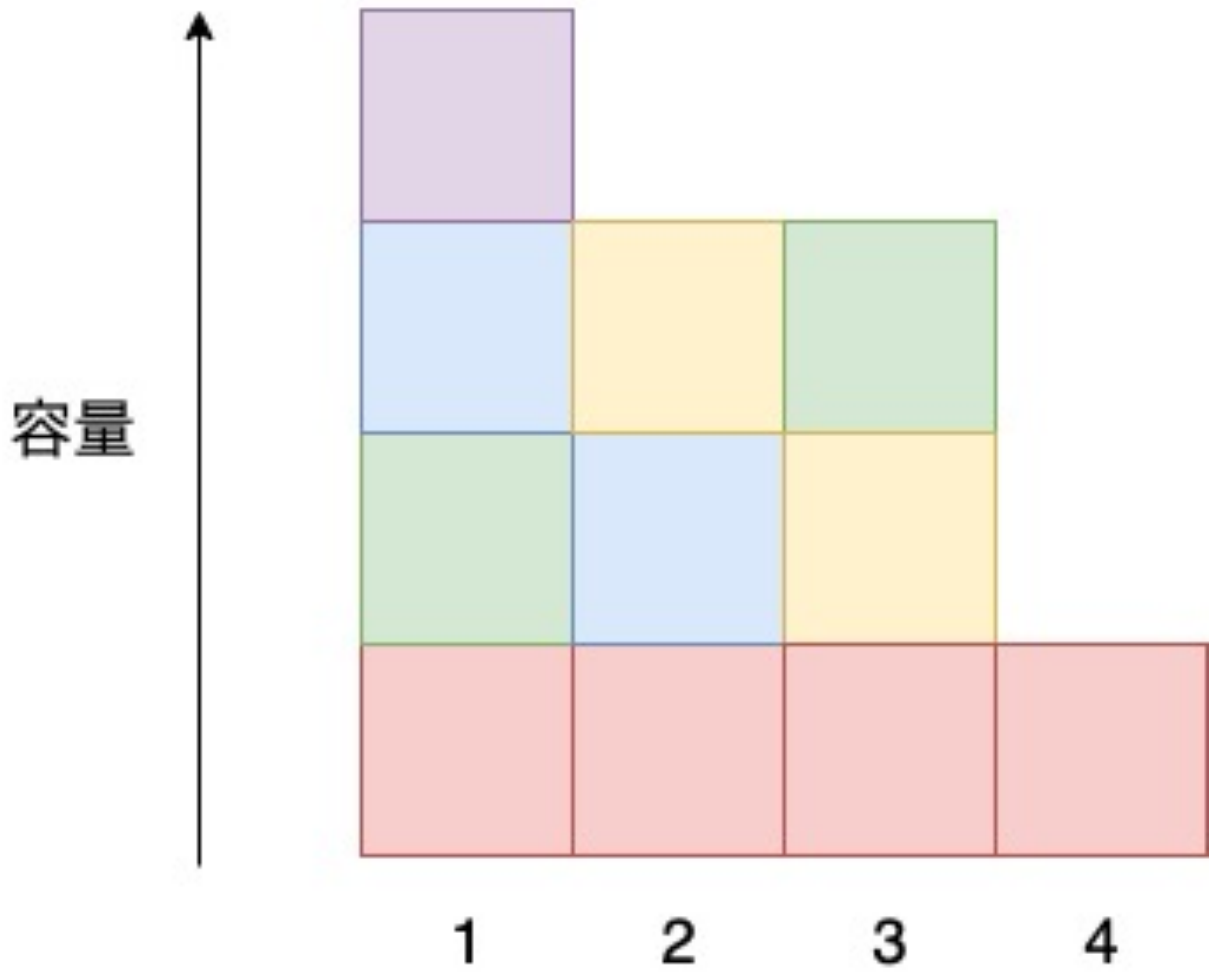
# 例

$A = (4,2,2,2,1),\ B = (1,3,4)$



# 例

$A = (4,2,2,2,1),\ B = (1,3,4)$



# 小課題 1 ($N \leqq 500, A_i = 1$)

- クッキーの種類の重複を考えなくていい
- 必要十分条件は $\displaystyle\sum_i A_i = \sum_j B_j$
- 部分和問題なので、DP で $O(N^2)$

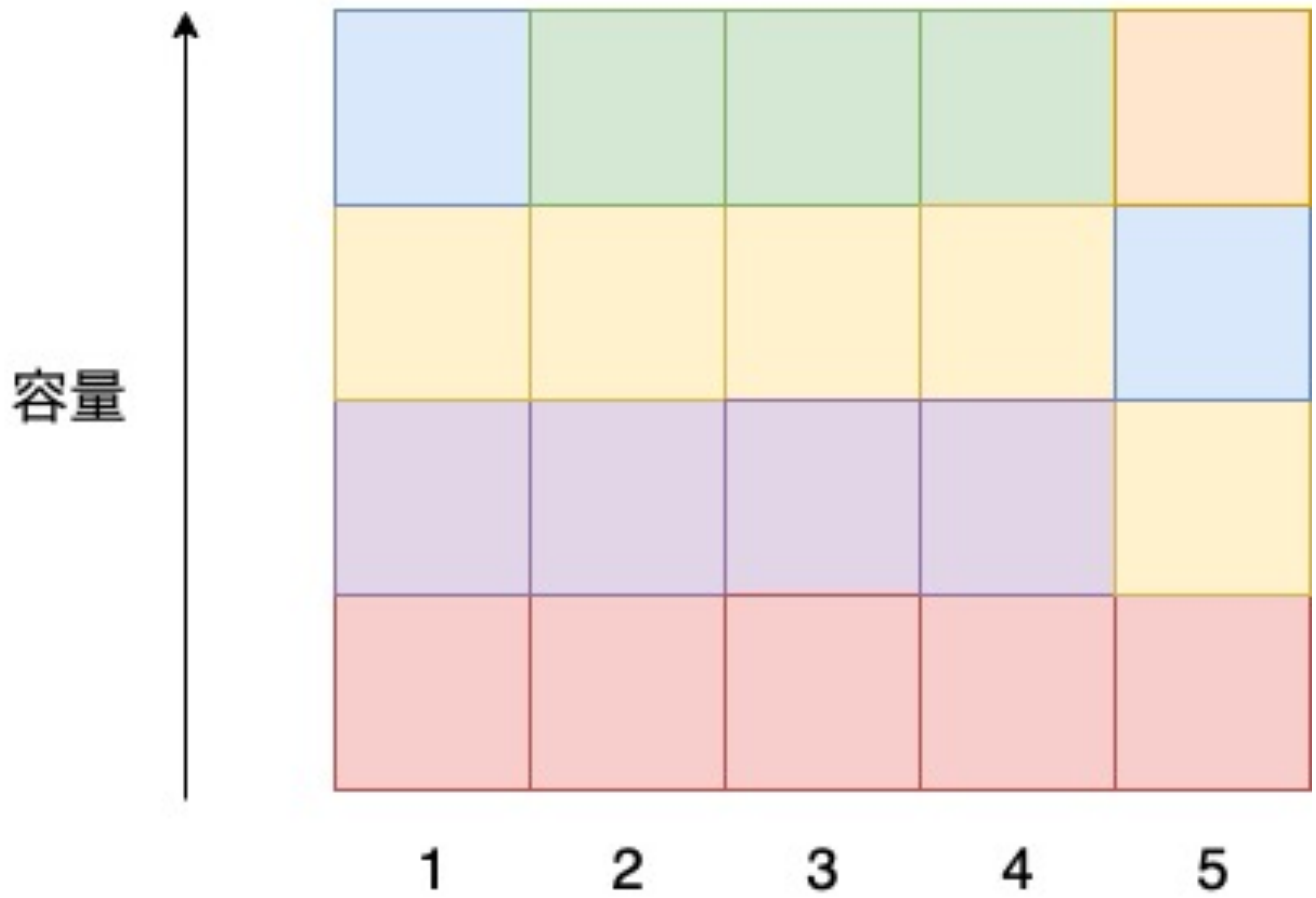
$\boxed{\text{ここまでで 6 点}}$

# 小課題 2 ($N \leqq 500,\ M = 1$)

- 以下、$S = \sum_i A_i$ とおく
- $S$ が $B_1$ で割り切れない場合、明らかに割り当て不可能
- そうでないならば、使う箱の数は一意に定まる

# 小課題 2 ($N \leqq 500,\ M = 1$)

- どんな時に割り当て可能か考えよう
  - $\max A_i$ 箱以上あることは必要条件
  - 逆にこれを満たせば ...?


容量
1
2
3
4
5

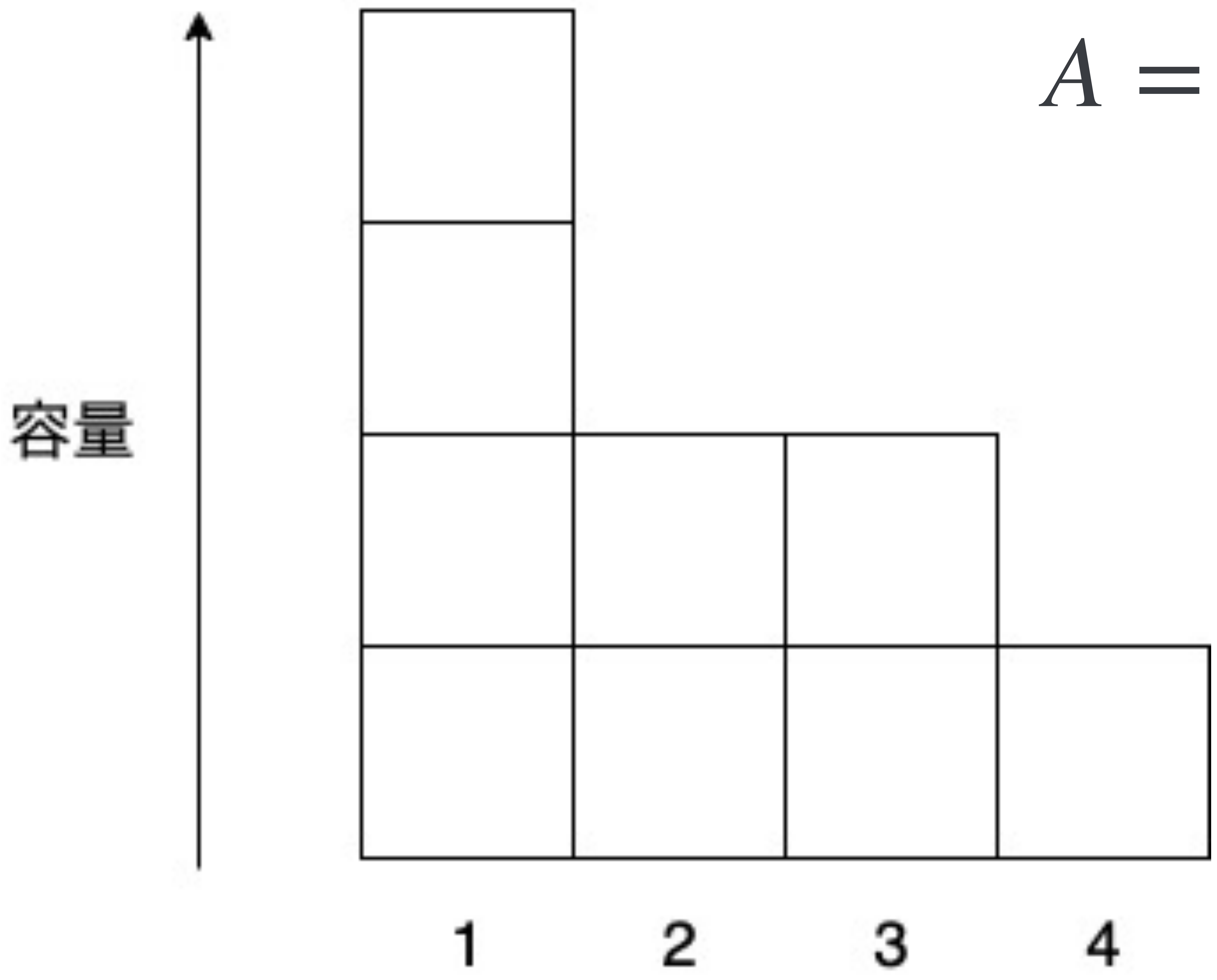
# 小課題 2 ($N \leqq 500, M = 1$)

- よって、条件を $O(1)$ で判定可能
- 構築はさっきの図のようにやればいい

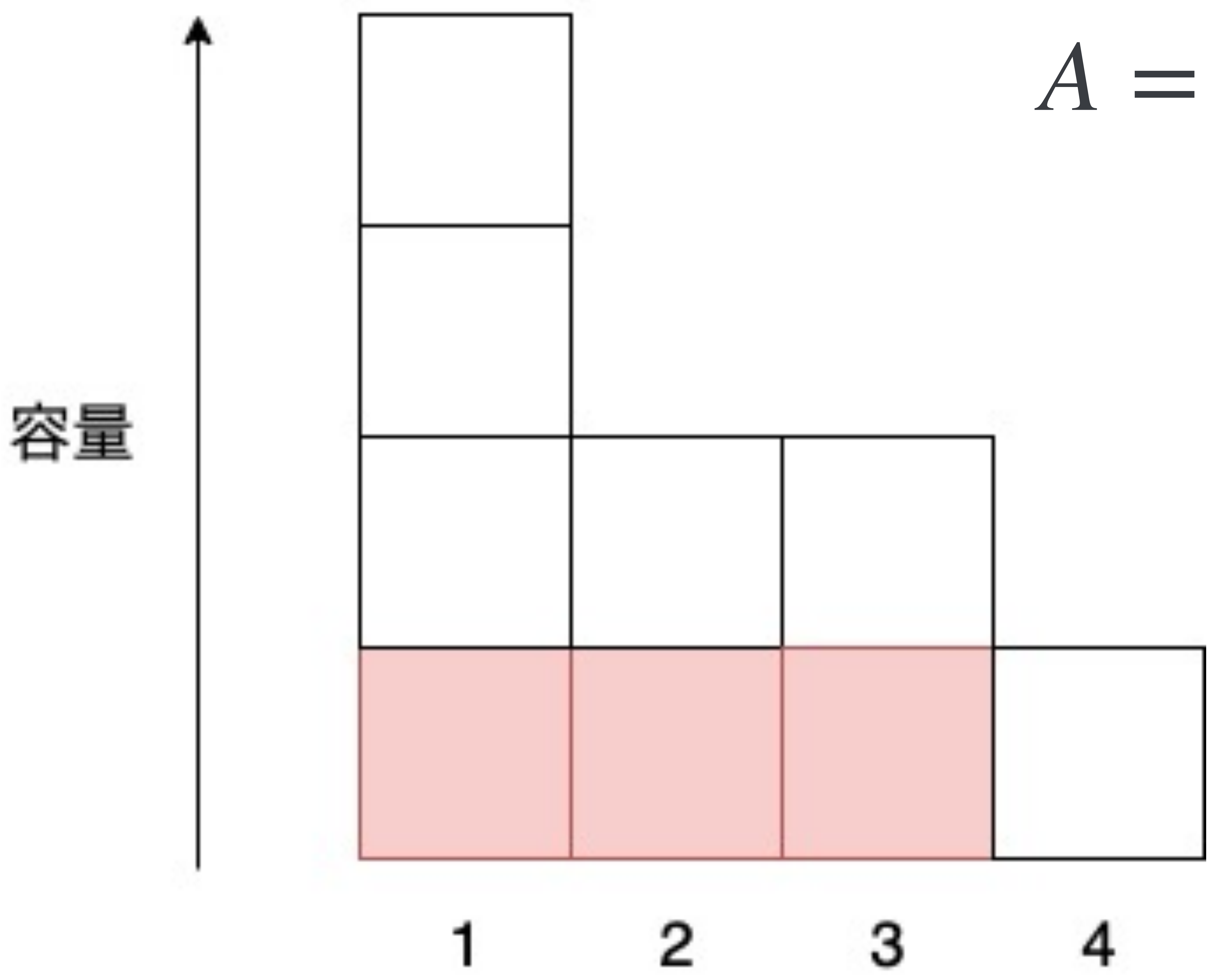
ここまでで 13 点

# 考察

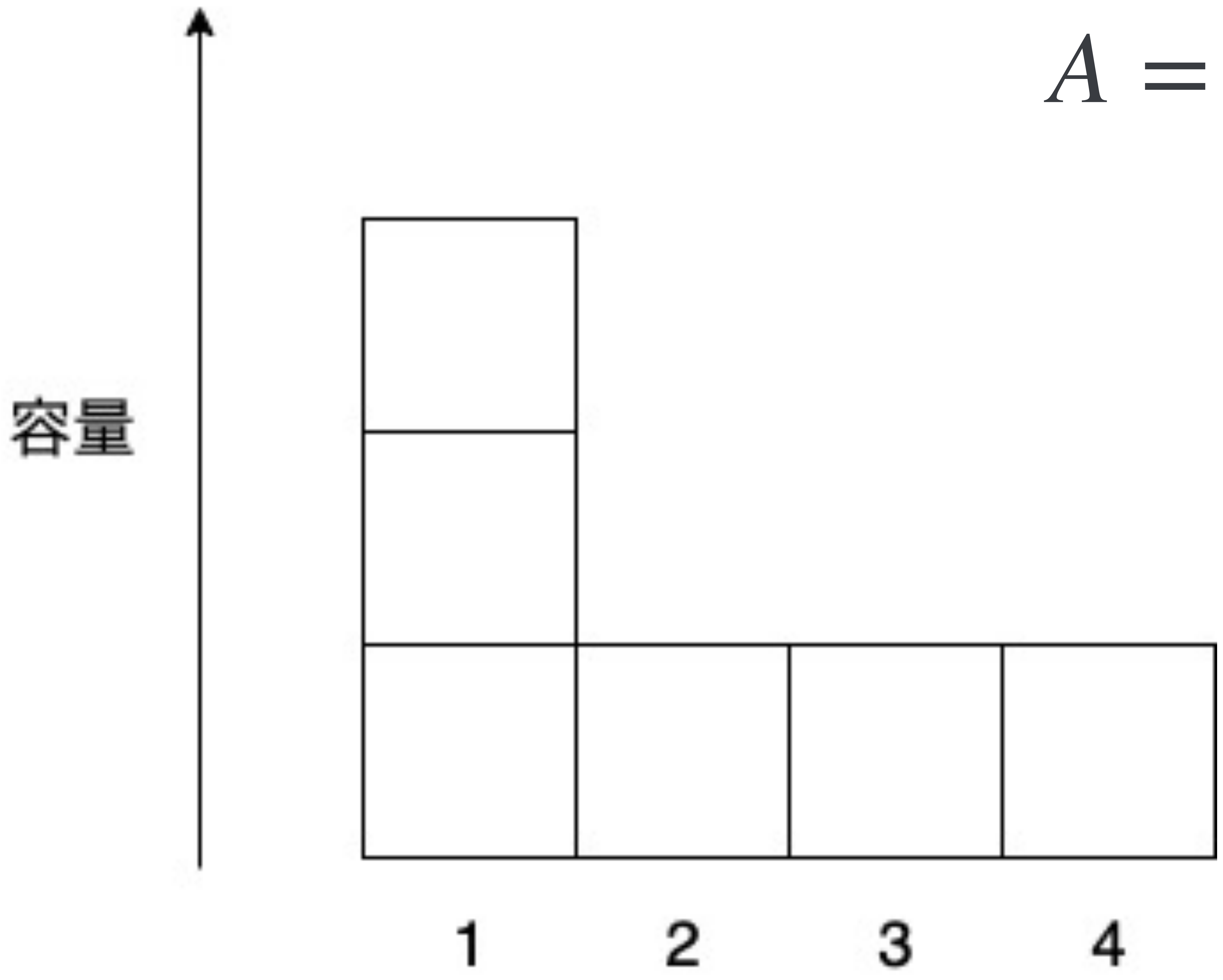
- 一般の場合で、**使う箱が既に決まっているとき、**割り当てを $1$ つ構築する or 割り当て不可能と判定する方法を考えたい (箱の容量の和が $S$ と一致していることは前提とする)
- 次のような貪欲法を考える
  - $i = 1,2,\ldots,N$ について以下を行う
    - 残り容量（元々の容量 − 既に入れたクッキーの枚数）が大きい順に箱を $A_i$ 個選んで、それぞれの箱に種類 $i$ のクッキーを $1$ 枚ずつ入れる
    - 残り容量が正の箱が $A_i$ 箱未満なら、割り当て不可能と判定する




$$A = (3,2,2,2)$$

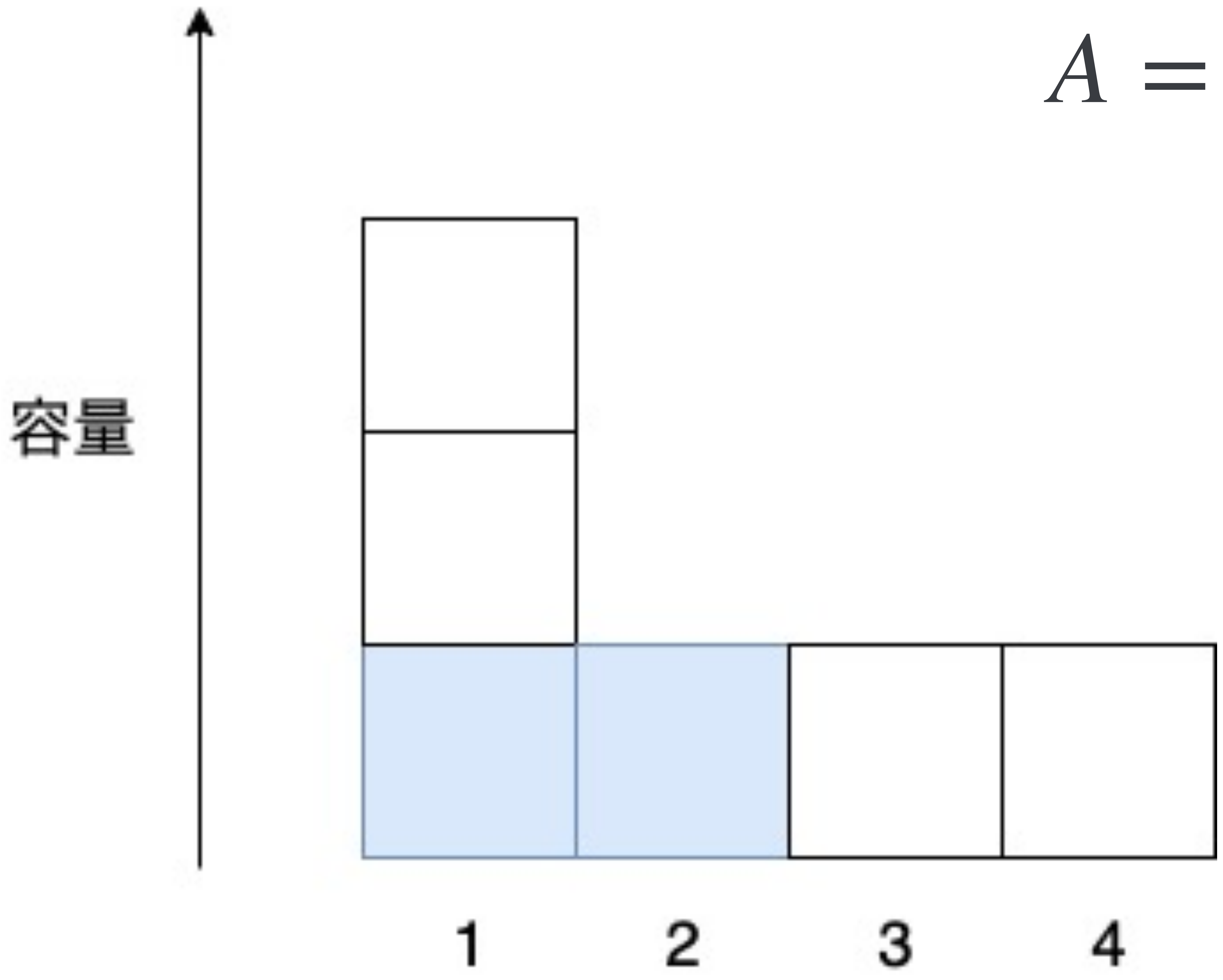


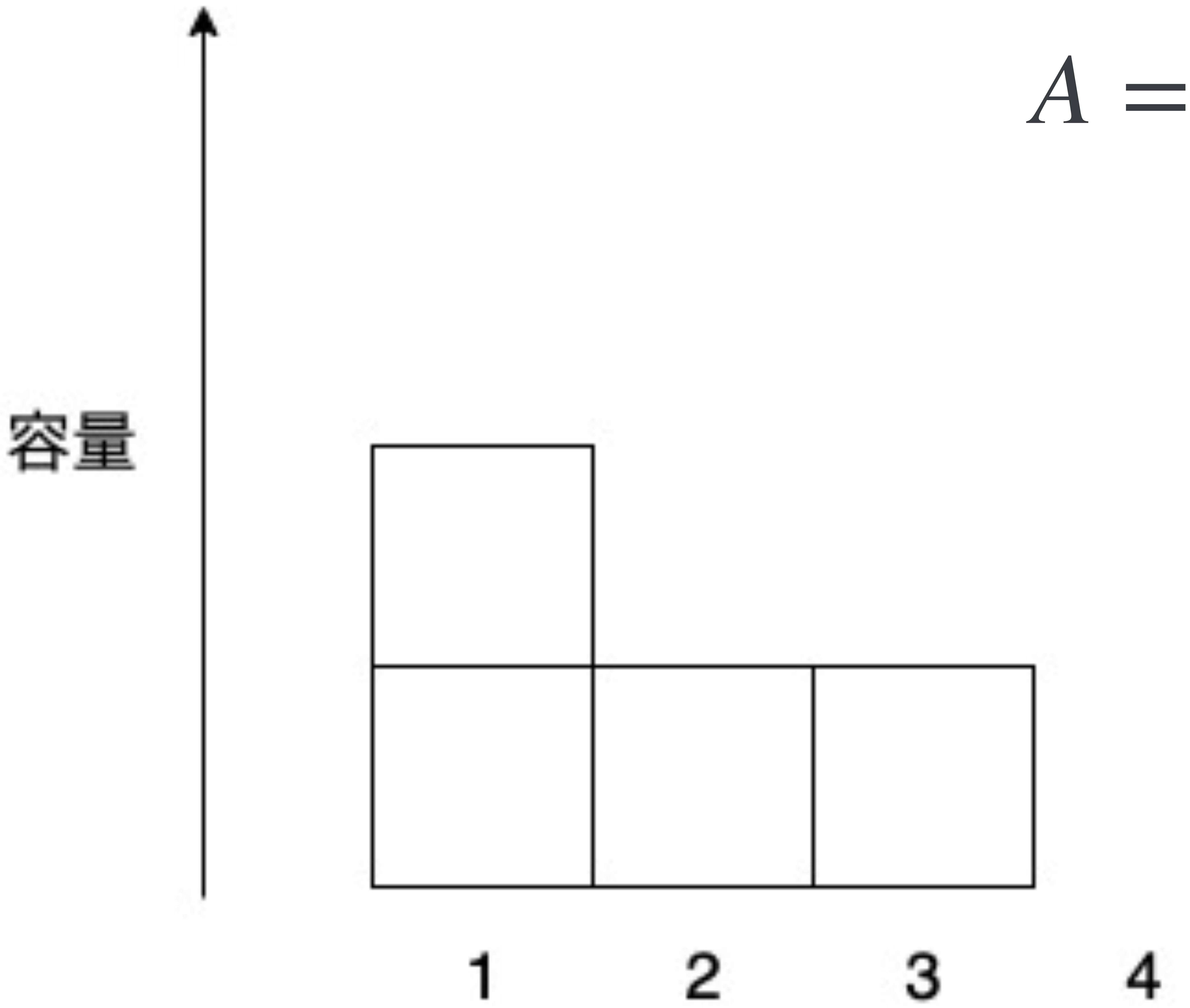

$$A = (3,2,2,2)$$



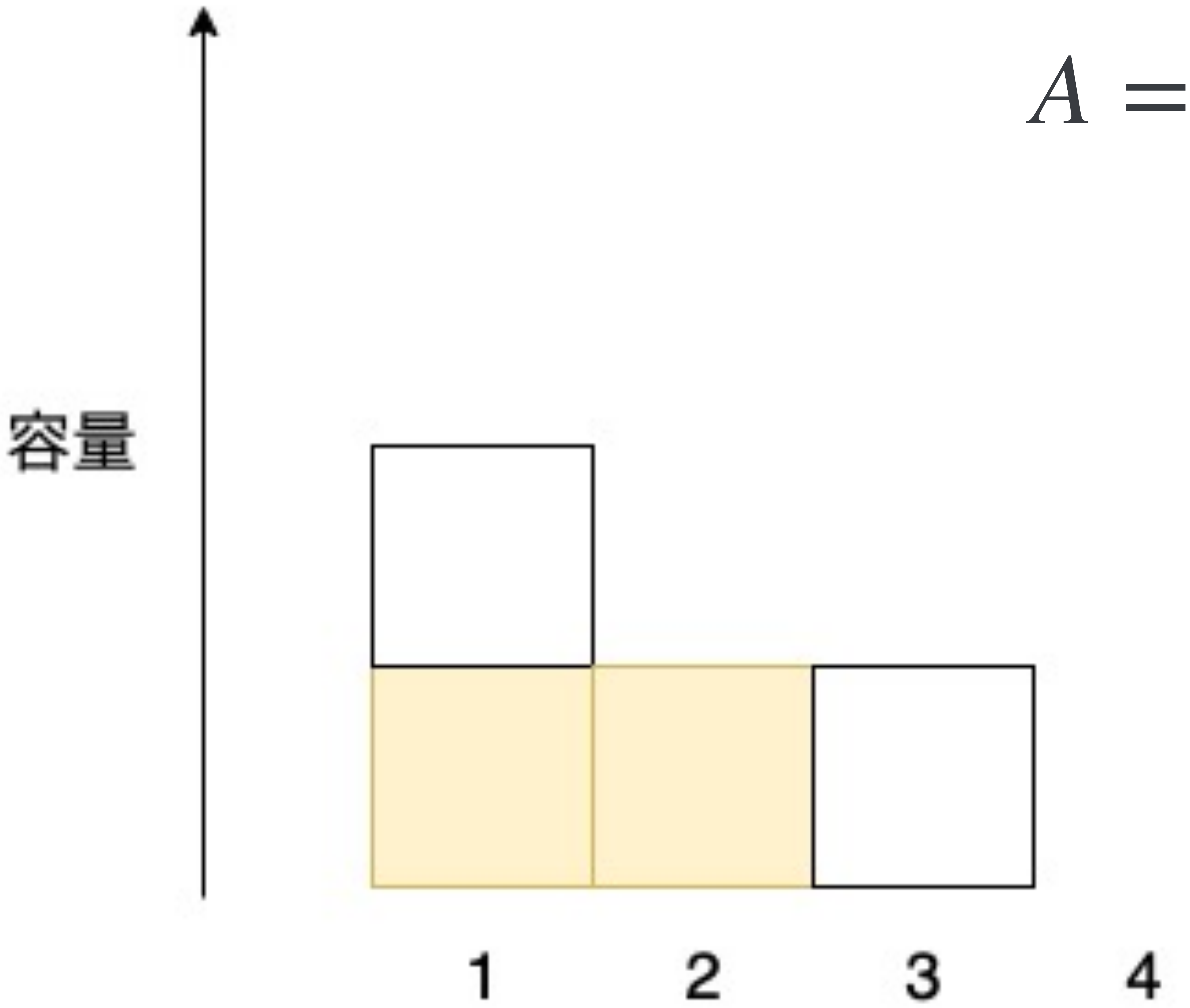

$$A = (3,2,2,2)$$

$$A = (3,2,2,2)$$



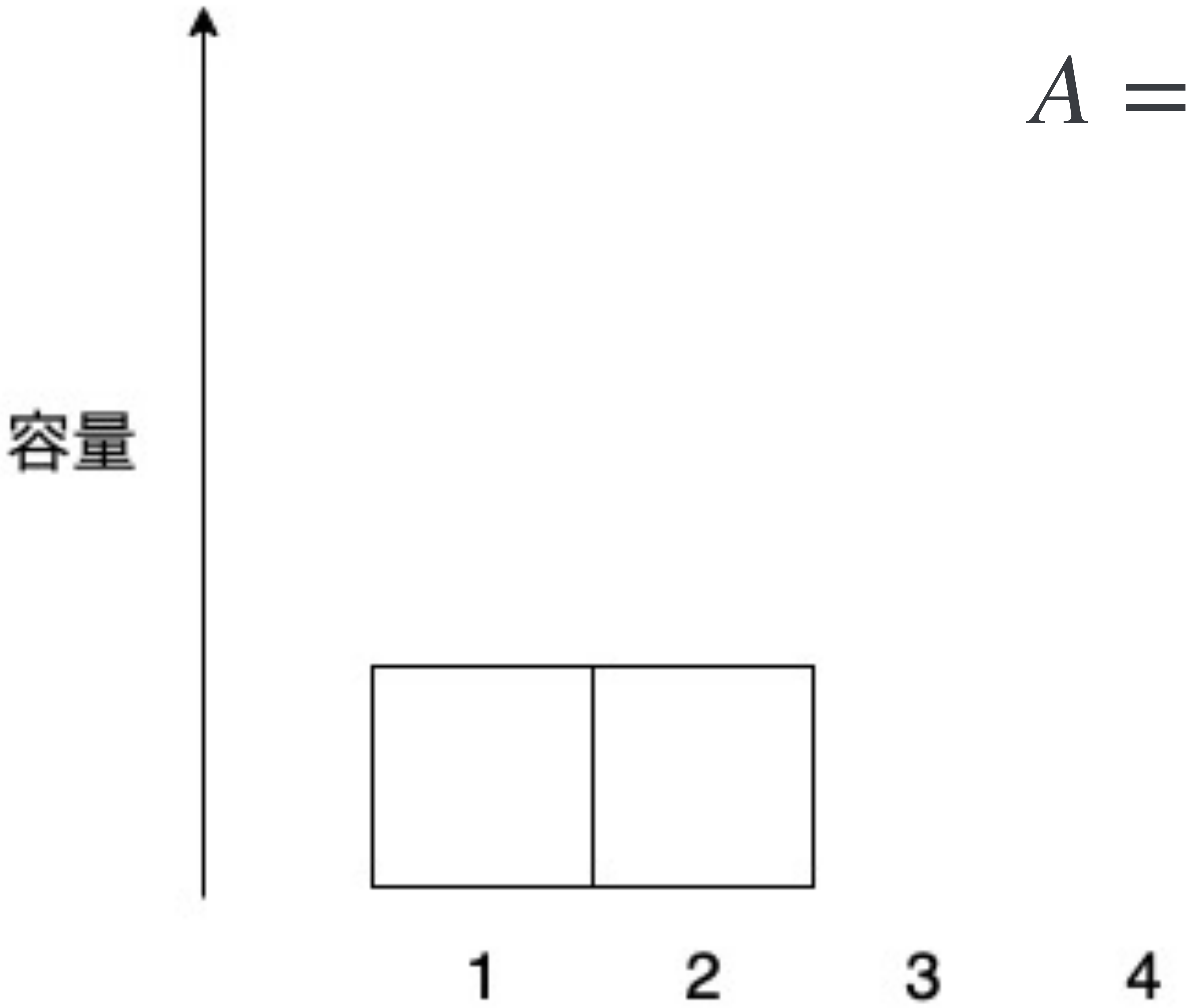



$$A = (3,2,2,2)$$

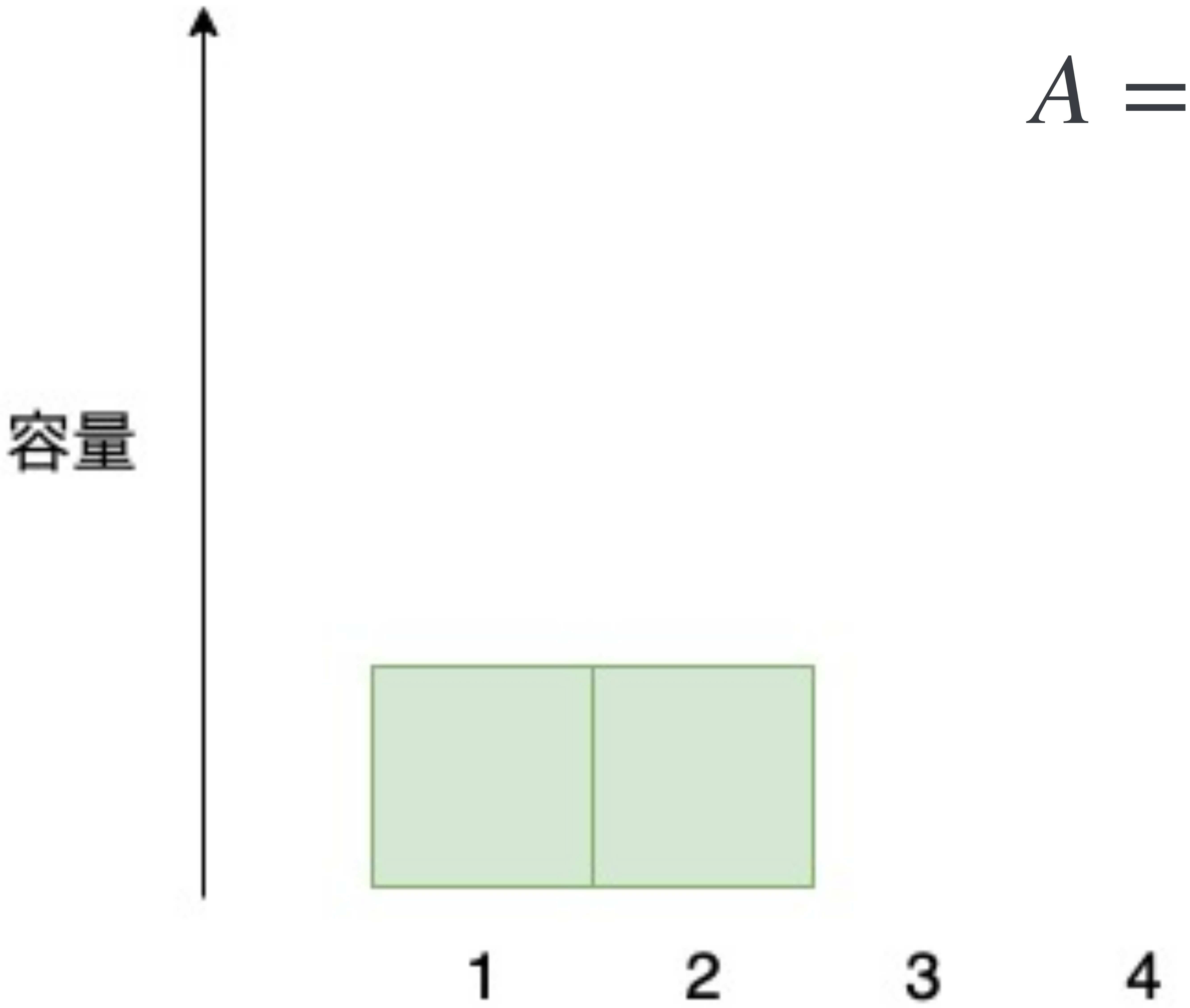



$$A = (3,2,2,2)$$

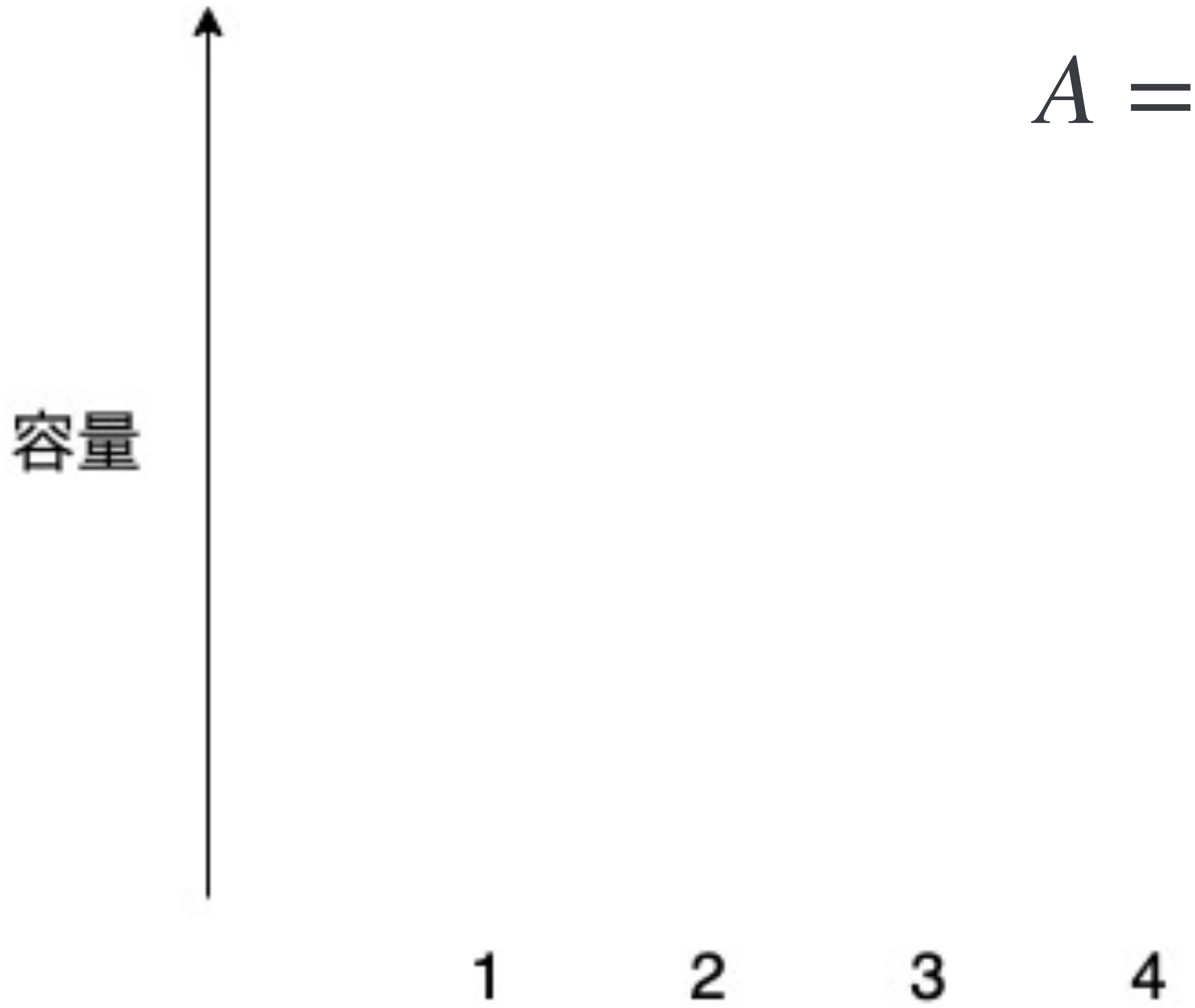
$$A = (3,2,2,2)$$



$$A = (3,2,2,2)$$

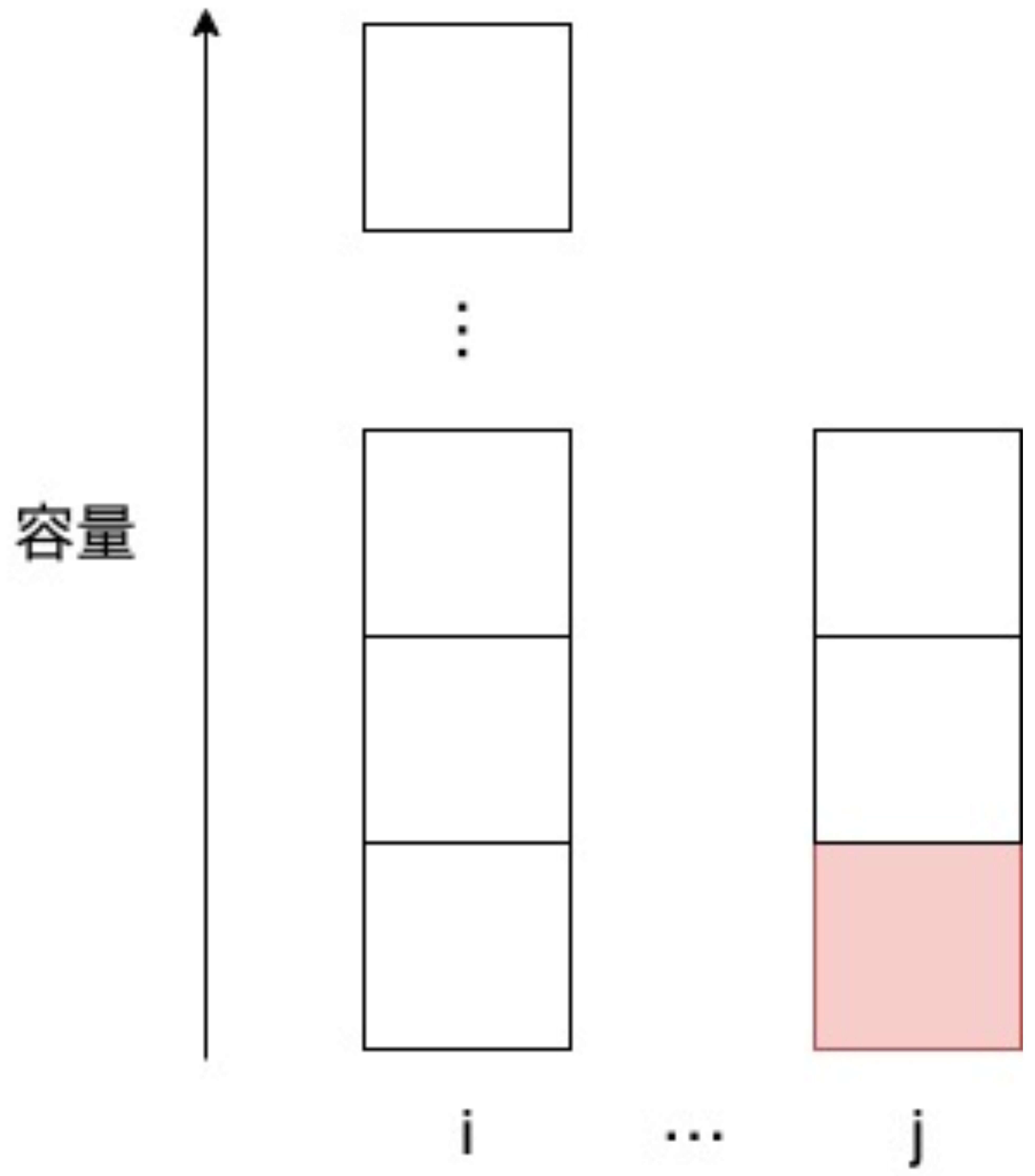


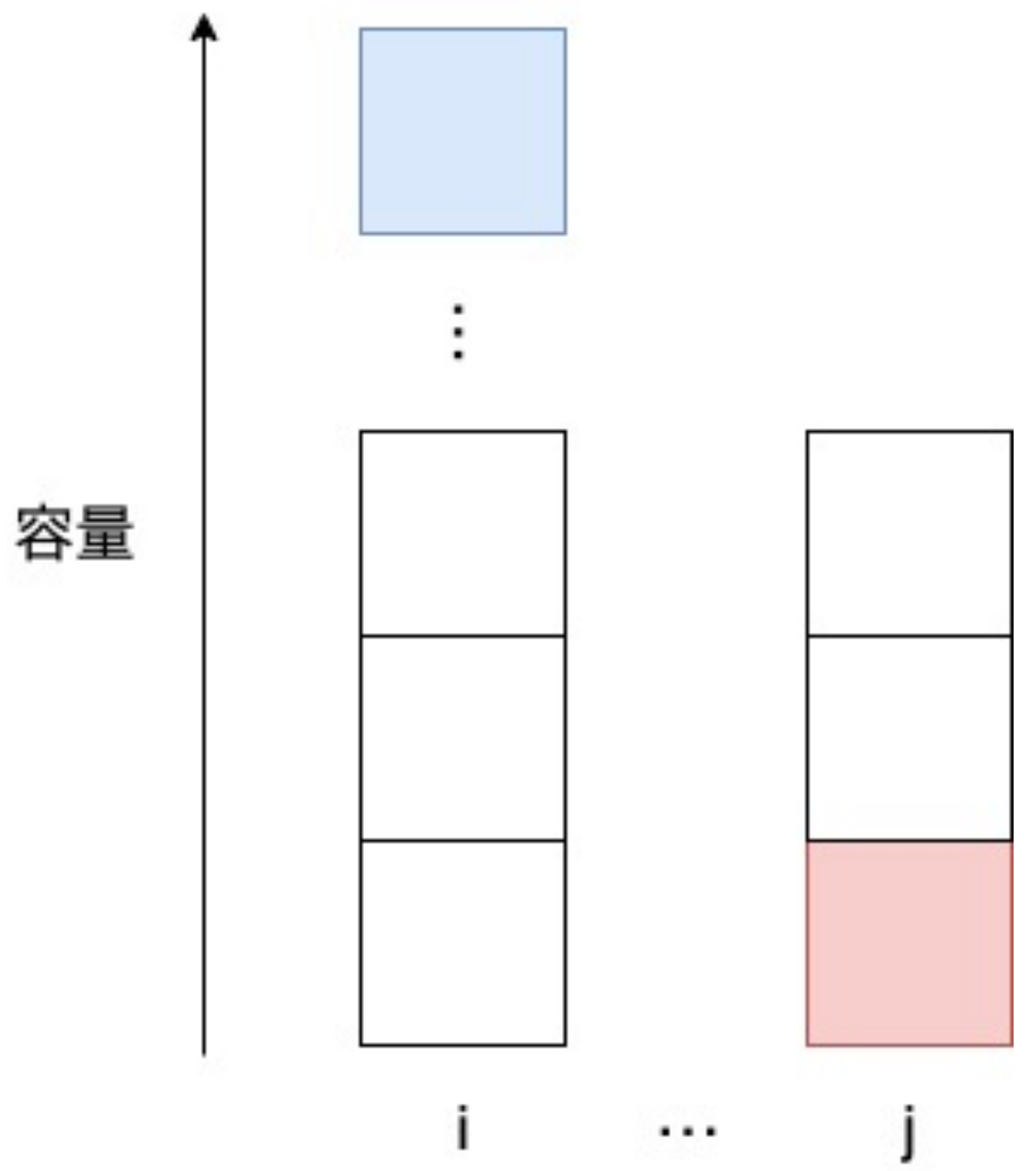
$$A = (3,2,2,2)$$

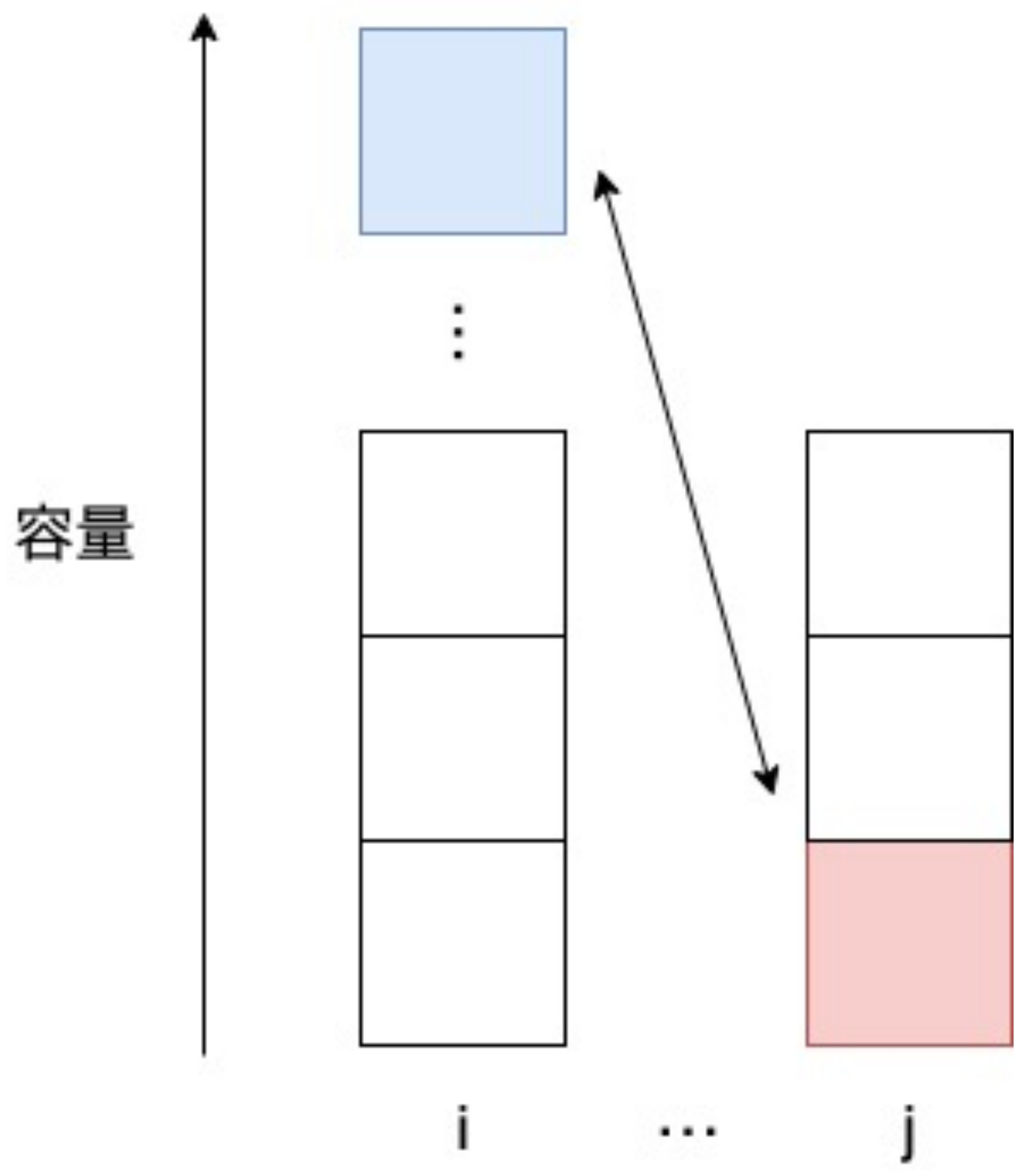


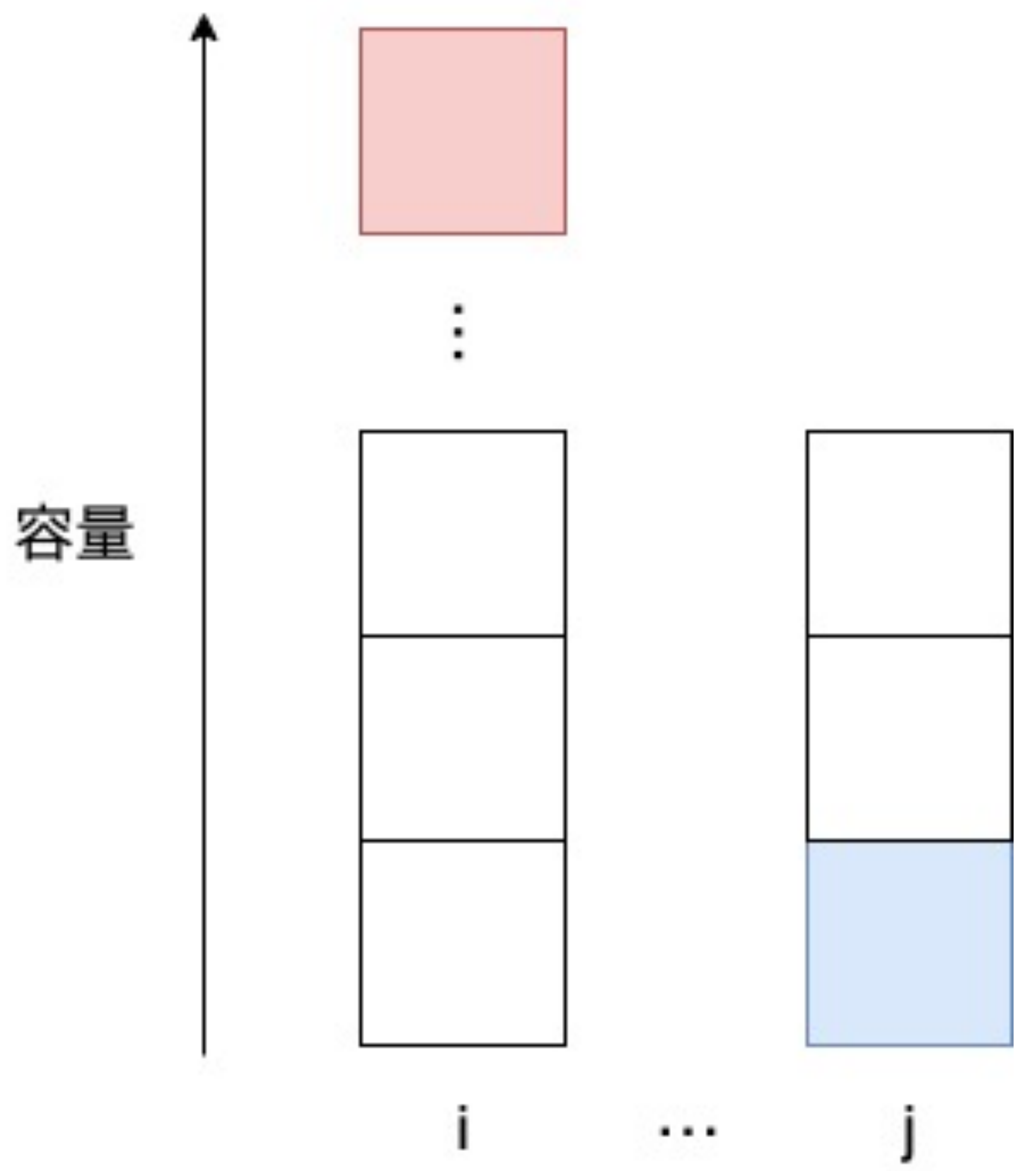
# 考察

- この貪欲法は正しい（つまり、貪欲法で割り当てを構築できないならば、条件を満たす構築は存在しない）
- 対偶を示す
  - 条件を満たす構築が存在 ⇒ 貪欲法で割り当てを構築できる


容量
⋮
i
…
j

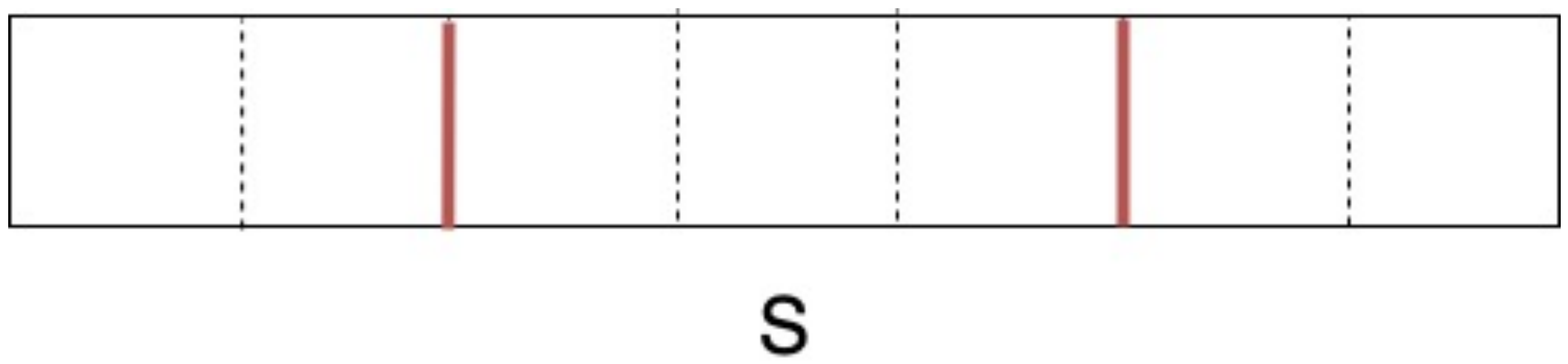

容量
⋮
i
···
j


容量
i
…
j


容量
i
…
j

# 小課題 3 ($S \leqq 15$)

- 総和が $S$ になるような箱の選び方は高々 $2^{S-1}$ 通り $\rightarrow$ 全探索できる




- それぞれの場合について、割り当て可能かどうかを貪欲法で判定

$\boxed{\text{ここまでで 25 点}}$

# 小課題 4 ($S \leqq 500$)

- 割り当て可能ならば、実際の構築は貪欲法でできる
  - 「貪欲法を実際にやってみて構築できるかどうか」で割り当て可能かどうか判定するのは間に合わない
  - もっと単純な条件を見つけたい

# 小課題 4 ($S \leqq 500$)

- 必要条件を列挙してみる　箱の容量を $C_1 \geqq C_2 \geqq \ldots$ として、
  - $C_1 \leqq N$
  - $C_1 + C_2 \leqq \sum_{i=1}^{N} \min(A_i, 2)$
  - $C_1 + C_2 + C_3 \leqq \sum_{i=1}^{N} \min(A_i, 3)$
  - $\vdots$

# 小課題 4 ($S \leqq 500$)

- 実は十分条件
- 証明は省略（帰納法など）

# 小課題 4 ($S \leqq 500$)

- 条件が単純になったので、容量が大きい順に箱を選んでいく DP ができる
  - $\mathrm{dp}[i][j][k] = i$ 箱選んで、容量が全て $j$ 以上で、容量の和が $k$ の状態にできるか
- 遷移
  - $\mathrm{dp}[i][j][k] \leftarrow \mathrm{dp}[i][j][k]\ or\ \mathrm{dp}[i][j+1][k]$
  - $\mathrm{dp}[i][j][k] \leftarrow \mathrm{dp}[i][j][k]\ or\ \mathrm{dp}[i-1][j][k-j]$ （容量 $j$ の箱を使えるとき）
- 計算量 : $O(NS^2)$

$\boxed{\text{ここまでで 70 点}}$

# 小課題 5 ($S \leqq 3\,000$)

- $\mathrm{dp}[i][j][k] = i$ 箱選んで、容量が全て $j$ 以上で、容量の和が $k$ の状態にできるか
- $ij > S$ のとき、容量の和が $S$ を超えてしまうので見ても無意味
- $(i, j)$ のペアとして考えるべきものは $S \log S$ 通り（調和級数）
- 計算量 : $O(S^2 \log S)$

$\boxed{\text{ここまでで 85 点}}$

# 小課題 6 ($S \leqq 15\,000$)

- 遷移
  - $\mathrm{dp}[i][j][k] \leftarrow \mathrm{dp}[i][j][k]\ or\ \mathrm{dp}[i][j+1][k]$
  - $\mathrm{dp}[i][j][k] \leftarrow \mathrm{dp}[i][j][k]\ or\ \mathrm{dp}[i-1][j][k-j]$ （容量 $j$ の箱を使えるとき）
- $\mathrm{dp}[i][j]$ の値をまとめて bitset で持つと、
  - $\mathrm{dp}[i][j] \leftarrow \mathrm{dp}[i][j]\ or\ \mathrm{dp}[i][j+1]$
  - $\mathrm{dp}[i][j] \leftarrow \mathrm{dp}[i][j]\ or\ \mathrm{dp}[i-1][j]$ << $j$
- 計算量 : $O\left(\dfrac{S^2 \log S}{w}\right)$

ここまでで 100 点

# 得点分布