Festival 2 解説

解説 : yuto1115

区間スケジューリング問題（前提知識）

- N 個の区間があり、i 番目の区間は $[A_i, B_i]$ で表される $(A_i < A_{i+1},\ A_i < B_i)$
- 区間同士が共通部分を持たないようにいくつかの区間を選ぶとき、最大で何個選べるか？
- この問題は貪欲法で解ける
  - B_i の昇順に区間を見ていって、「いままで選んだどの区間とも共通部分を持たないならばその区間を選ぶ」ことを繰り返す

区間スケジューリング問題（前提知識）

- 一方で、誤った貪欲法の例として以下のアルゴリズムがよく紹介される
  - A_i の昇順に区間を見ていって、「いままで選んだどの区間とも共通部分を持たないならばその区間を選ぶ」ことを繰り返す

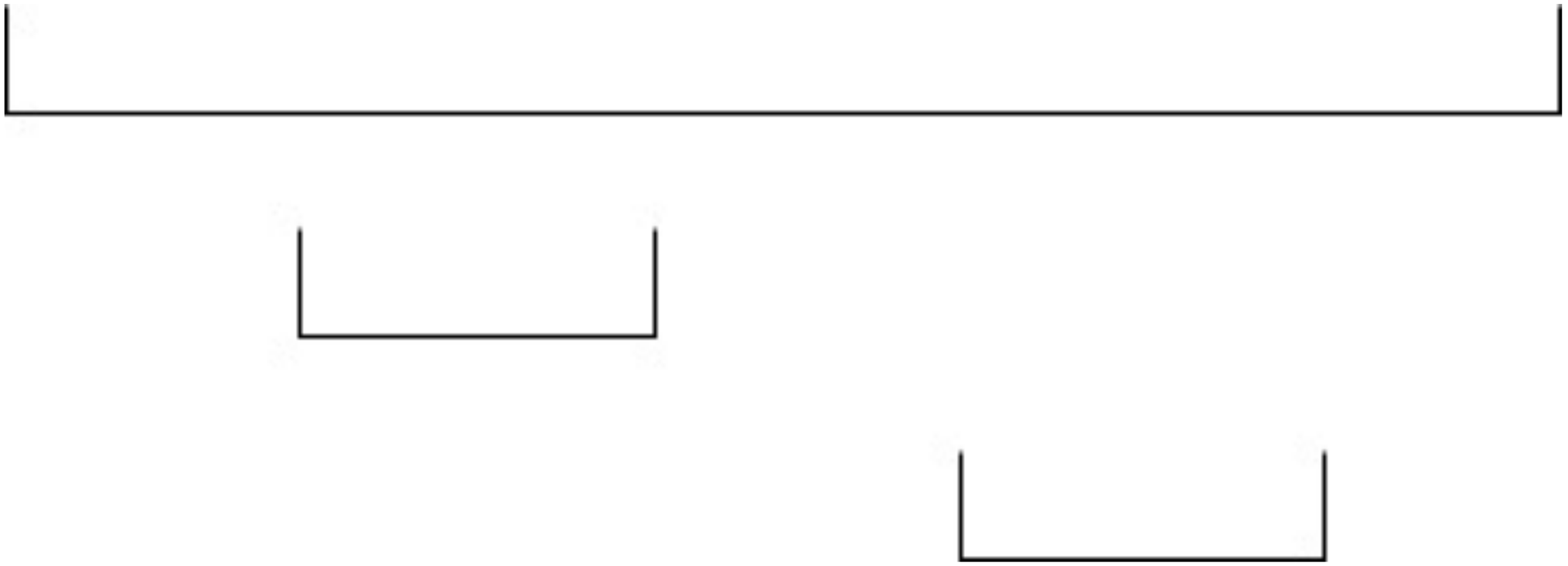
- 以降、↑ の貪欲法のことを嘘貪欲と呼ぶ

問題概要

- N が与えられる。N 区間からなる区間スケジューリング問題の入力としてありうるもののうち、正しい貪欲で得られる解が嘘貪欲で得られる解より改善するものの数を数えよ。
- 区間の端点は 1 以上 $2N$ 以下の整数であり、$2N$ 個の整数がすべてちょうど一度ずつ現れるものとする
- 大きい素数 P が与えられるので、答えを P で割った余りを出力する
  （埋め込み防止）

入力例 ($N = 3$)

$A = (1,2,4), B = (6,3,5)$

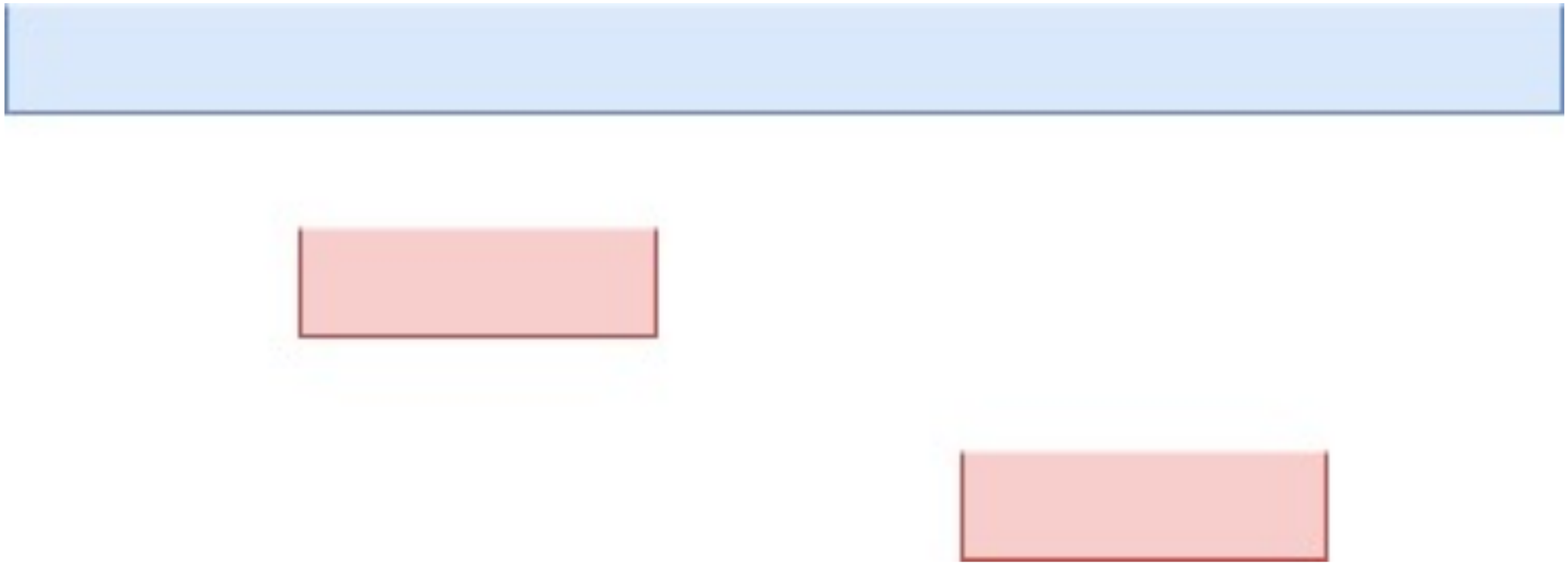


入力例 ($N = 3$)

正しい貪欲

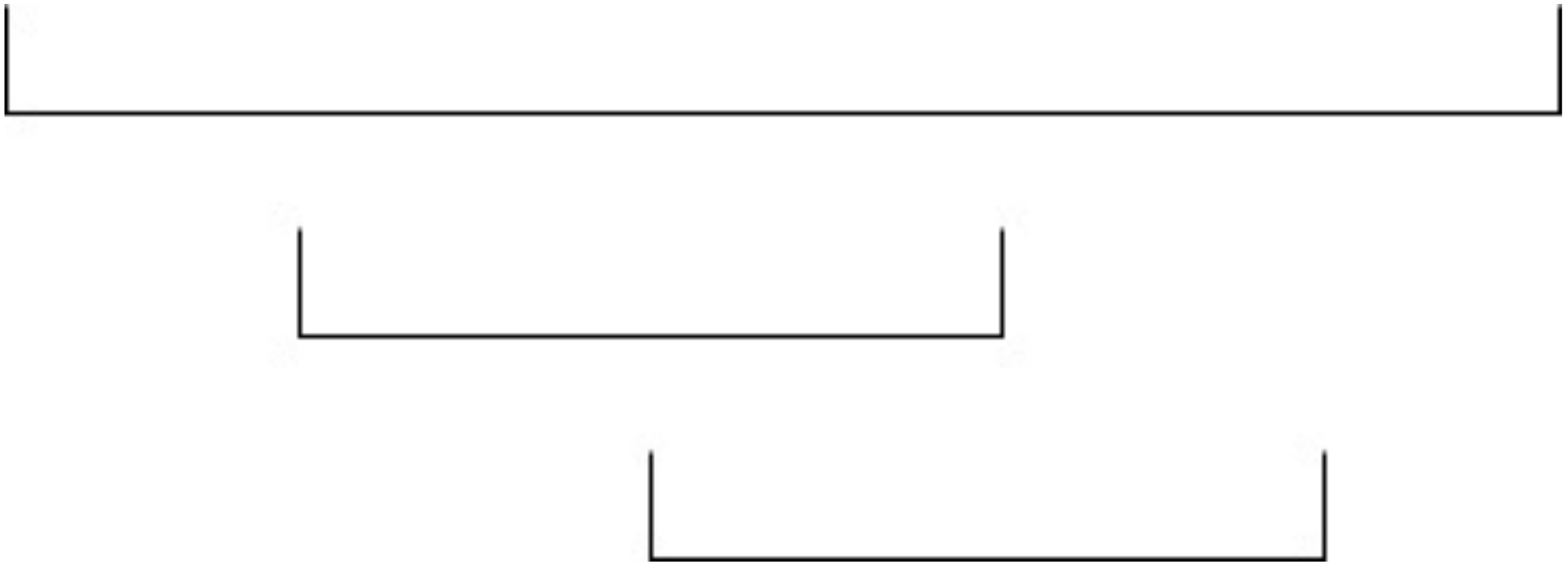
嘘貪欲

$A = (1,2,4), B = (6,3,5)$



入力例 ($N = 3$)

$A = (1,2,3), B = (6,4,5)$

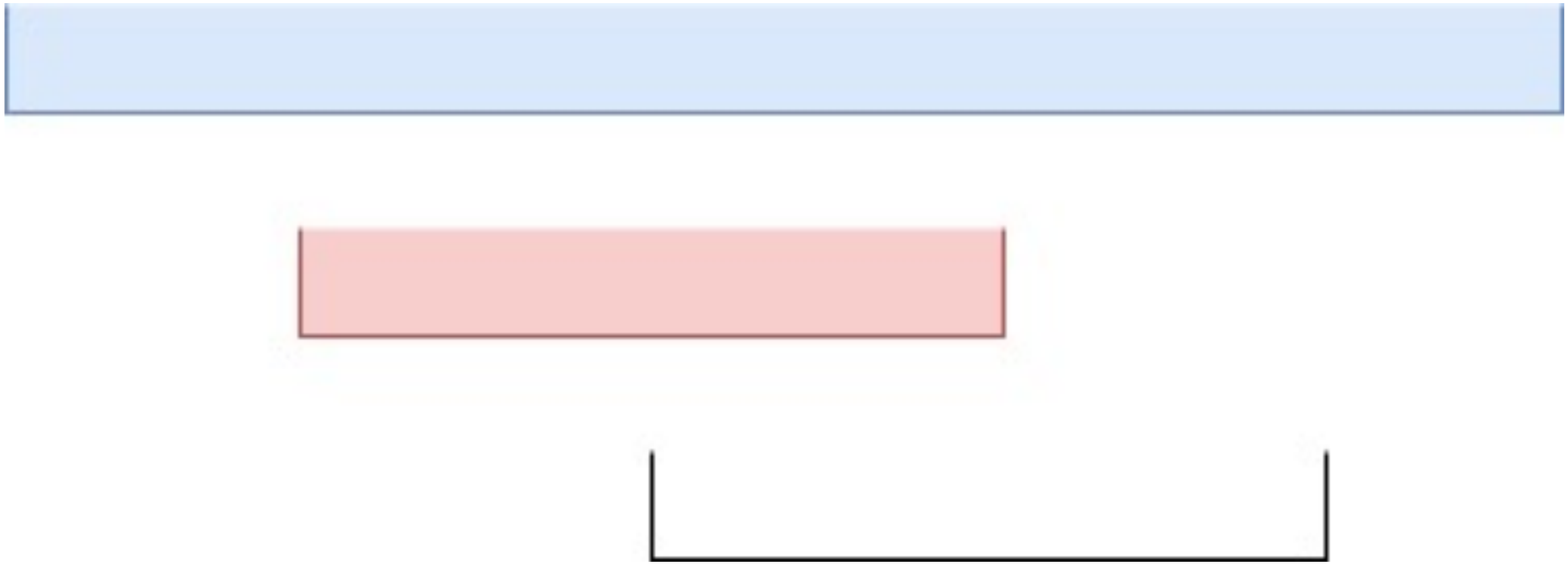


入力例 ($N = 3$)

正しい貪欲

嘘貪欲

$A = (1,2,3), B = (6,4,5)$



小課題 1 ($N \leqq 5$)

- A_i, B_i に 1 から $2N$ までの数を割り当てる方法として $(2N)!$ 通りを全探索する
- それぞれの割り当てについて、
  - valid な入力 ($A_i < A_{i+1}$, $A_i < B_i$) かどうかチェックする
  - 正しい貪欲 > 嘘貪欲かどうかチェックする

ここまでで 5 点

小課題 2 ($N \leqq 8$)

- $(2N)!$ 通りを全探索するのは無駄なので、最初から valid な入力だけを探索したい
- DFS を用いて、i の昇順に A_i, B_i を定めていく
  - A_i を選ぶとき：まだ使っていない数のうち最小のものを選ぶしかない
  - B_i を選ぶとき：まだ使っていない数のうちどれを選んでもいい
- $N = 8$ のとき、valid な入力は $15 \times 13 \times 11 \times 9 \times 7 \times 5 \times 3 \times 1 = 2027025$ 通り
- それぞれについて、正しい貪欲 > 嘘貪欲かどうかチェックすればよい

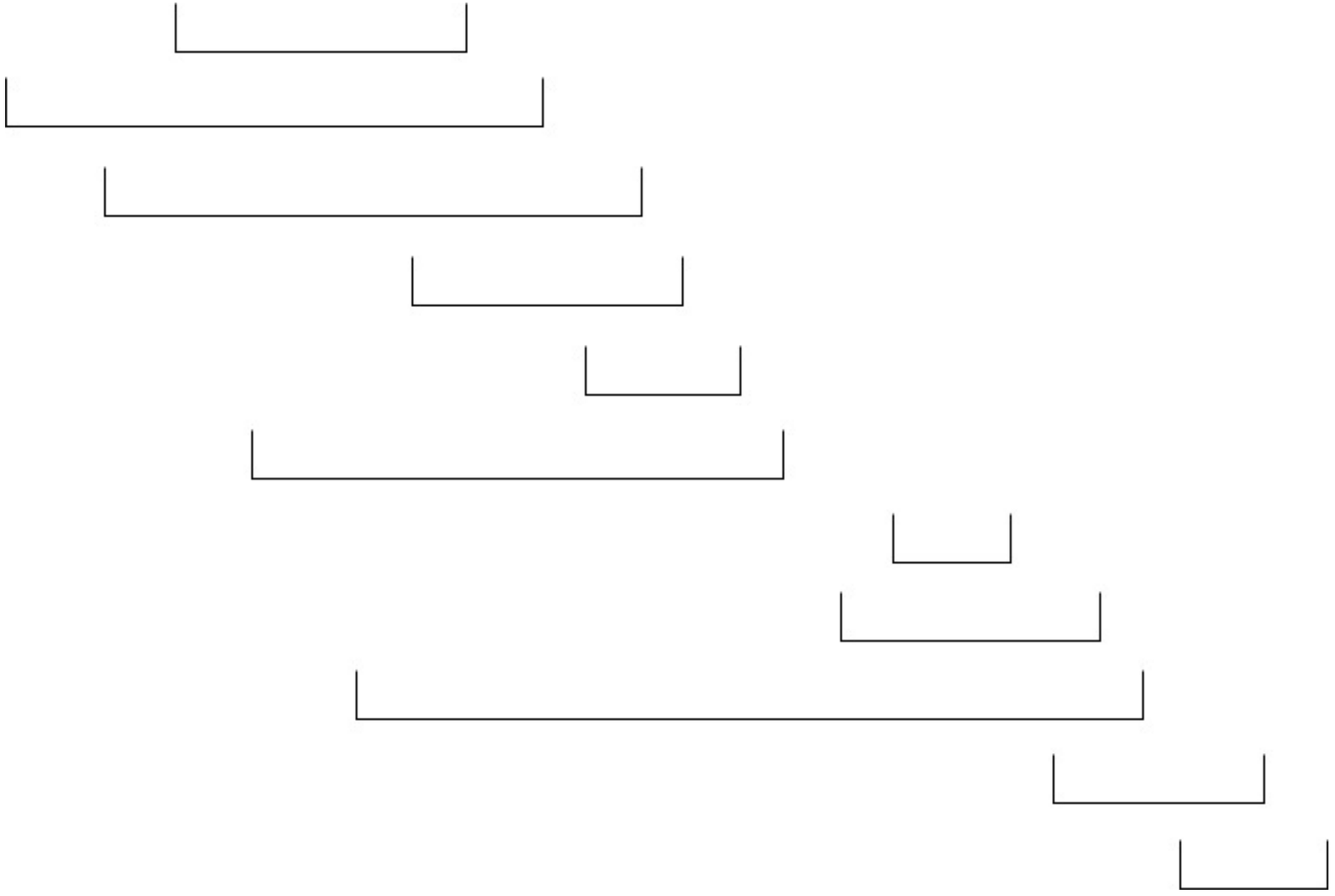
ここまでで 10 点

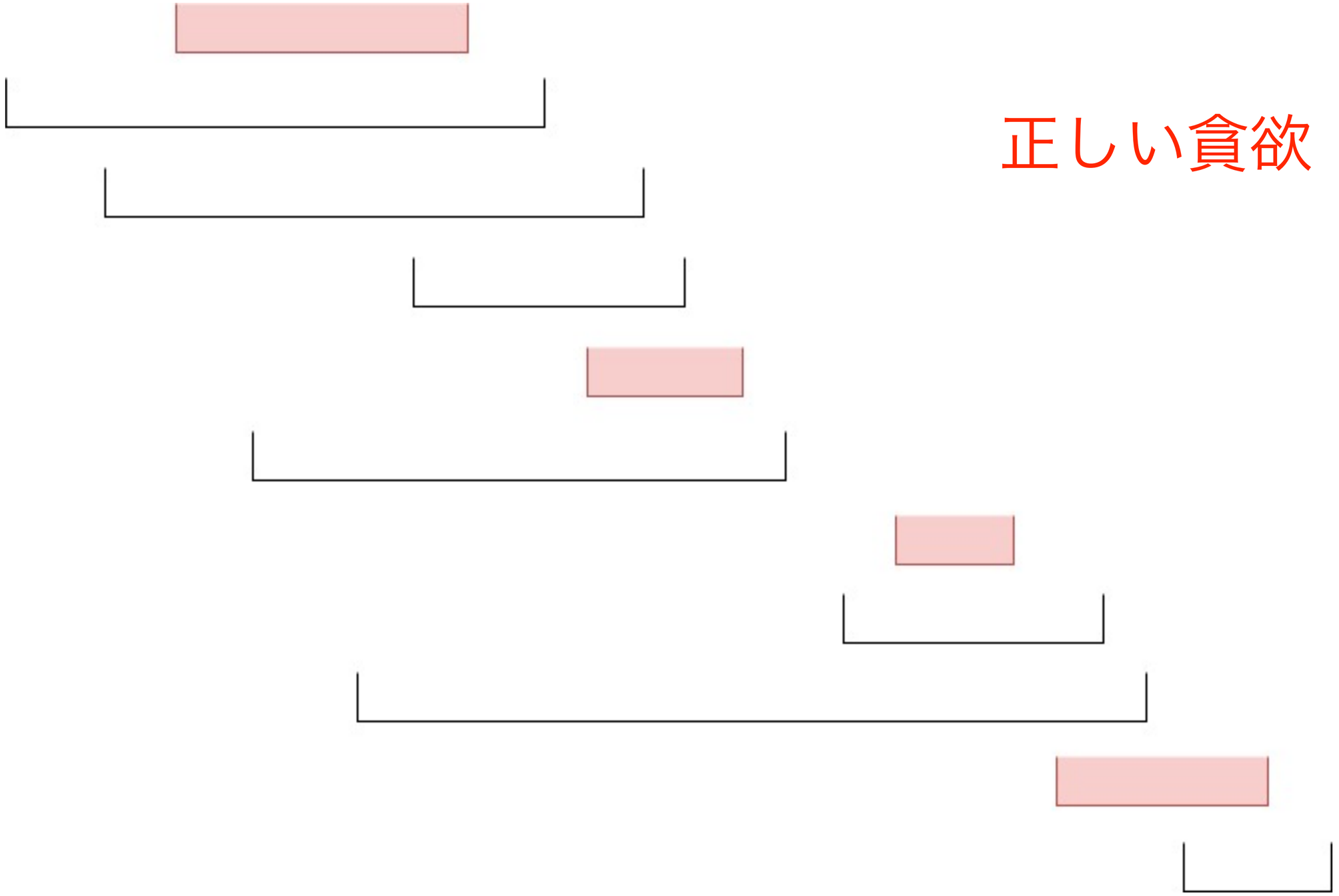
小課題 3 ($N \leqq 30$)

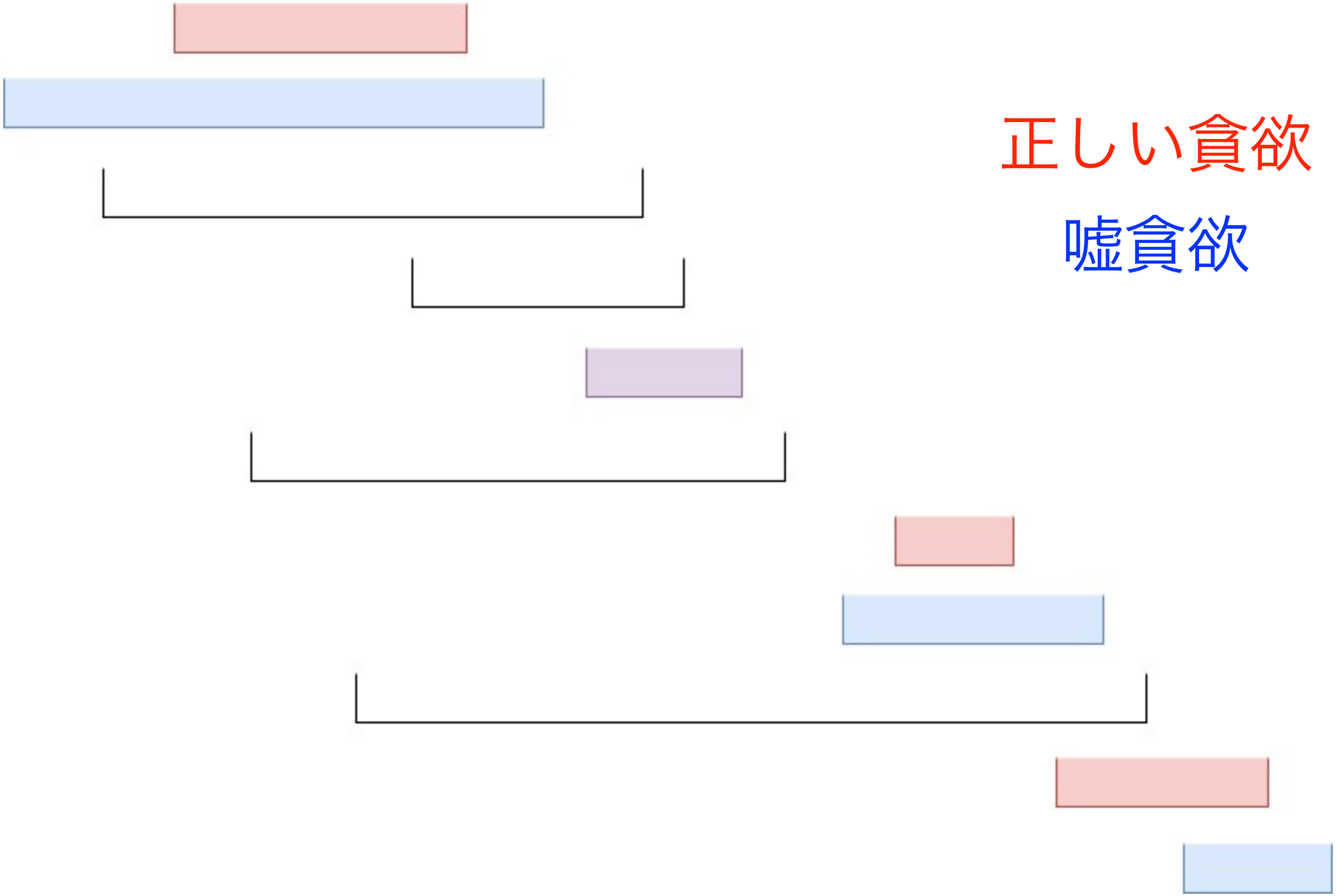
- ここからが本番
- うまく動的計画法に落とし込んで、多項式時間で解こう
- いろいろな解法が考えられるが、一例を紹介
- 簡単のため、嘘貪欲を正しい貪欲にしても改善**しない**ものの数を数えることにする

小課題 3 ($N \leqq 30$)

- 区間を順に定めていきながら、「正しい貪欲で選ばれる区間」と「嘘貪欲で選ばれる区間」をそれぞれ管理していく
- それぞれの貪欲で選ばれる区間を本当にすべて保持したら破綻する
- 図を書いて、正しい貪欲 = 嘘貪欲となるときの構造を観察しよう



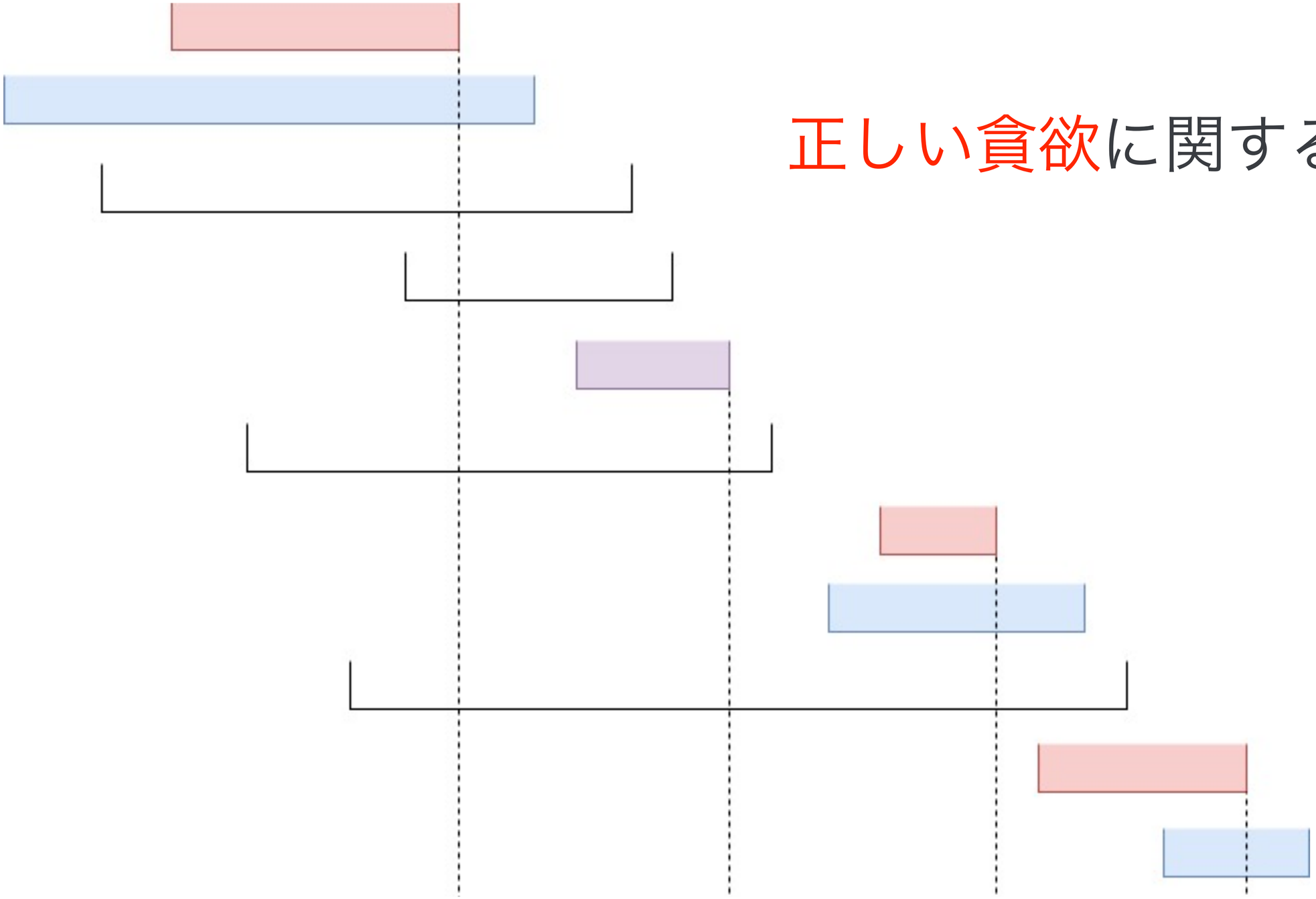

正しい貪欲


正しい貪欲
嘘貪欲

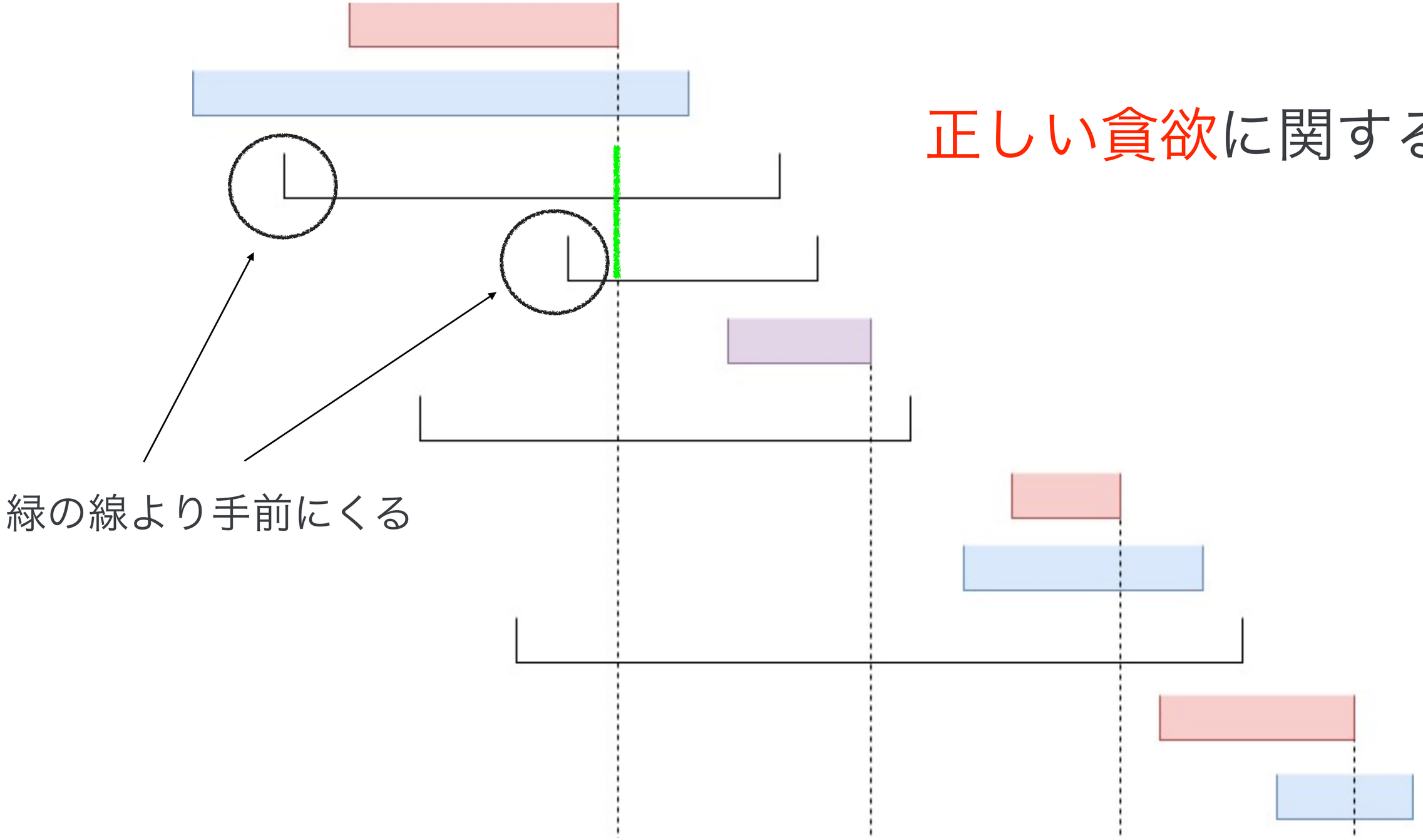
考察

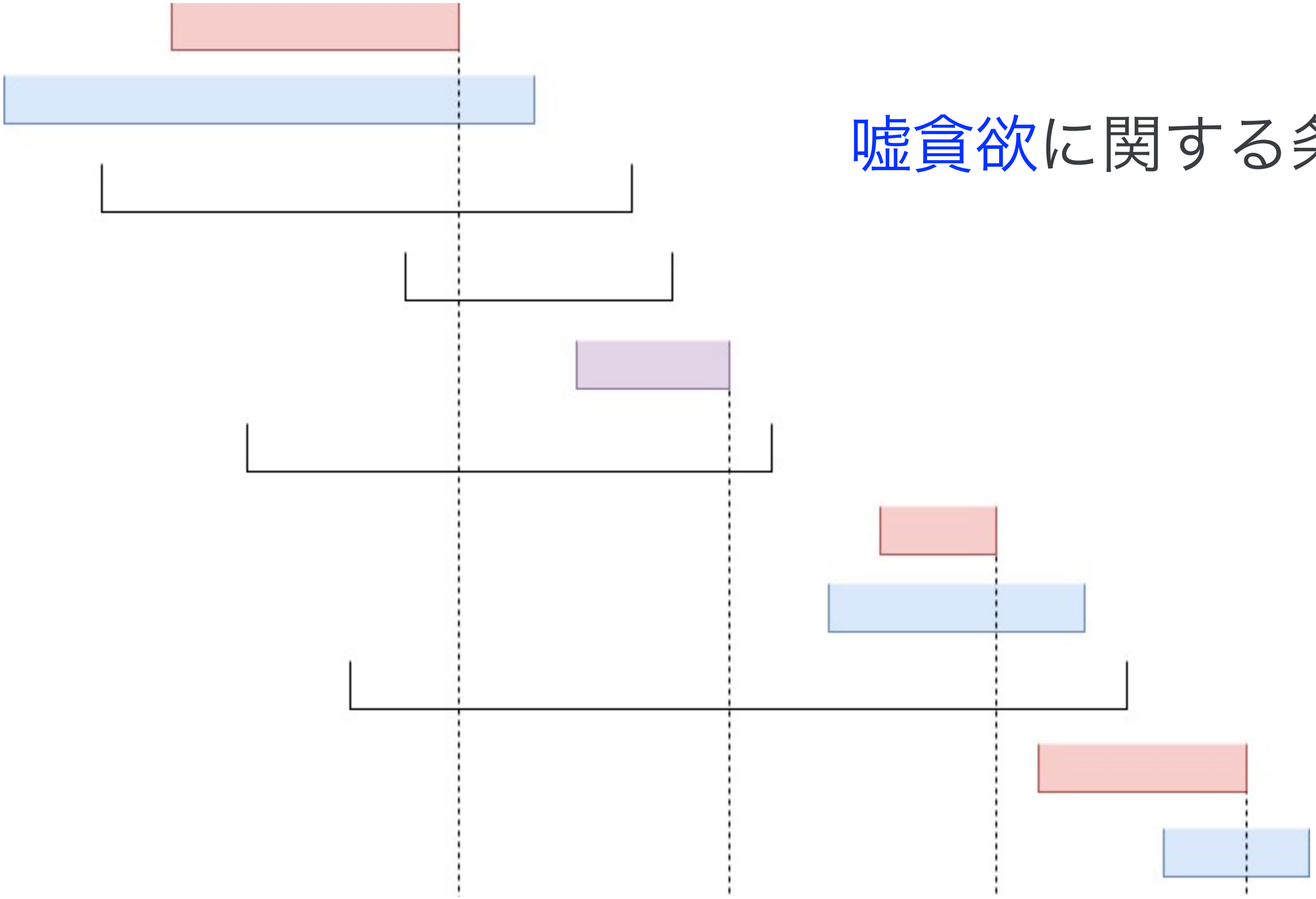
- B_i の昇順に区間を並べると、「正しい貪欲で選ばれる区間」と「嘘貪欲で選ばれる区間」が交互にやってくる（被るときもある）
- B_i の昇順に区間を定めながら、「正しい貪欲で選ばれる区間」と「嘘貪欲で選ばれる区間」を交互に指定する感じで DP をすればよさそう
- それぞれの貪欲で選ばれる区間が、指定した区間に本当に一致するための条件は？

正しい貪欲に関する条件

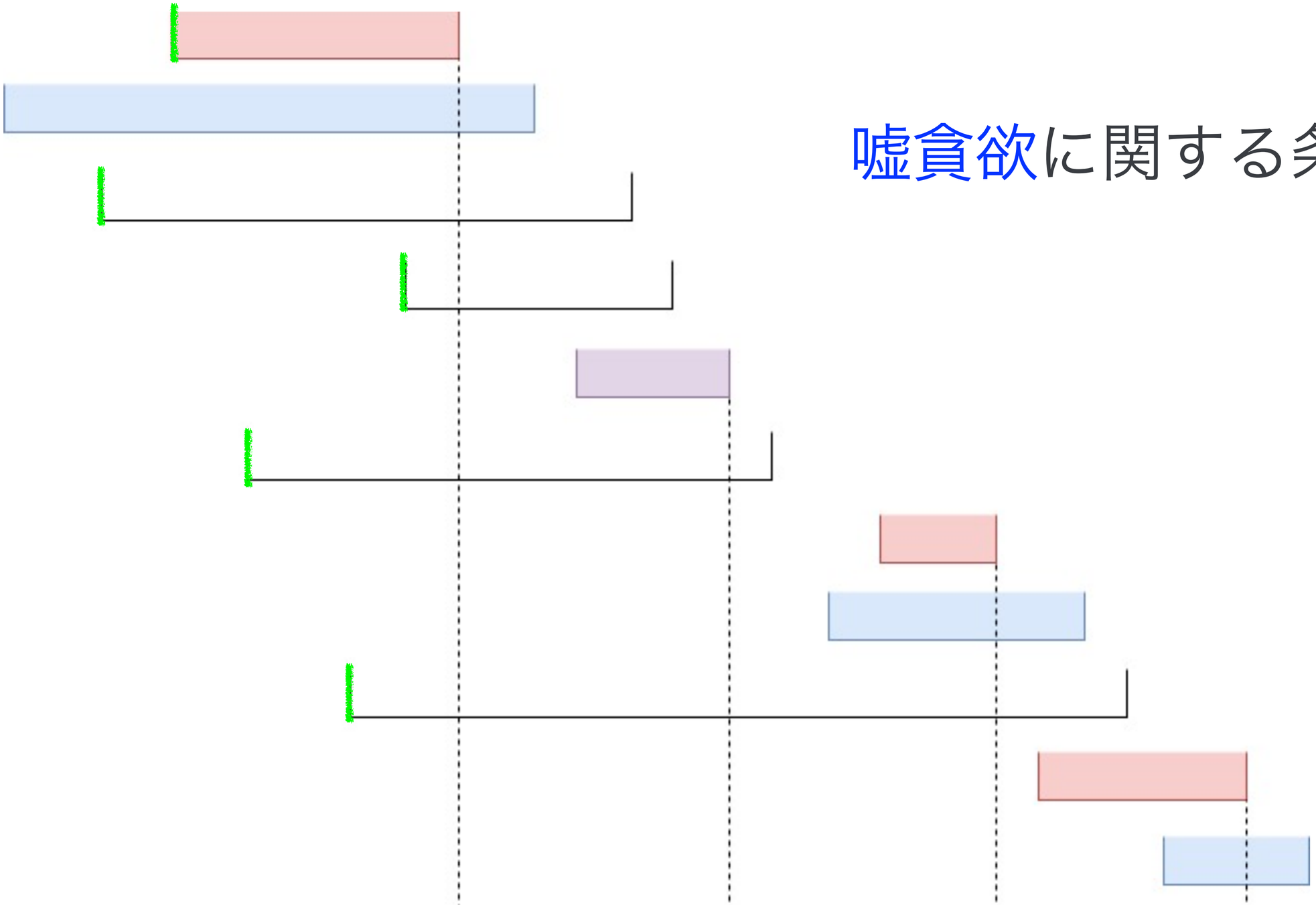


正しい貪欲に関する条件



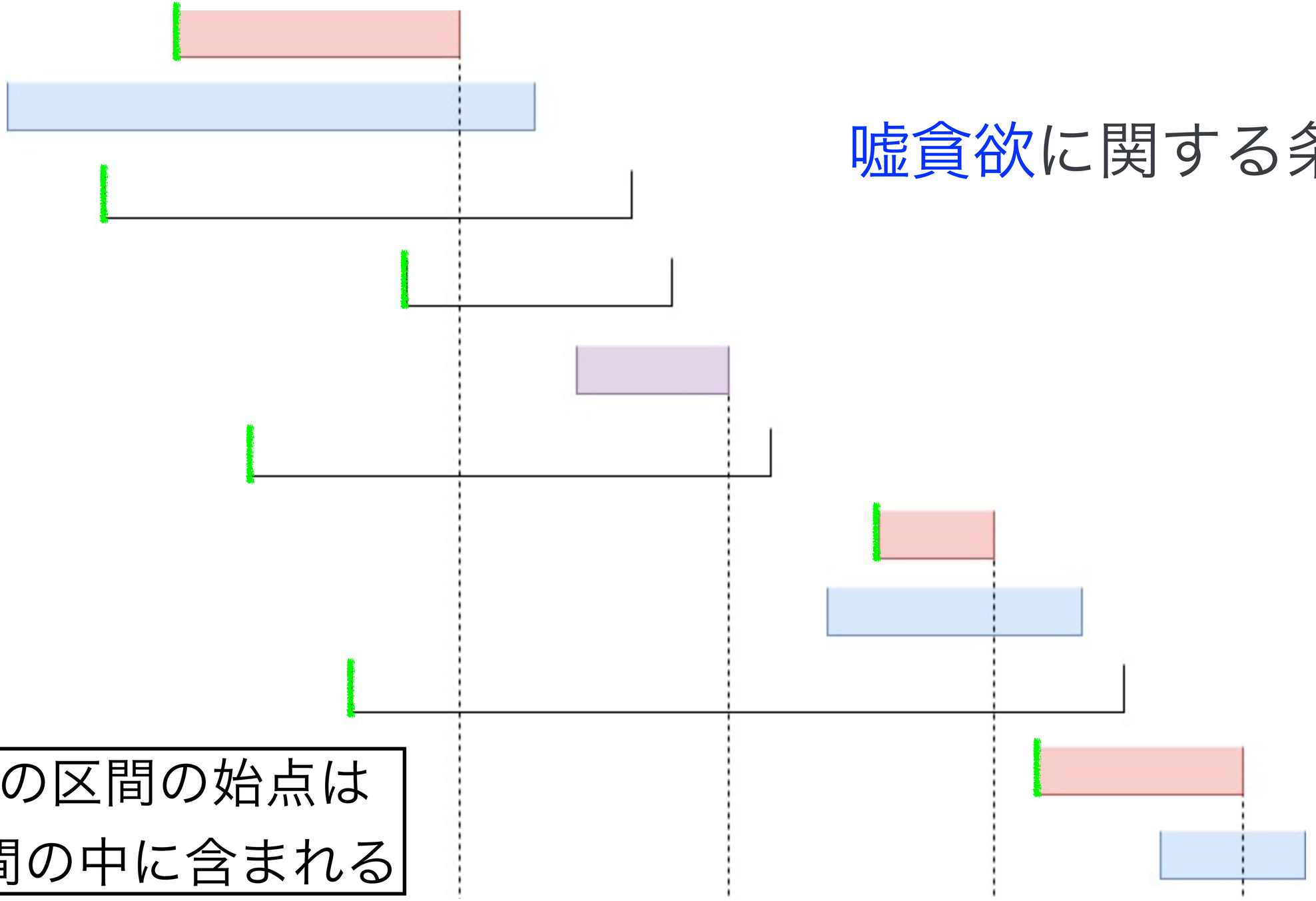


嘘貪欲に関する条件



嘘貪欲に関する条件

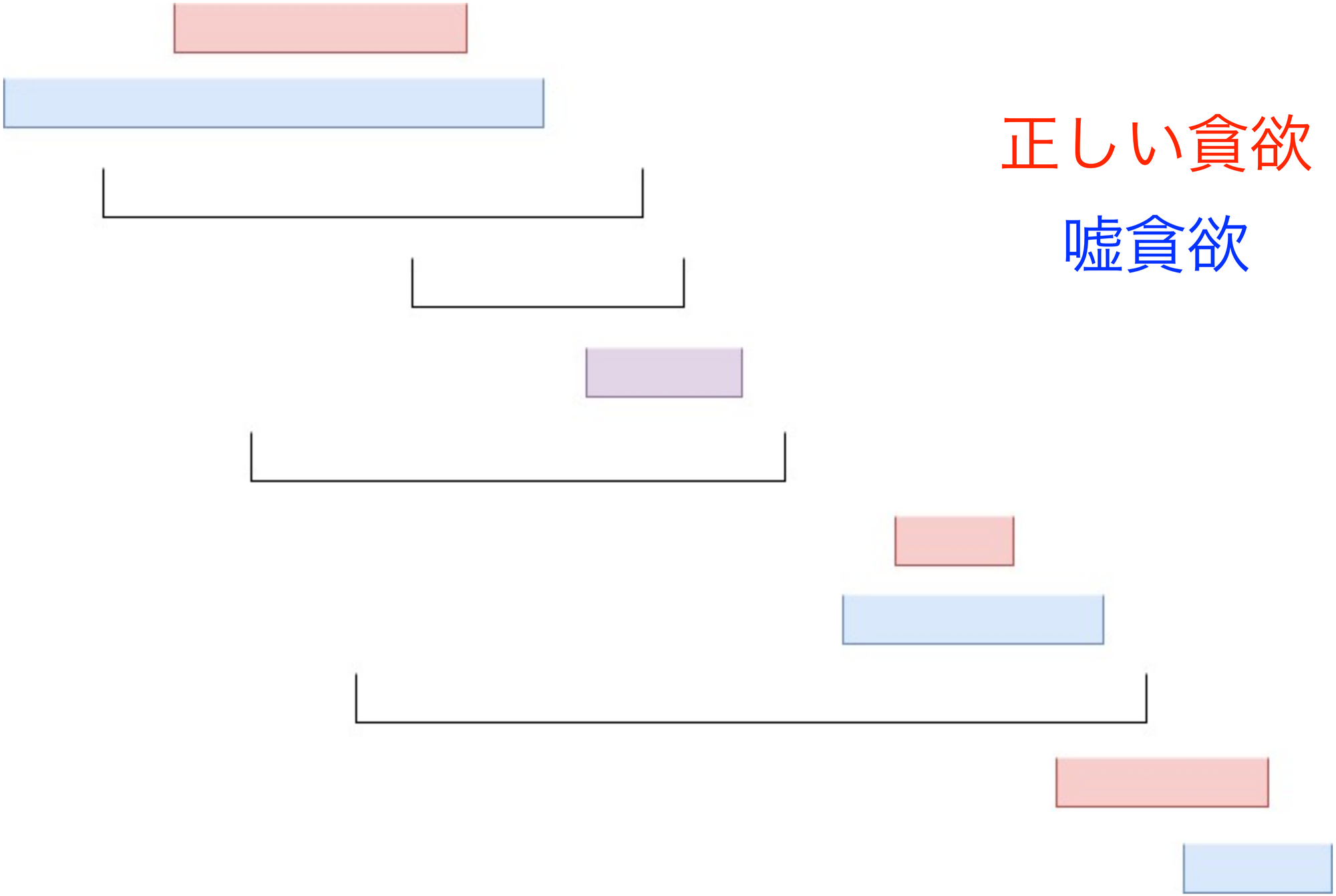
嘘貪欲に関する条件



青以外の区間の始点は
青い区間の中に含まれる

小課題 3 ($N \leqq 30$)

- B_i の昇順に区間を定める挿入 DP
- 以下の情報を持つ
  - 定めた区間の個数
  - 選んだ赤い区間の数 − 青い区間の数 (0 or 1)
  - 最後に選んだ赤い区間は何番目か
  - 最後に選んだ青い区間は何番目か
  - 挿入できる箇所のうち、青い区間の中に含まれるのは何箇所か


正しい貪欲
嘘貪欲

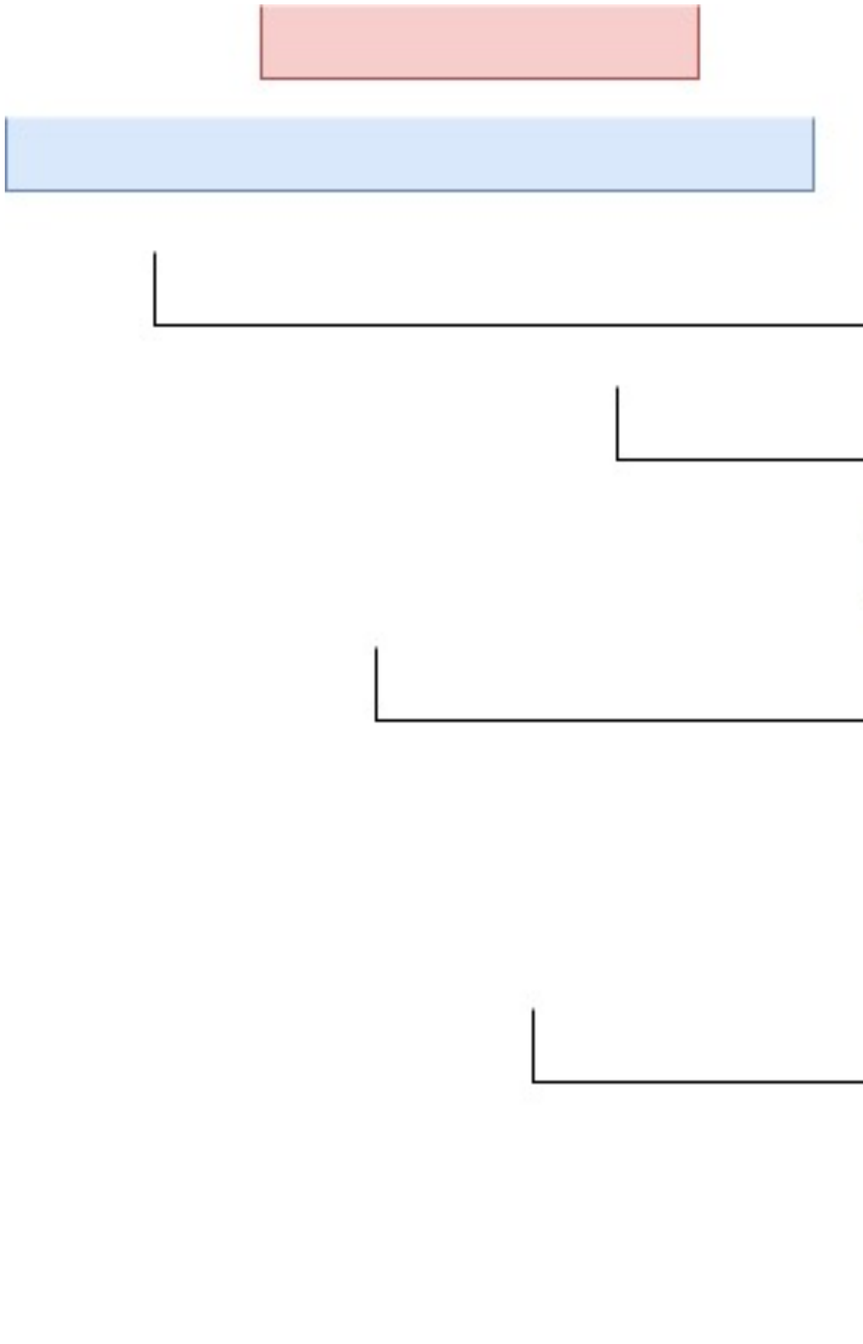
小課題 3 ($N \leqq 30$)

- 丁寧に遷移を書くと、$O(N^5)$, $O(N^6)$ などになる
- この方針を高速化すると $O(N^3)$ まで減らせるが、実装が煩雑なので省略

ここまでで 37 点

小課題 4 ($N \leqq 300$)

- 挿入 DP ではなく、1 から $2N$ までの数を順番に区間の端点に割り当てていく
- 「始点は決まったが終点がまだ決まっていない区間」がいくつか存在することになる（これを単に「開いた区間」と呼びます）
- まず図でざっくり説明します



正しい貪欲

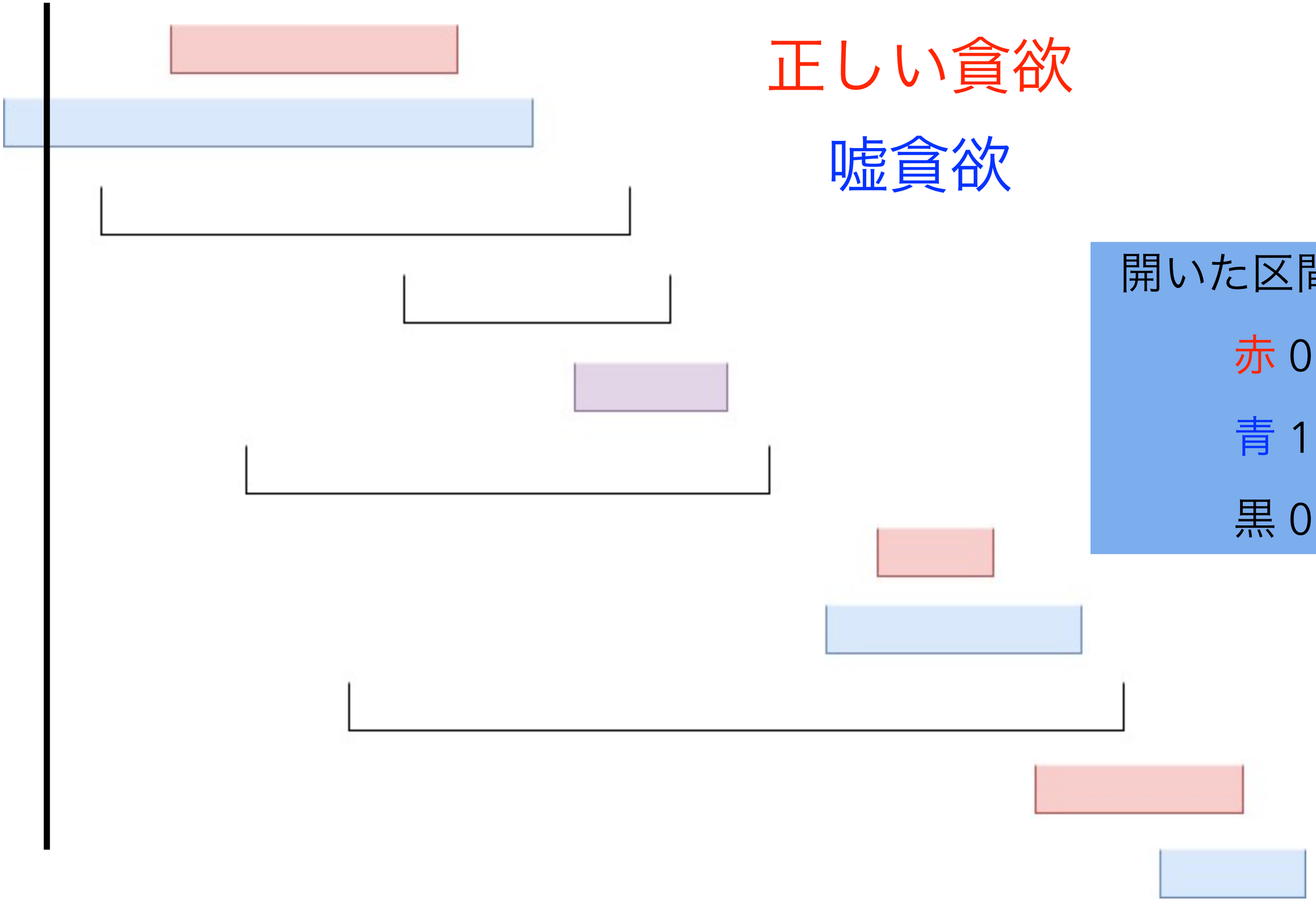
嘘貪欲

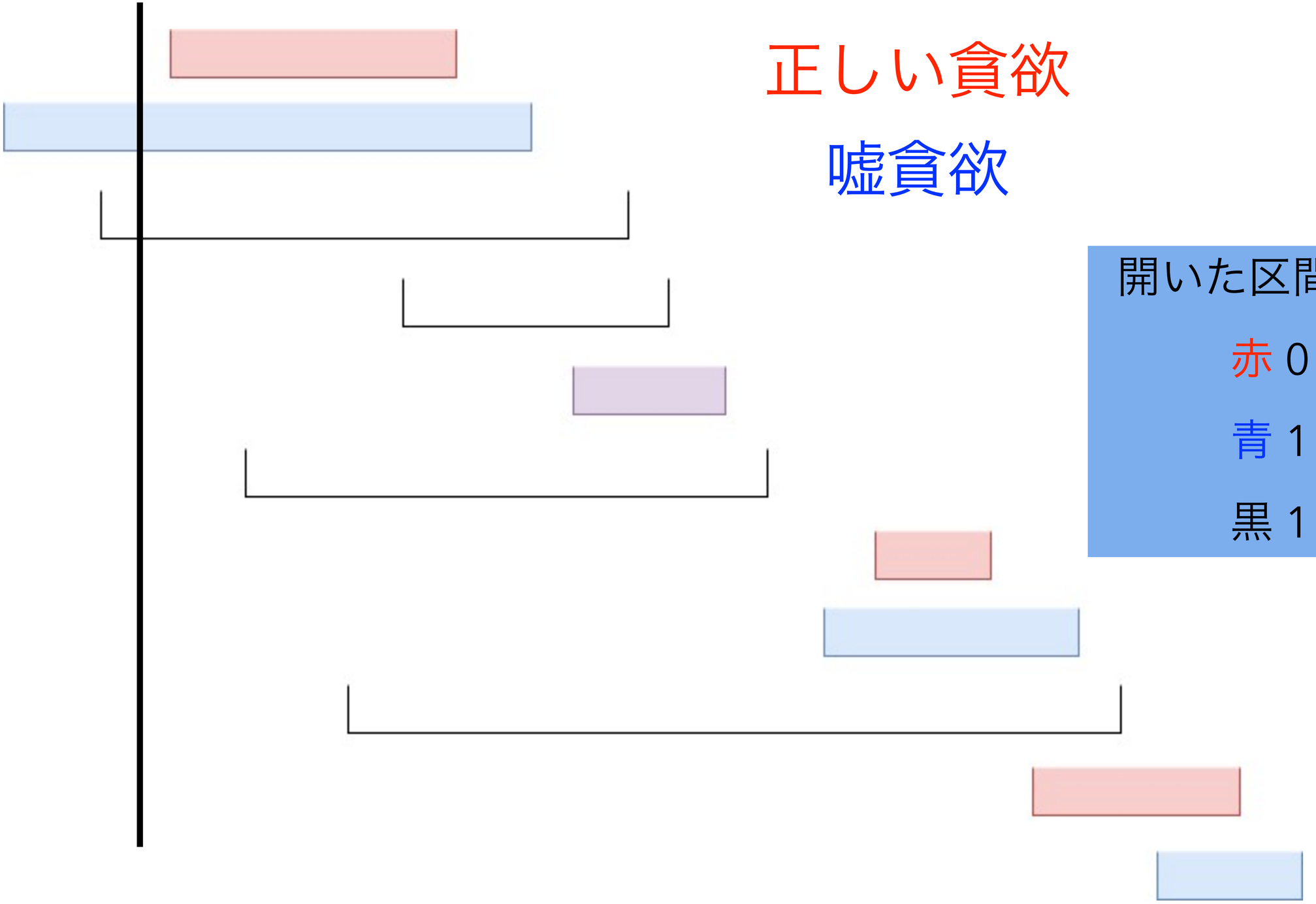
開いた区間の数

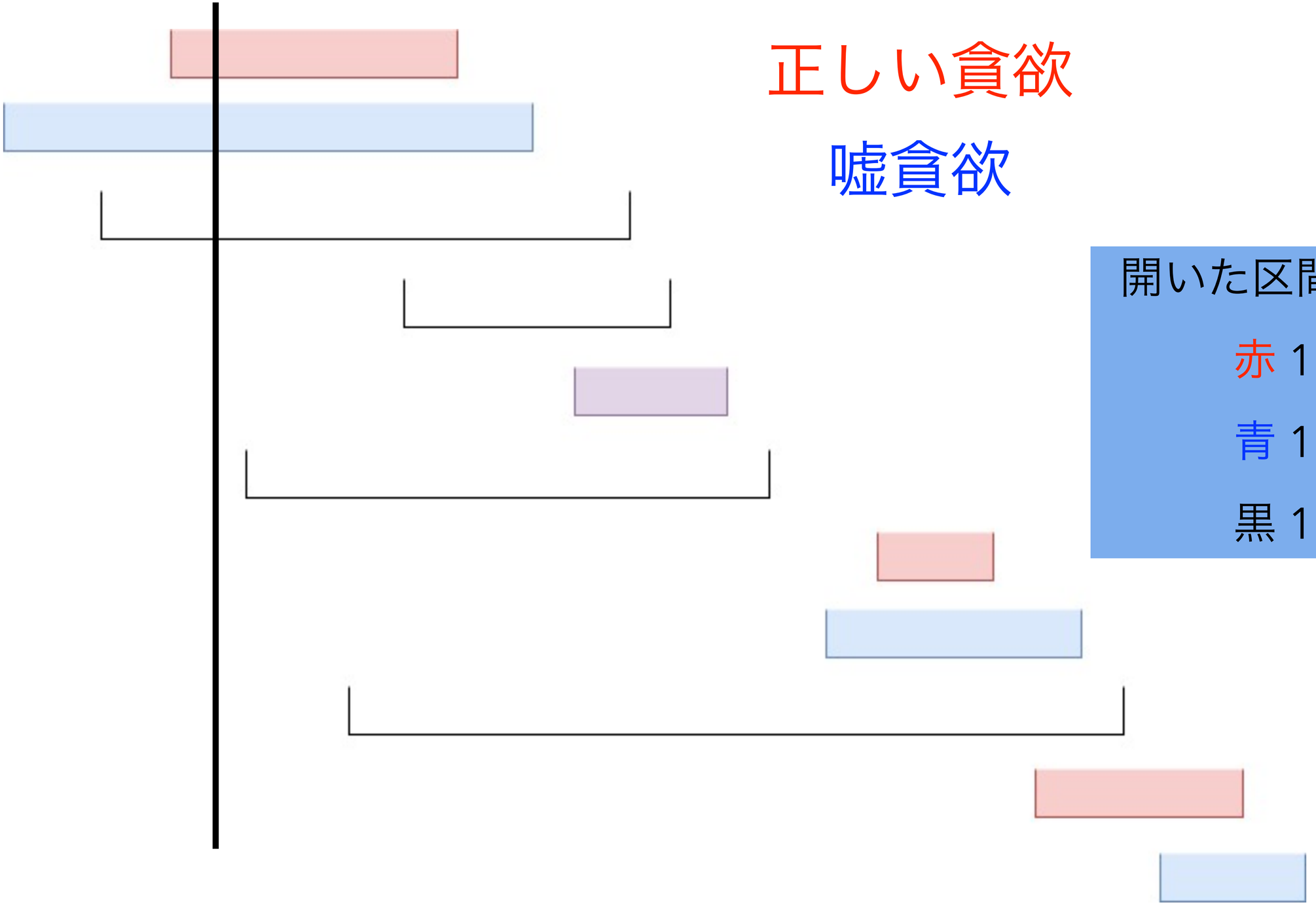
赤 0

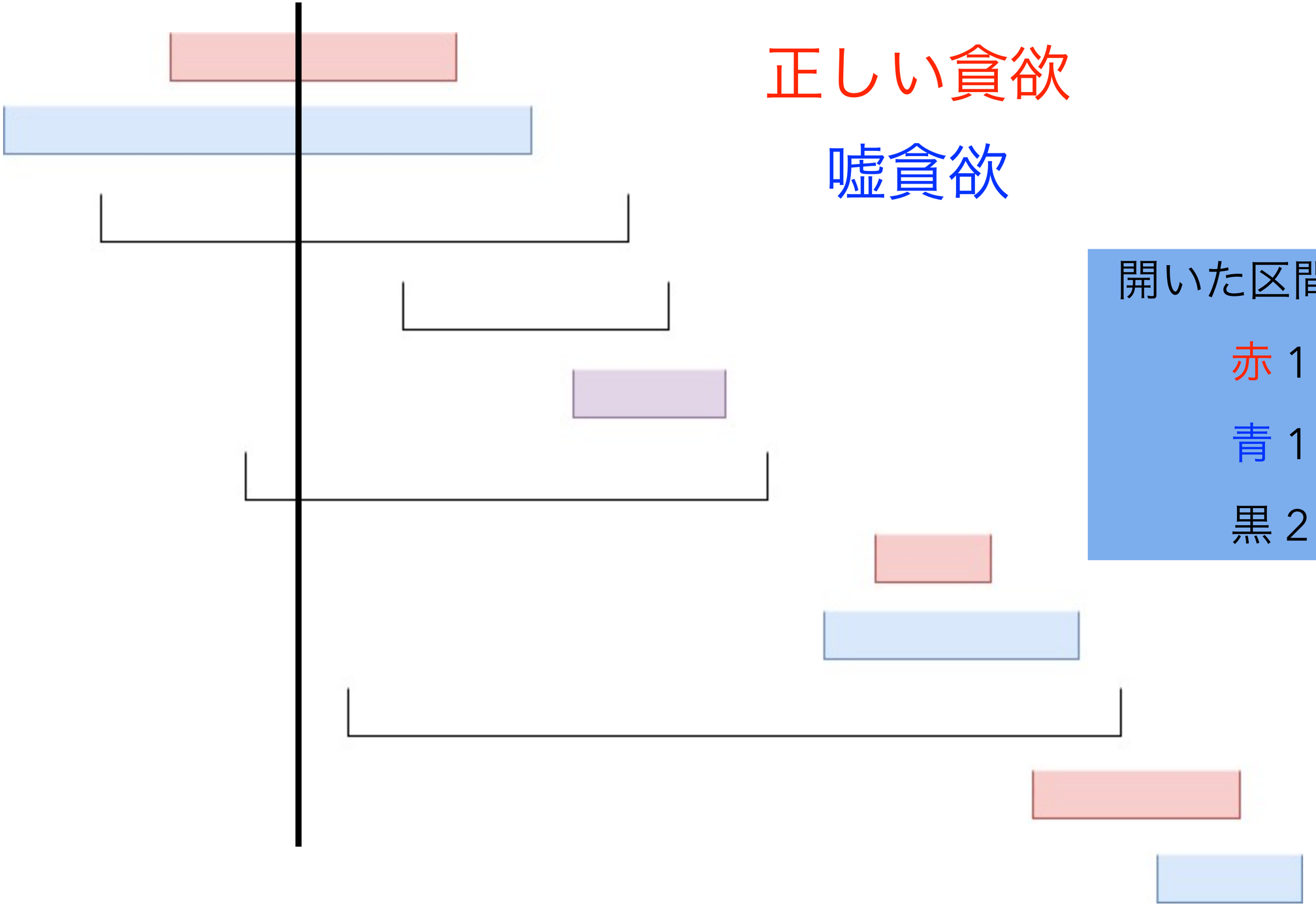
青 0

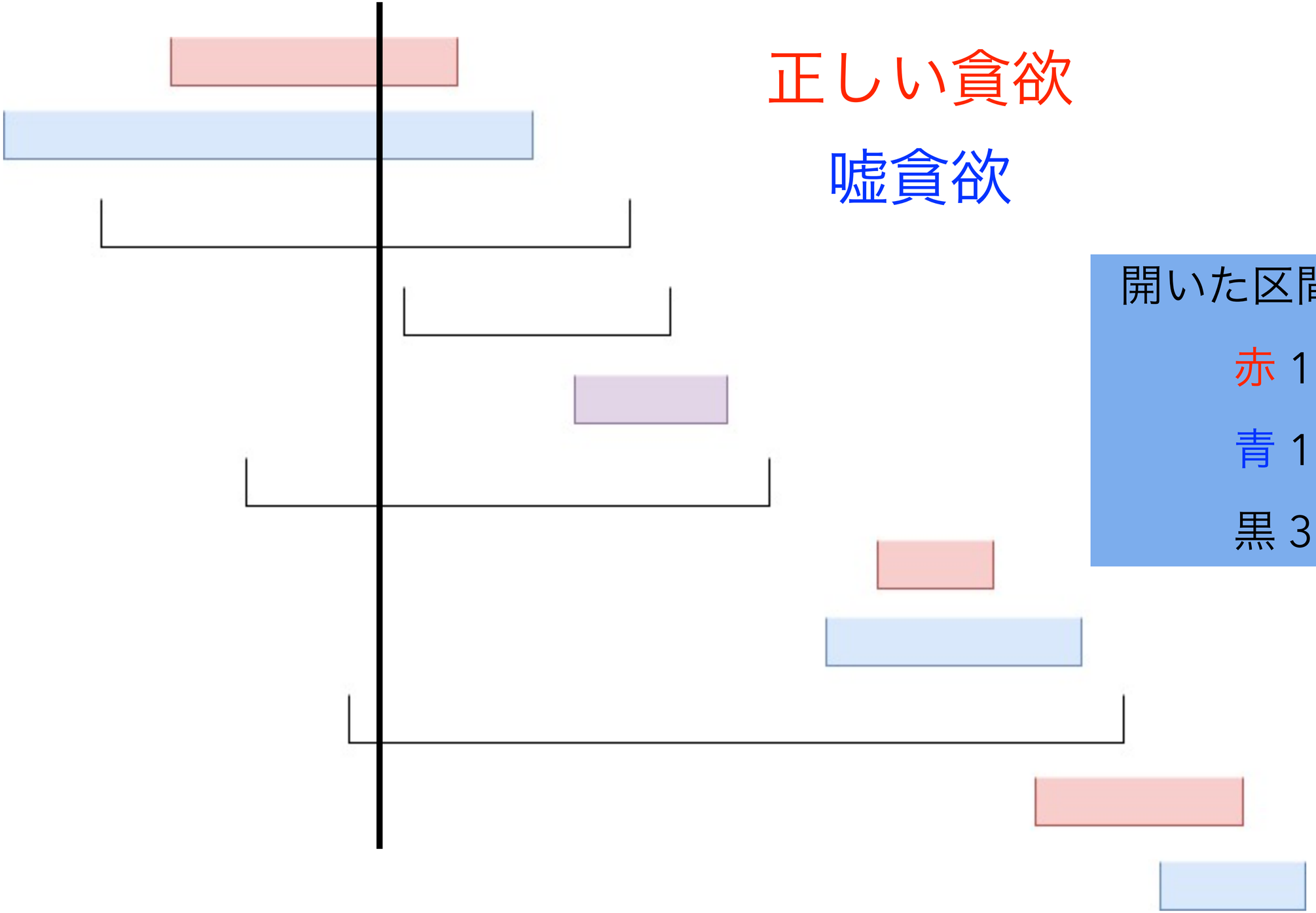
黒 0

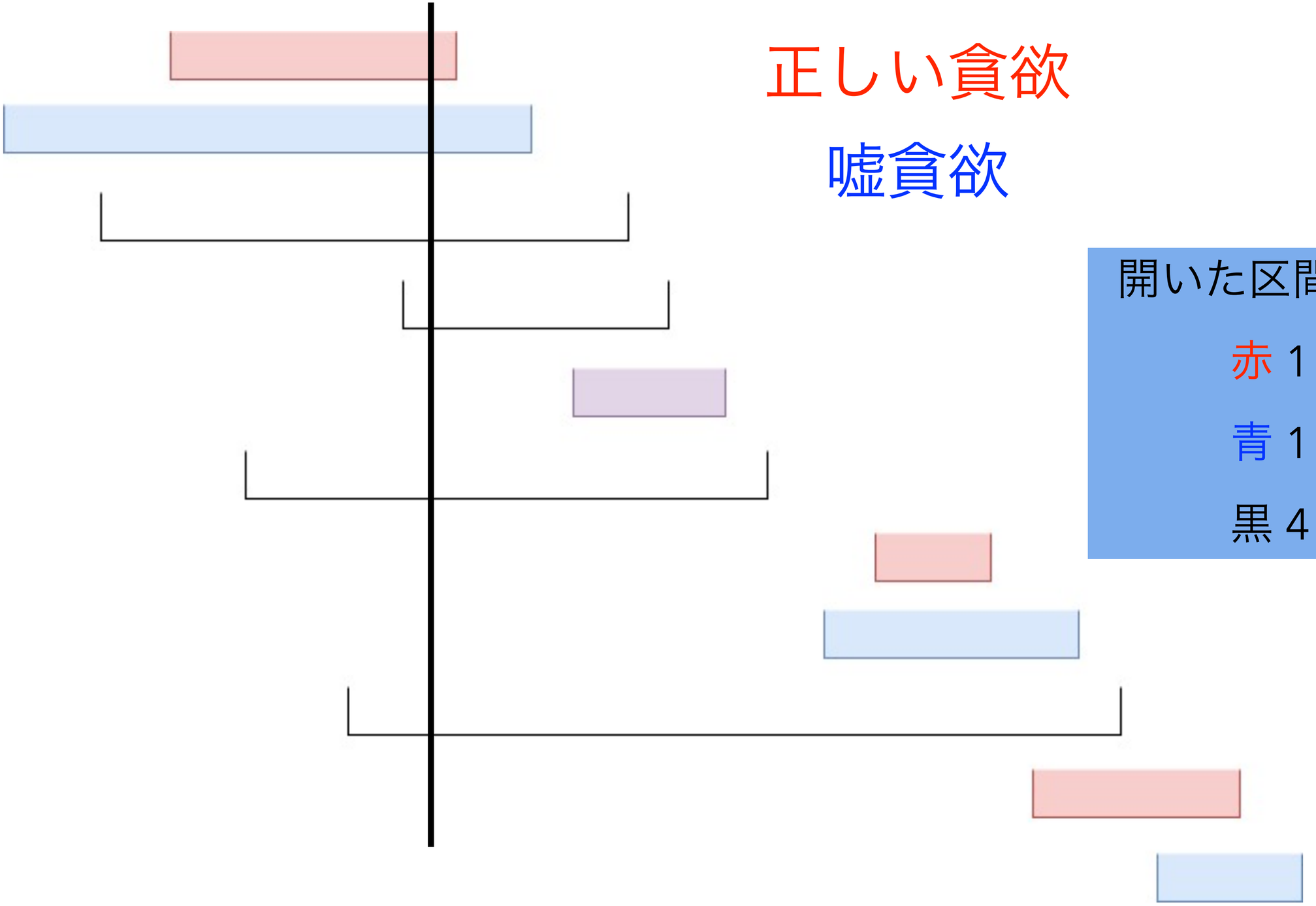

正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 0


正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 1


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 1


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 2


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 3



正しい貪欲

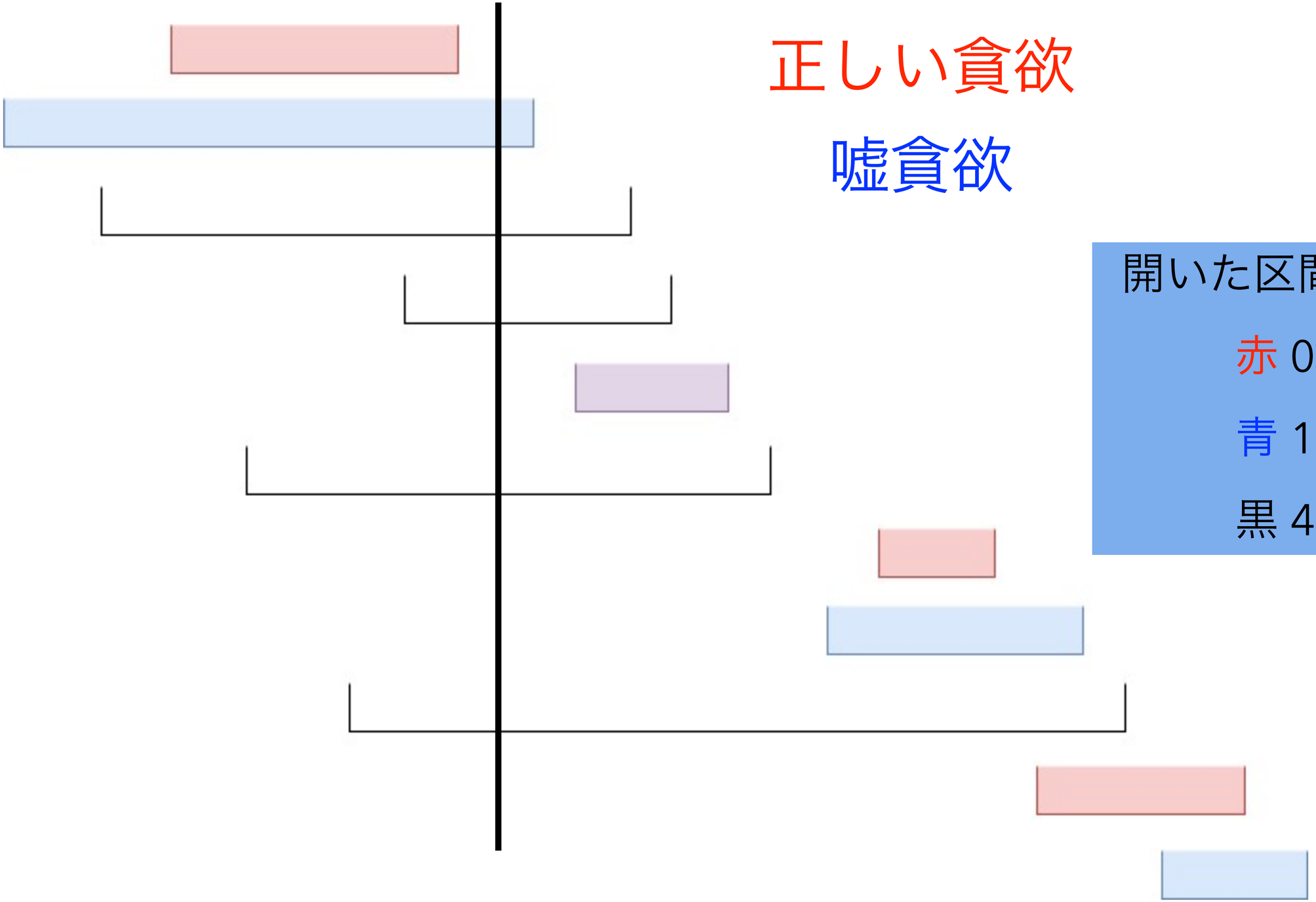
嘘貪欲

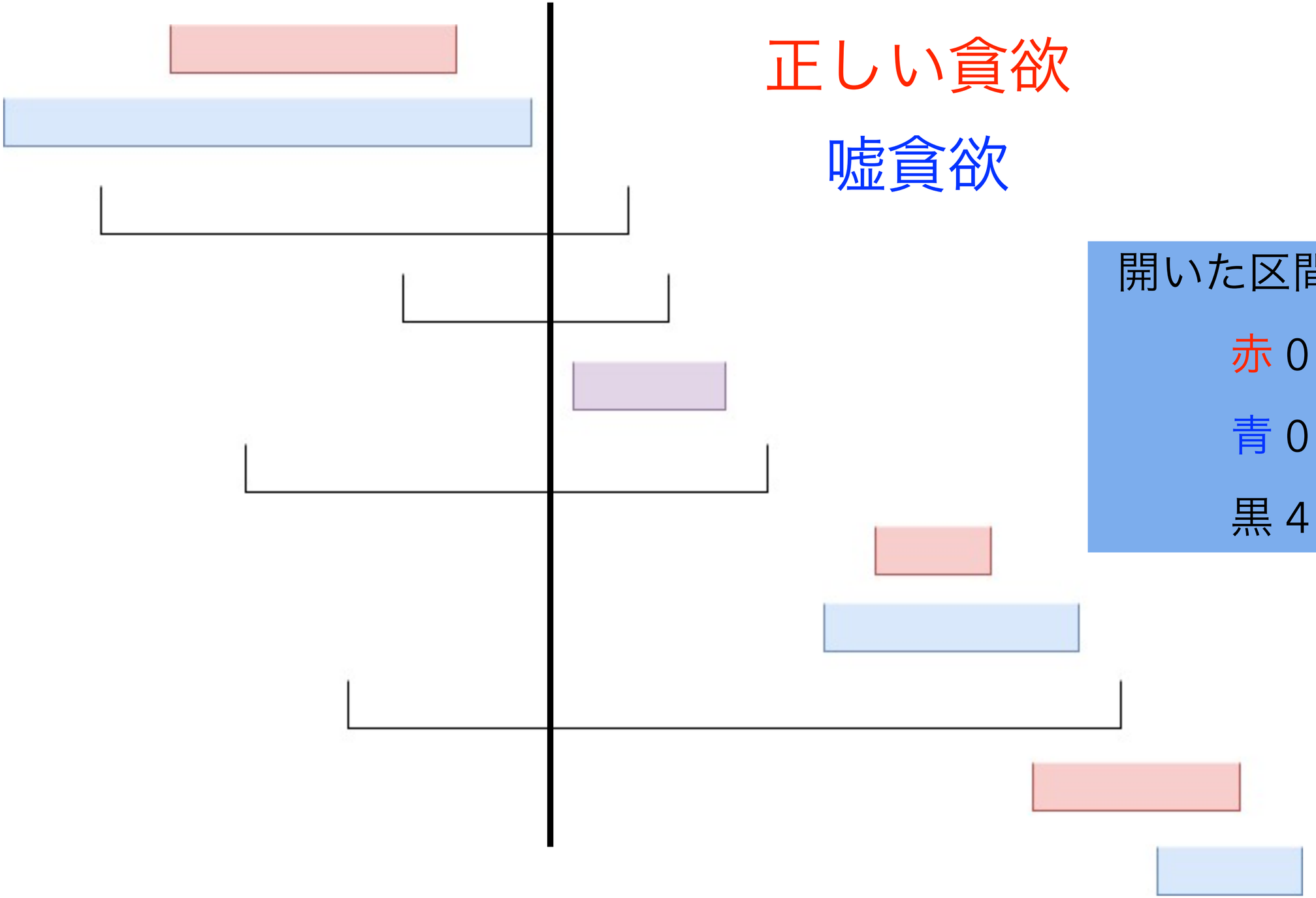
開いた区間の数

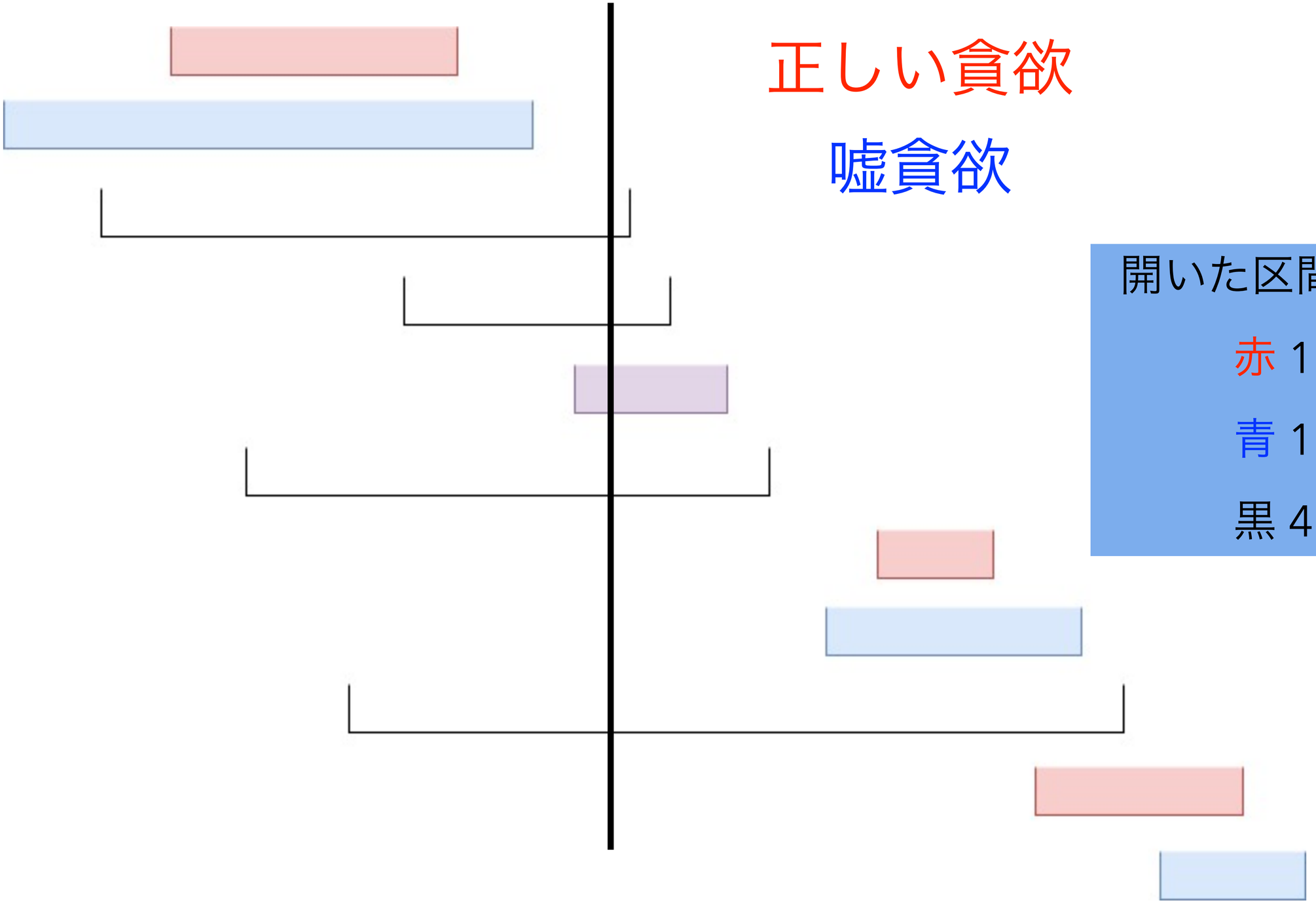
赤 1

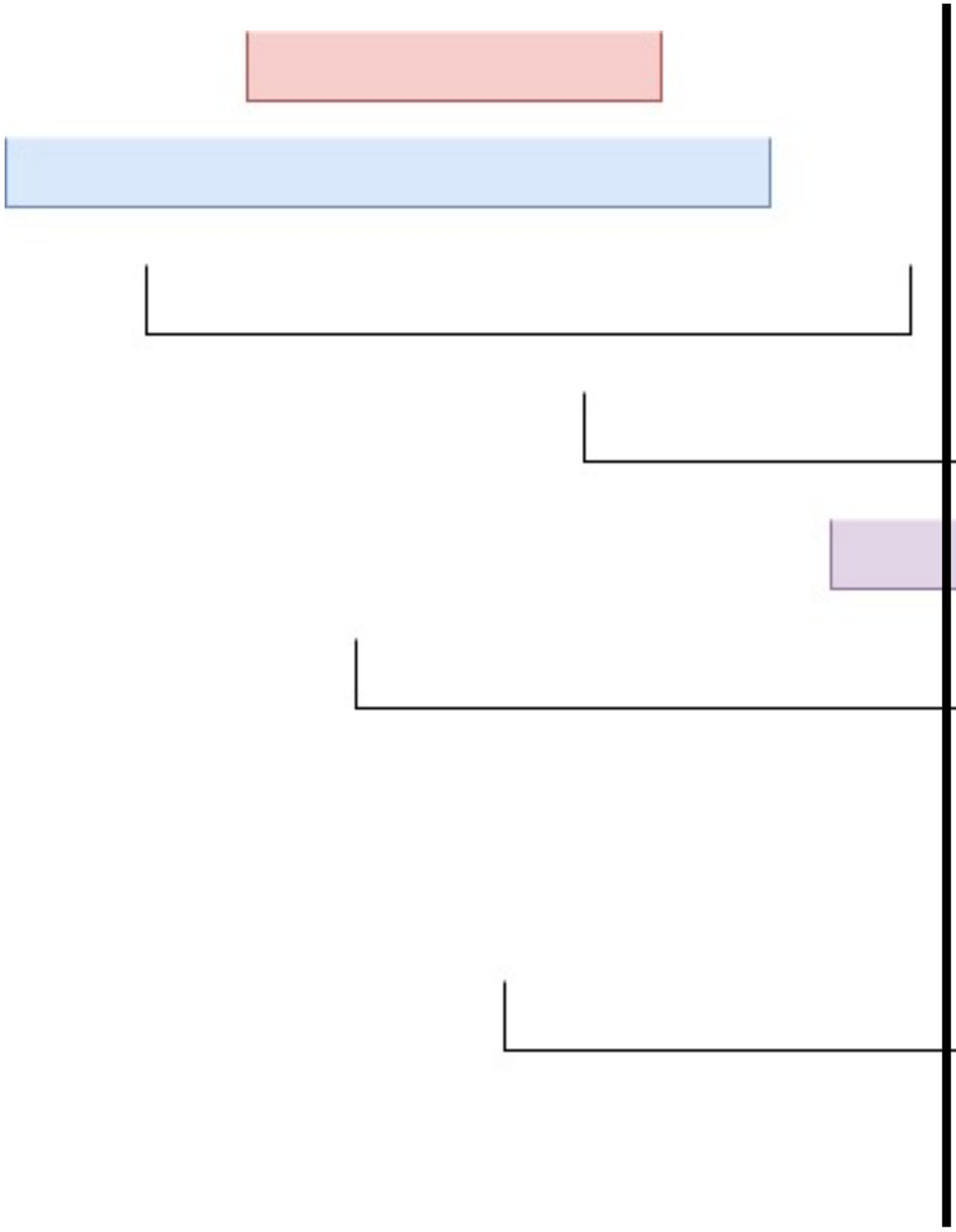
青 1

黒 4


正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 4


正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 0
黒 4


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 4



正しい貪欲

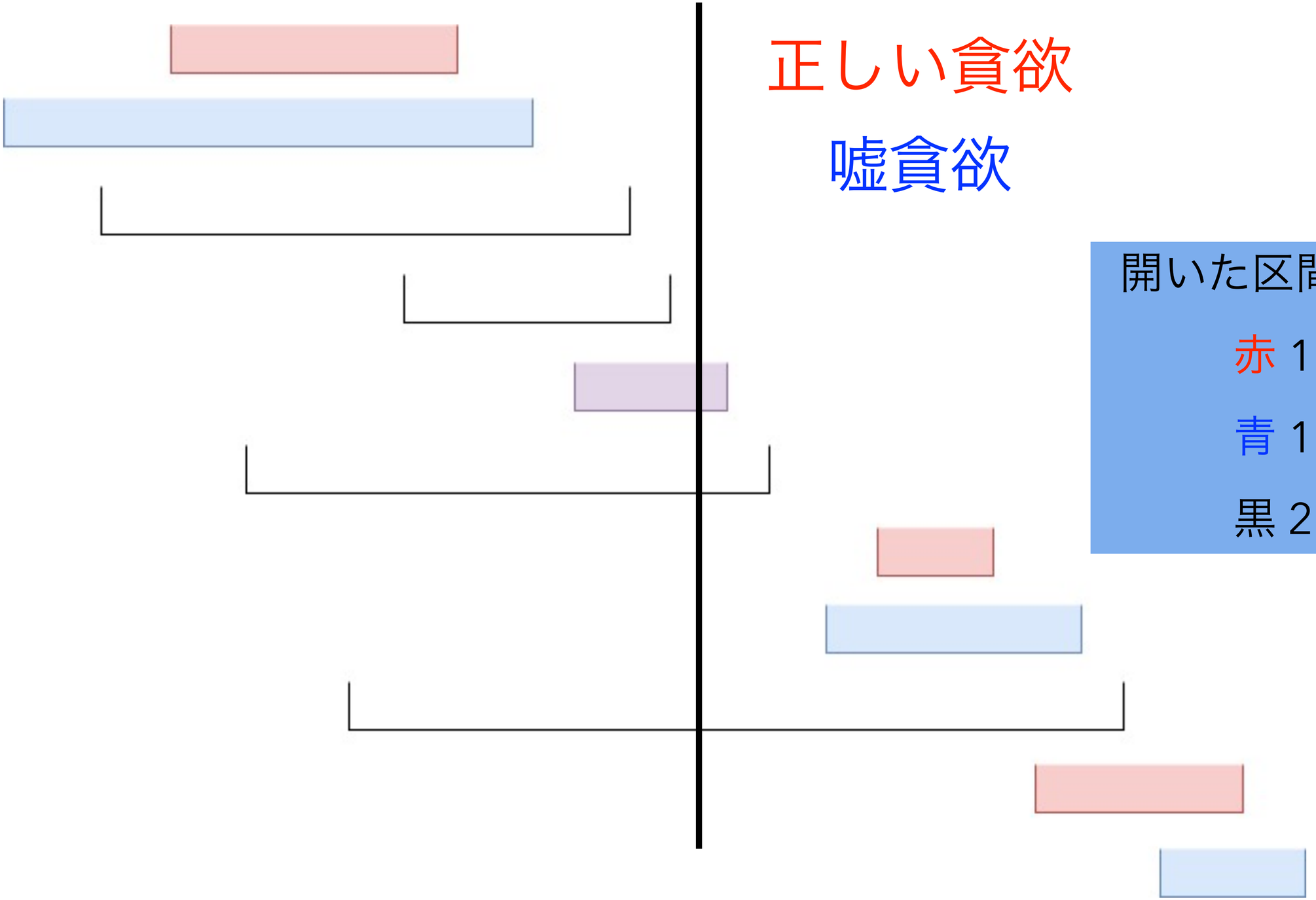
嘘貪欲

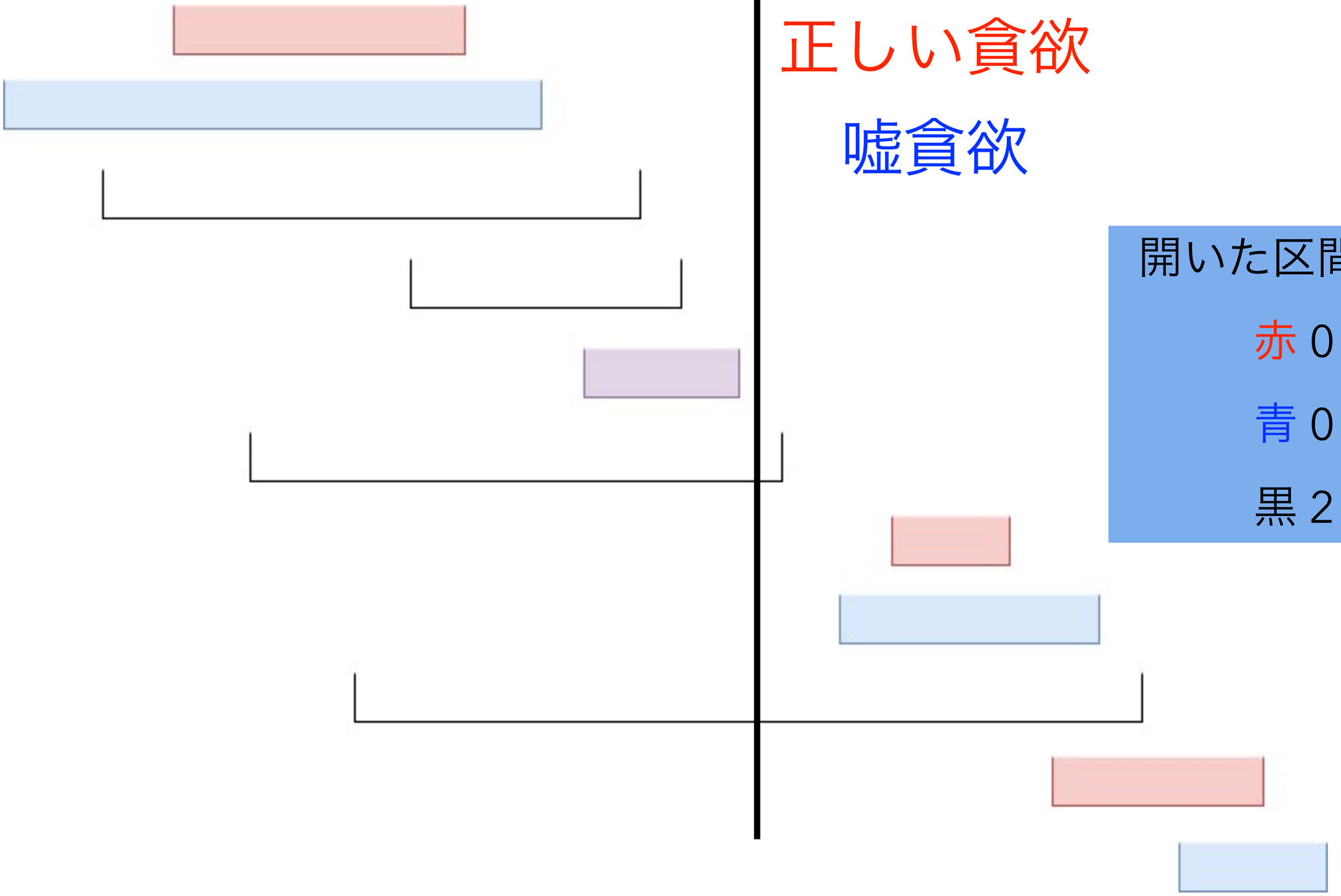
開いた区間の数

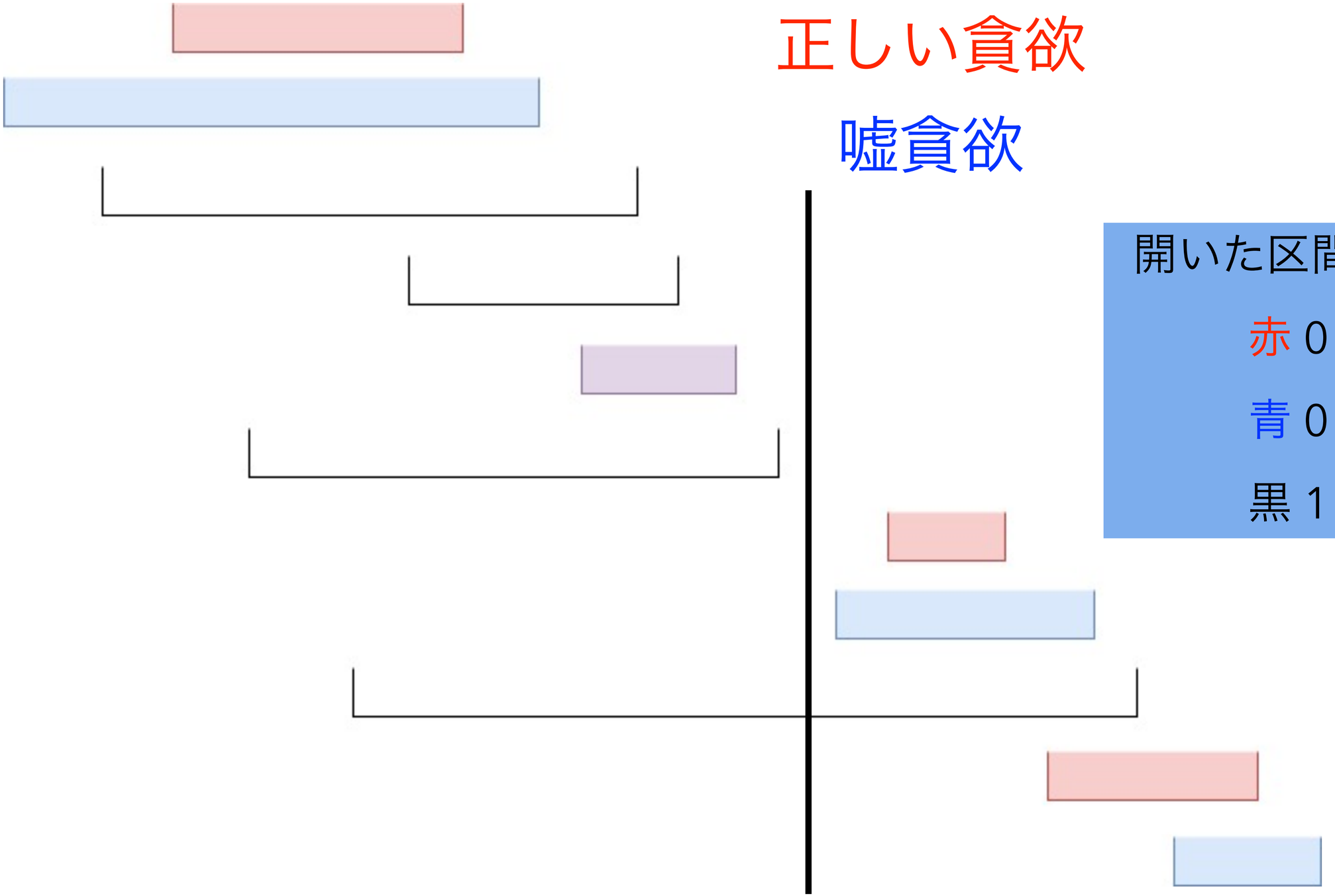
赤 1

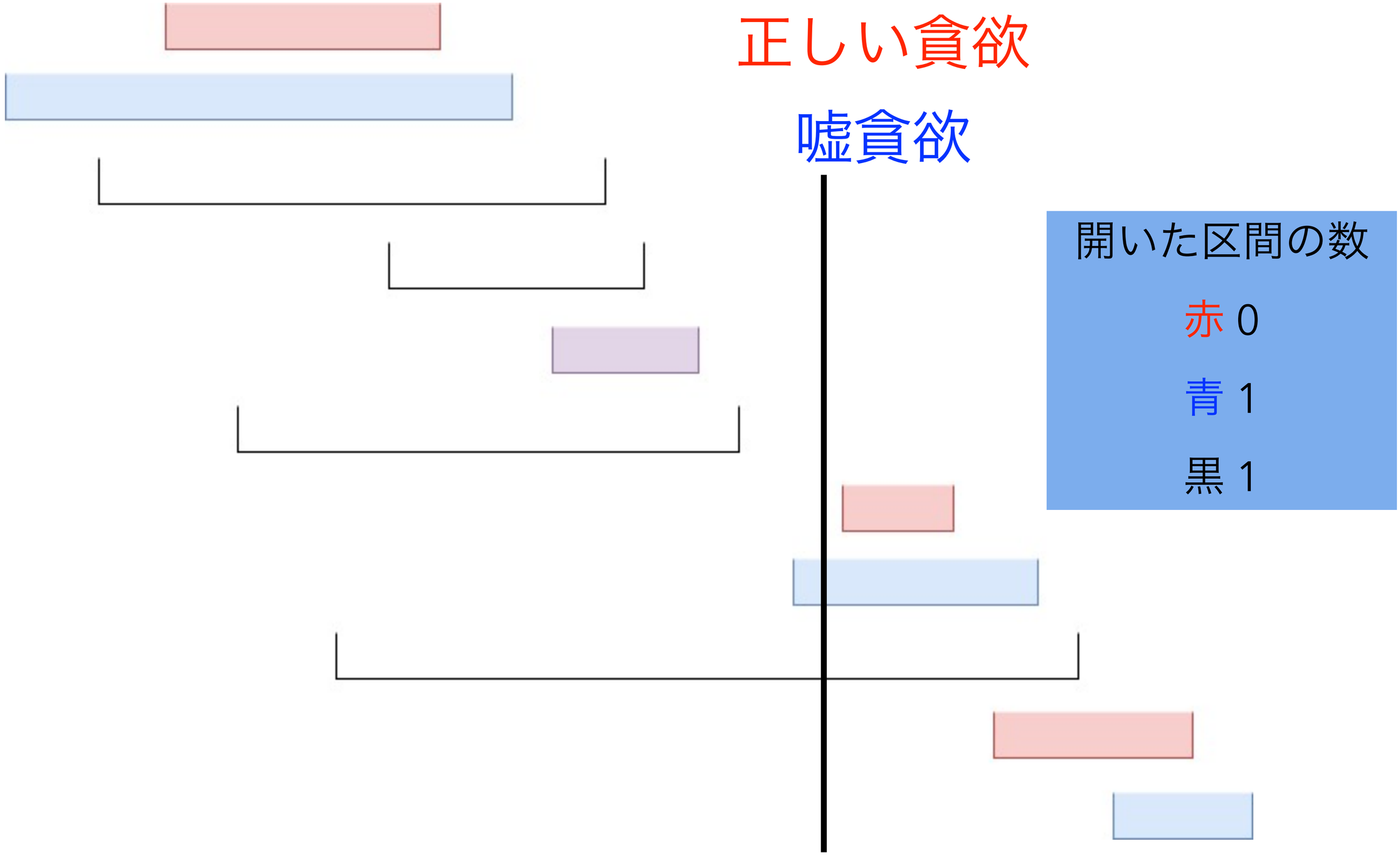
青 1

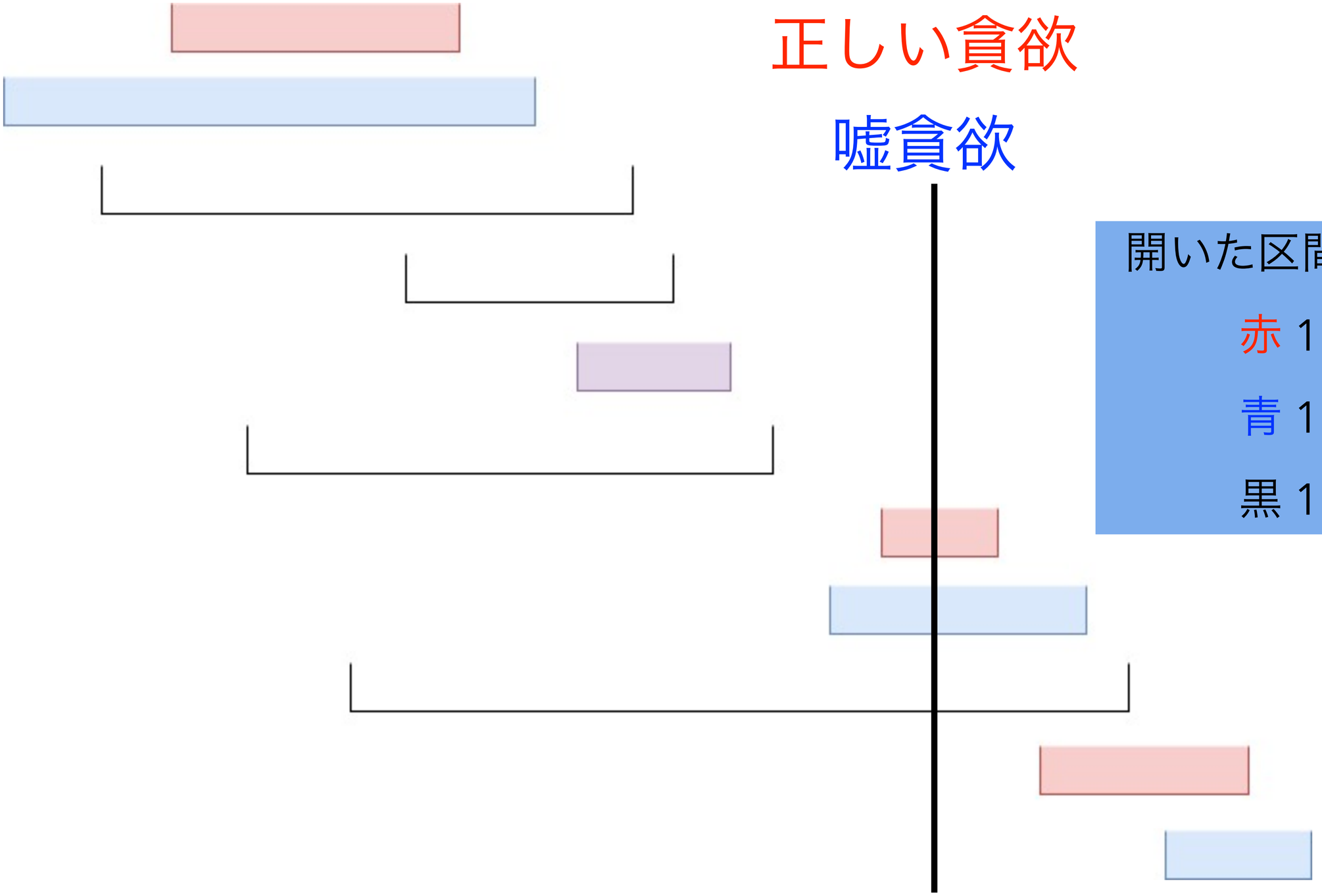
黒 3

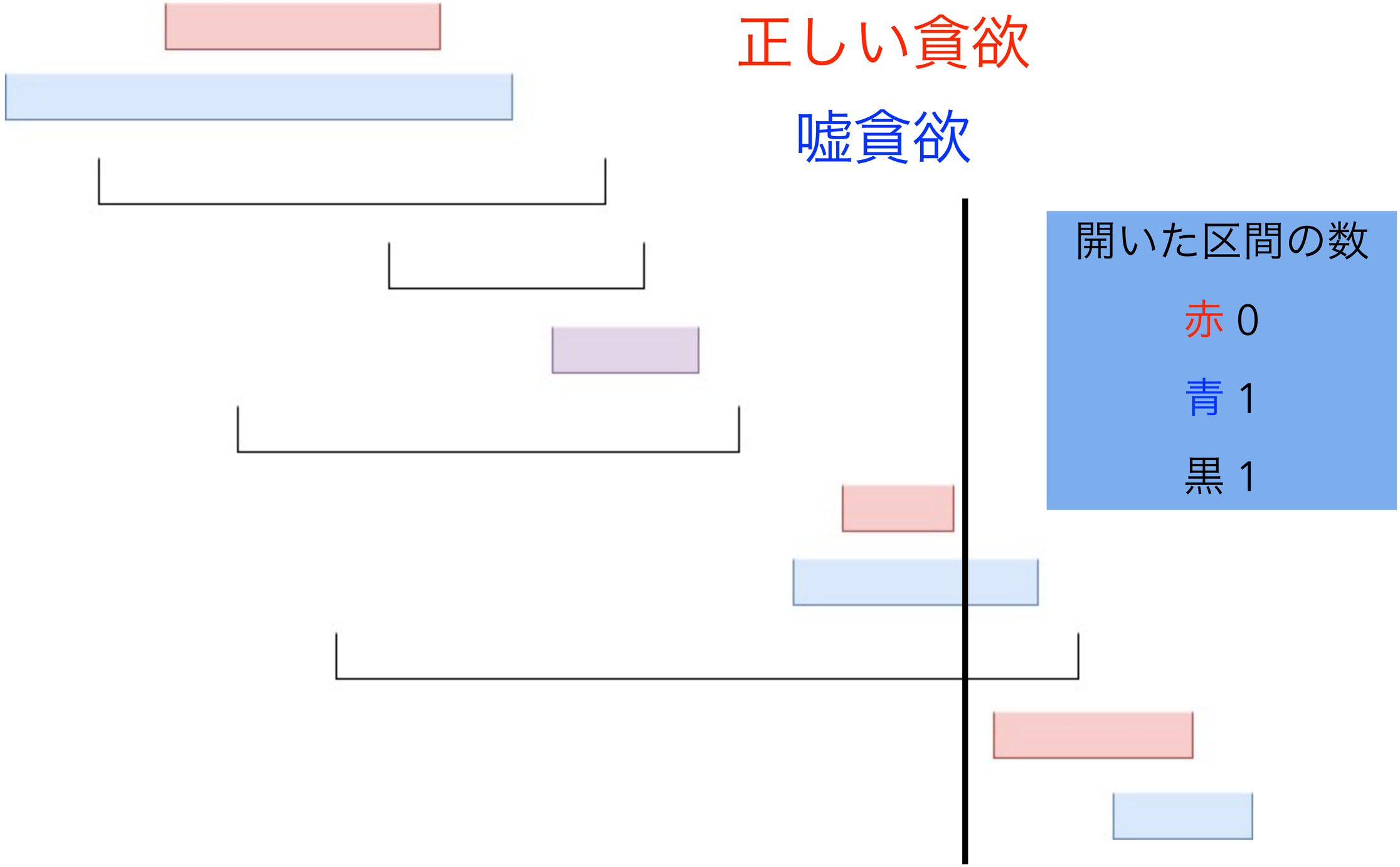

正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 2

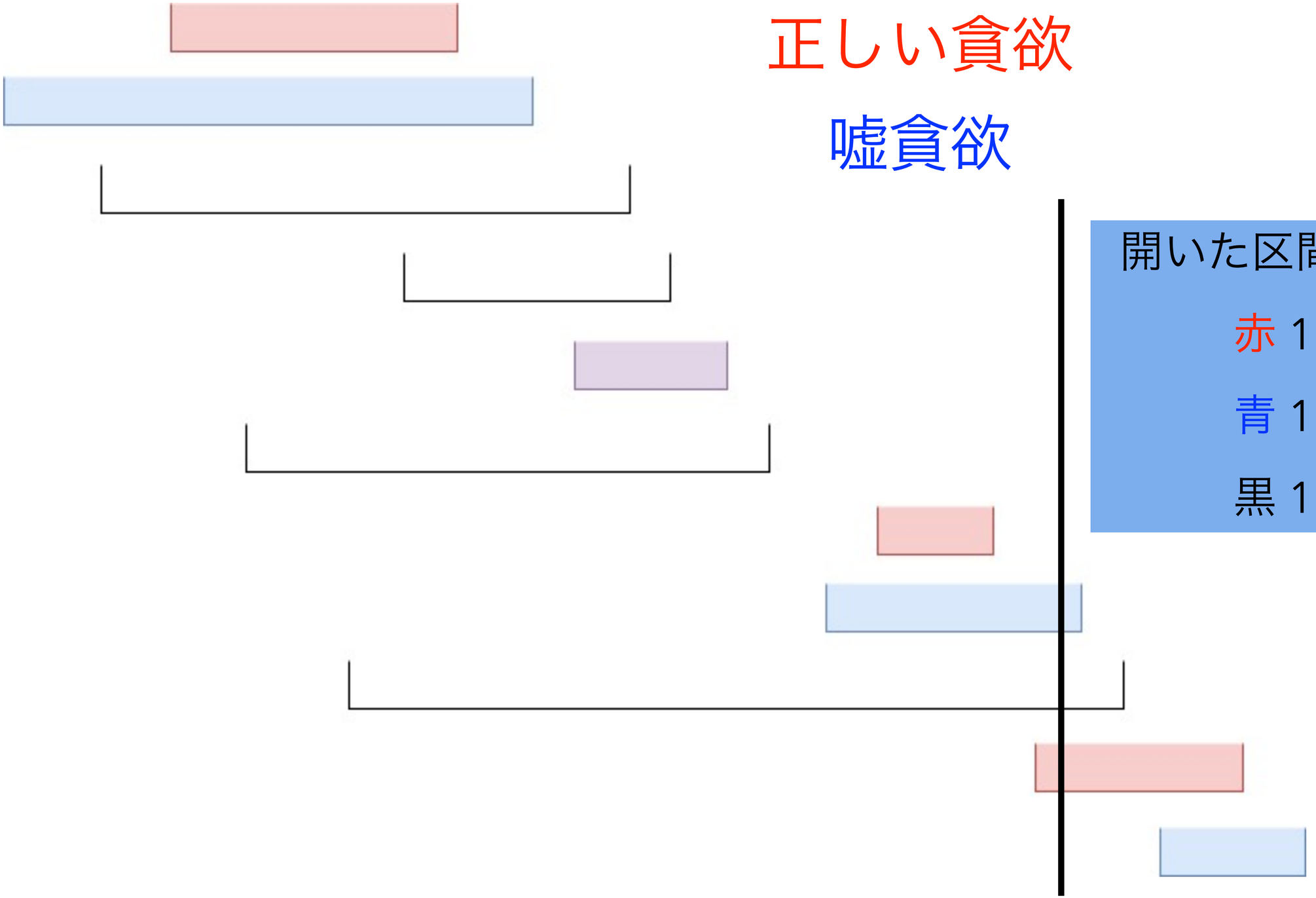

正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 0
黒 2

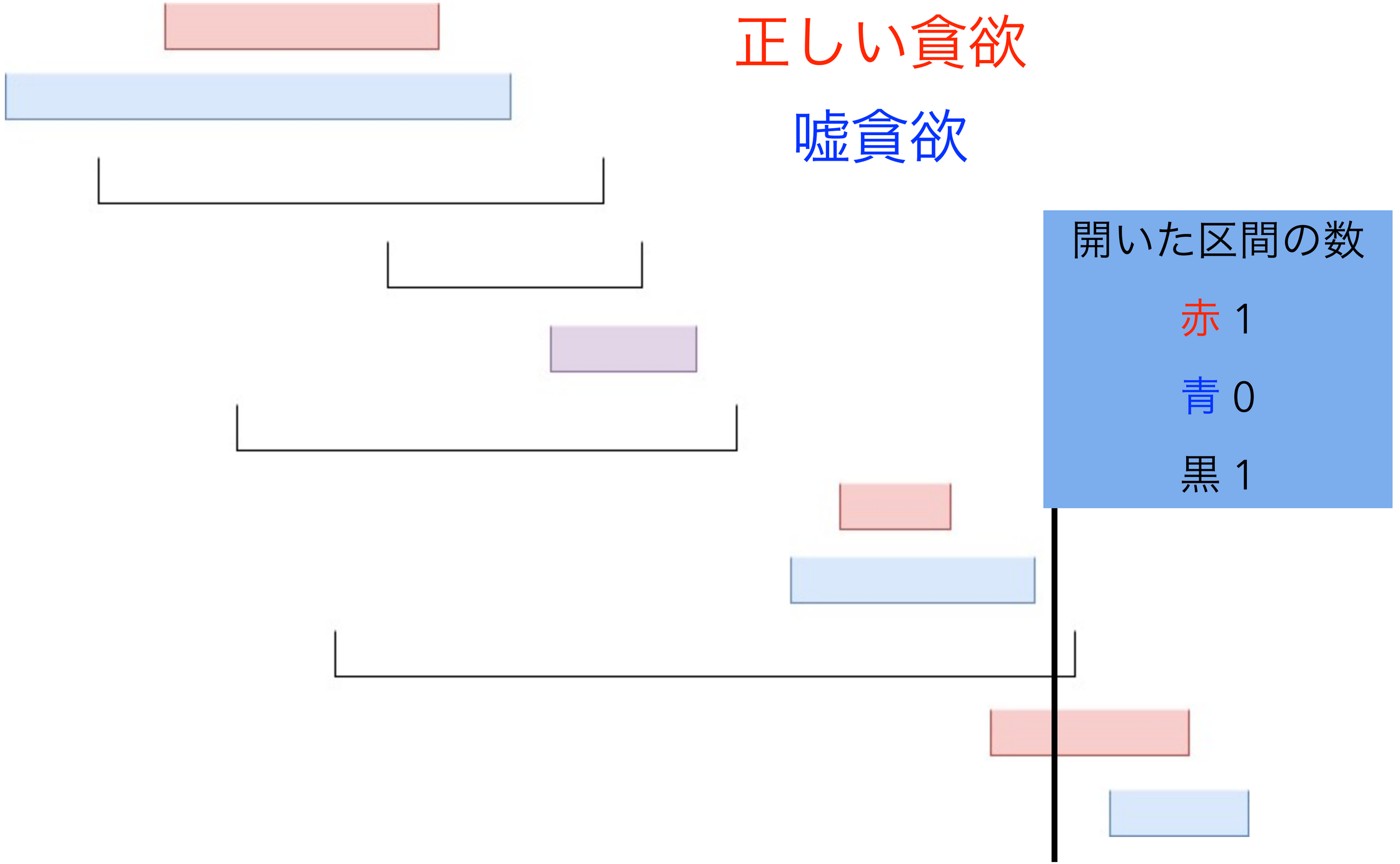

正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 0
黒 1

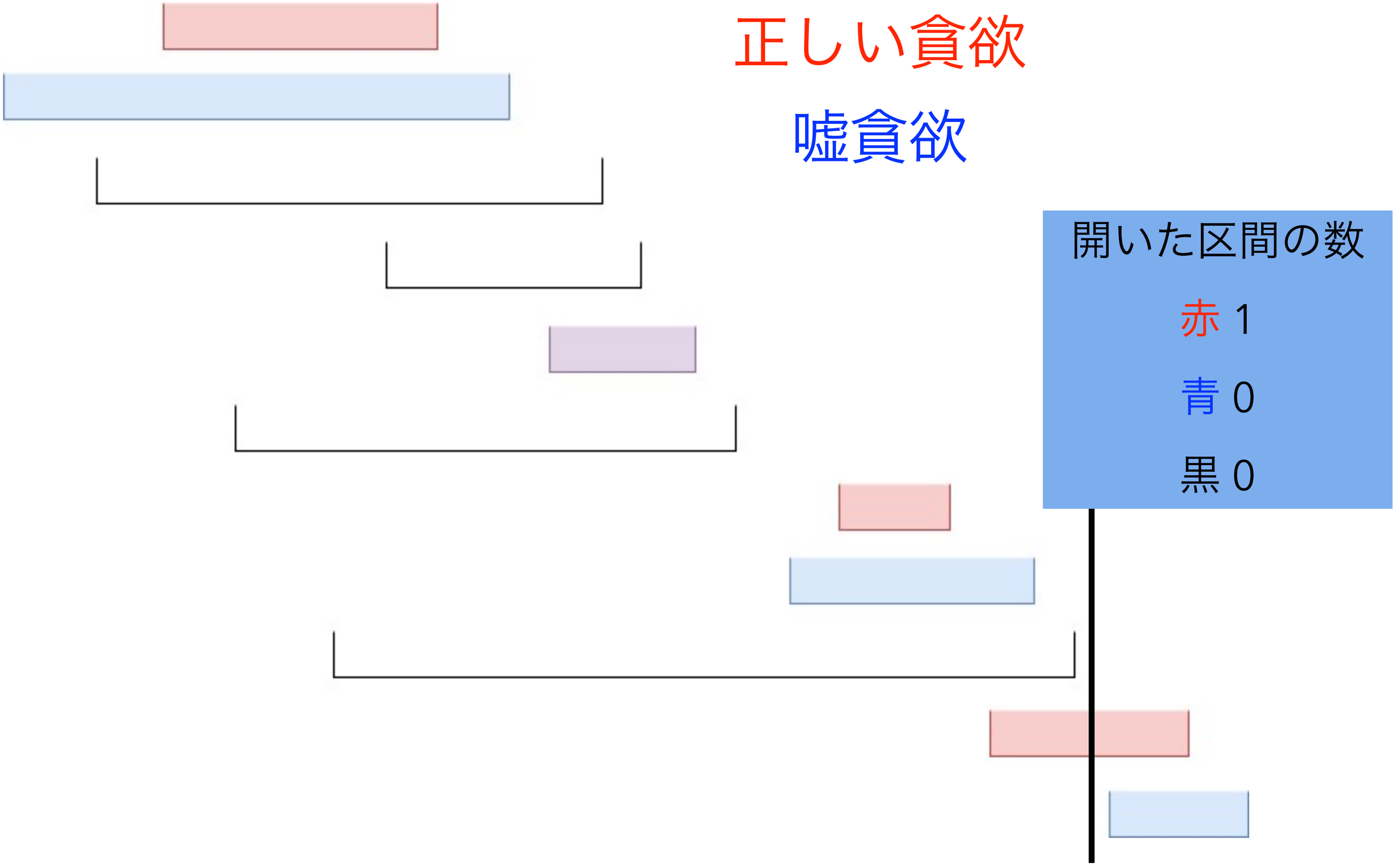

正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 1

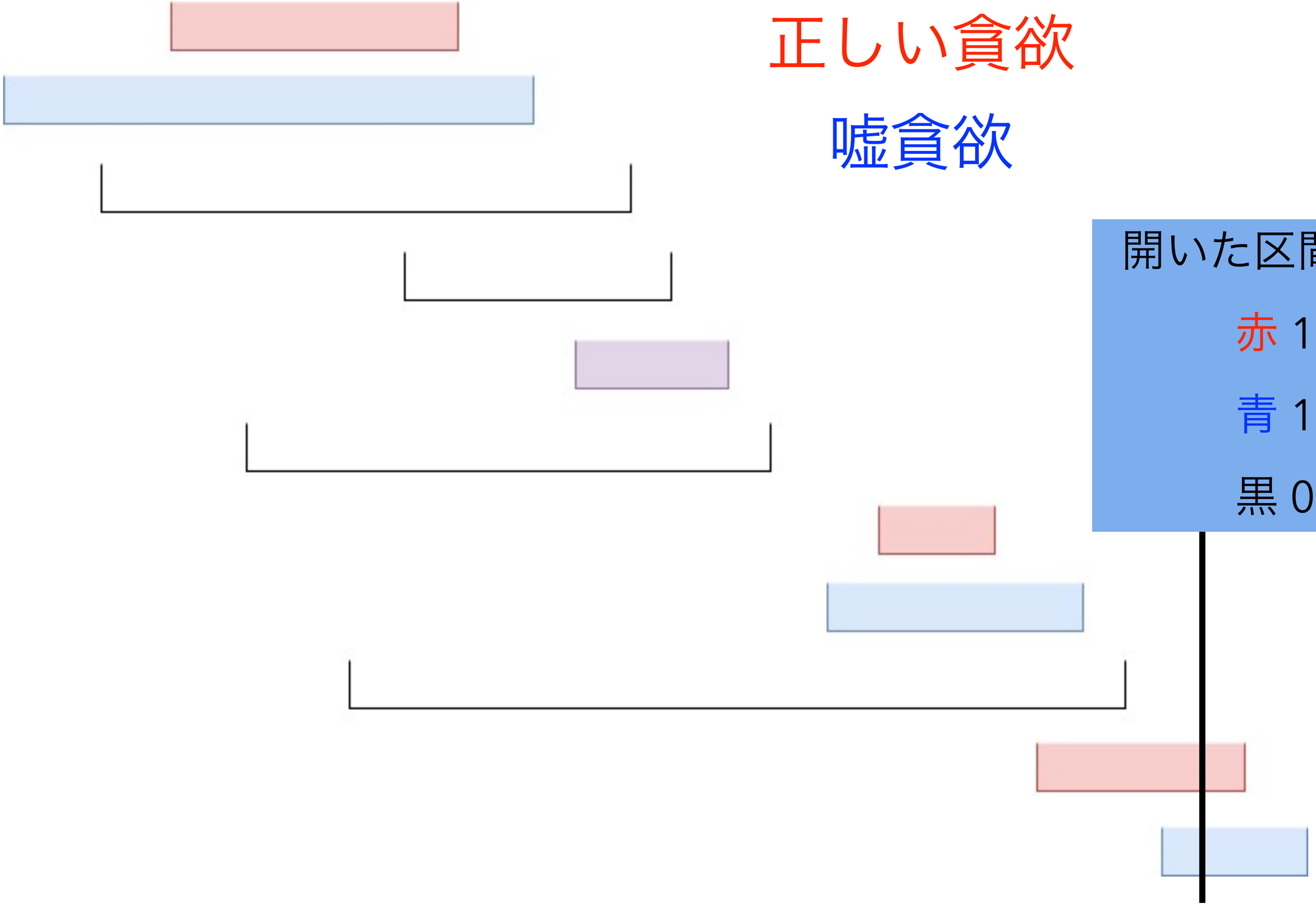

正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 1

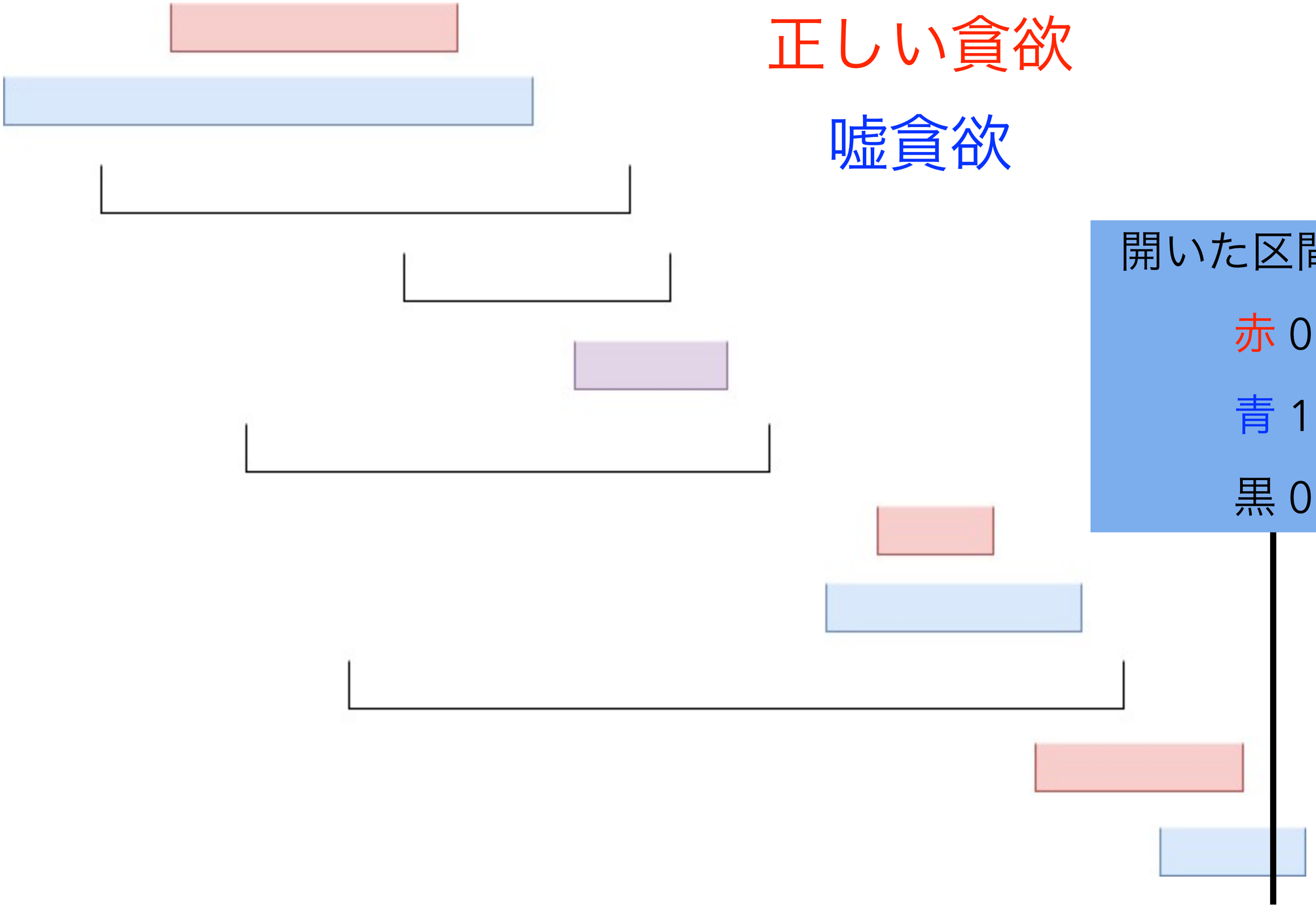

正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 1

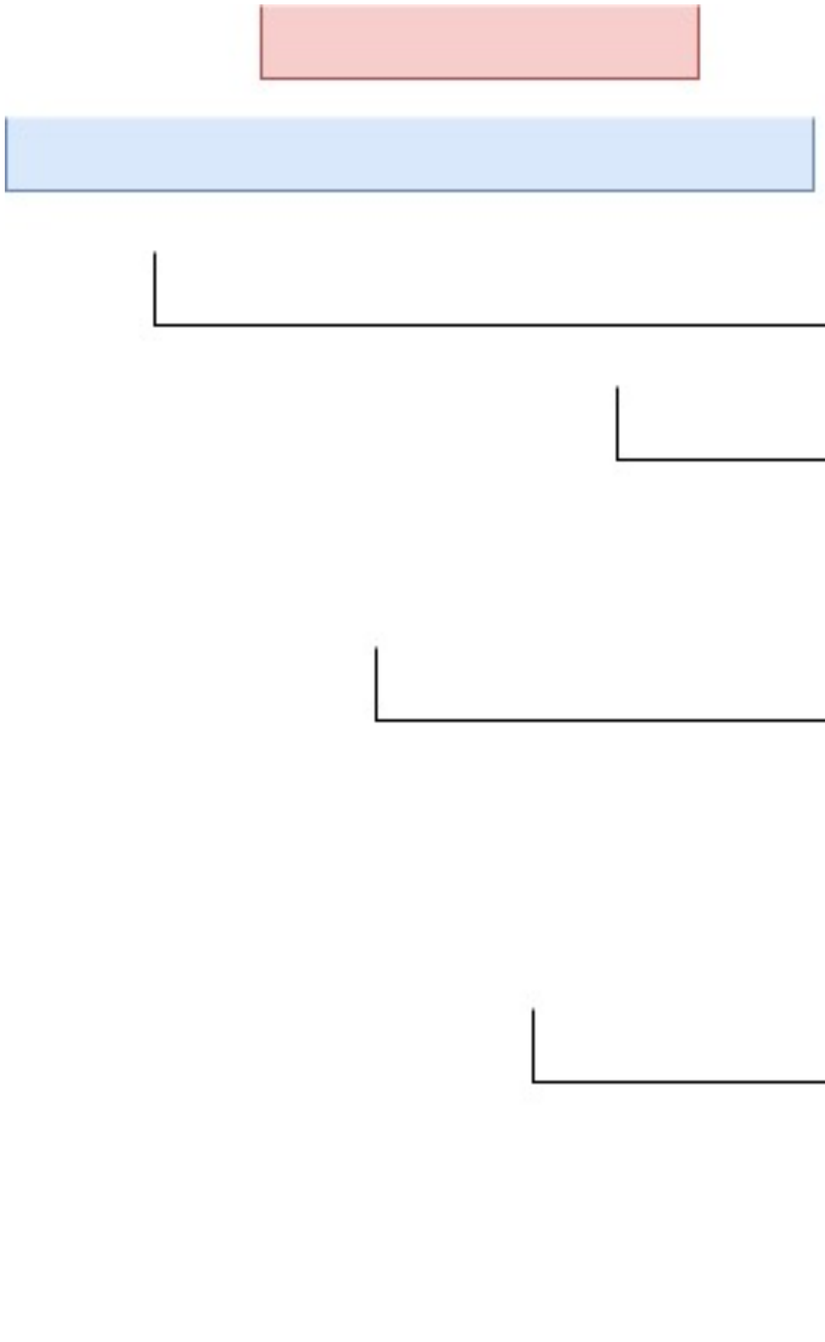

正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 1


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 0
黒 1


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 0
黒 0


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 0


正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 0



正しい貪欲

嘘貪欲

開いた区間の数

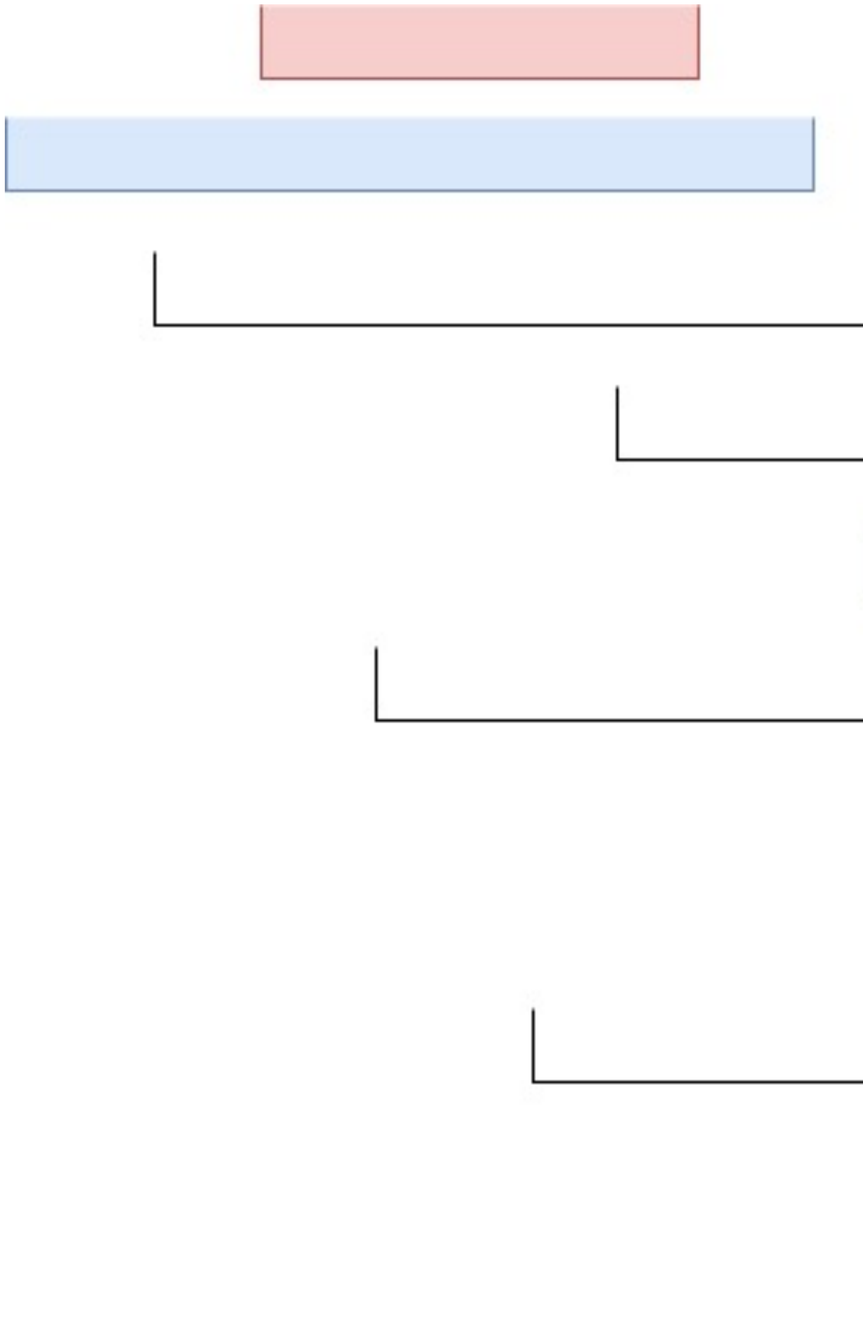
赤 0

青 0

黒 0

小課題 4 ($N \leqq 300$)

- 持つべき情報は、
  - 1 から $2N$ までの数のうち、どこまで割り当てたか
  - 開いた黒い区間の数
  - $\uparrow$ のうち、最後に赤い区間を閉じたときより後に開けたものの数
  - 赤、青の区間に関する “状態”
- 状態って？
  - ざっくりいうと「赤い区間が開いているかどうか」「青い区間が開いているかどうか」だが、それだけでは足りない



正しい貪欲

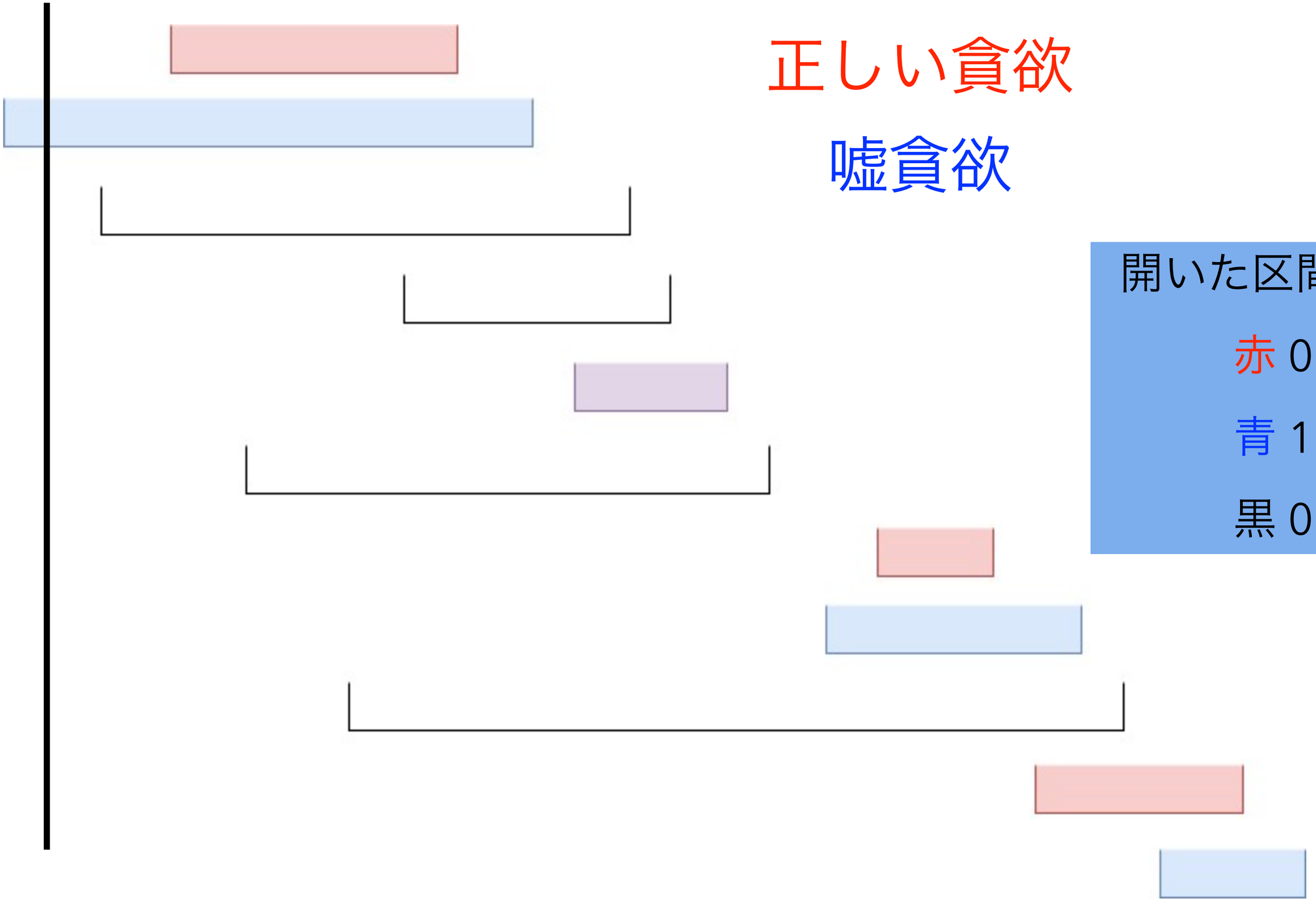
嘘貪欲

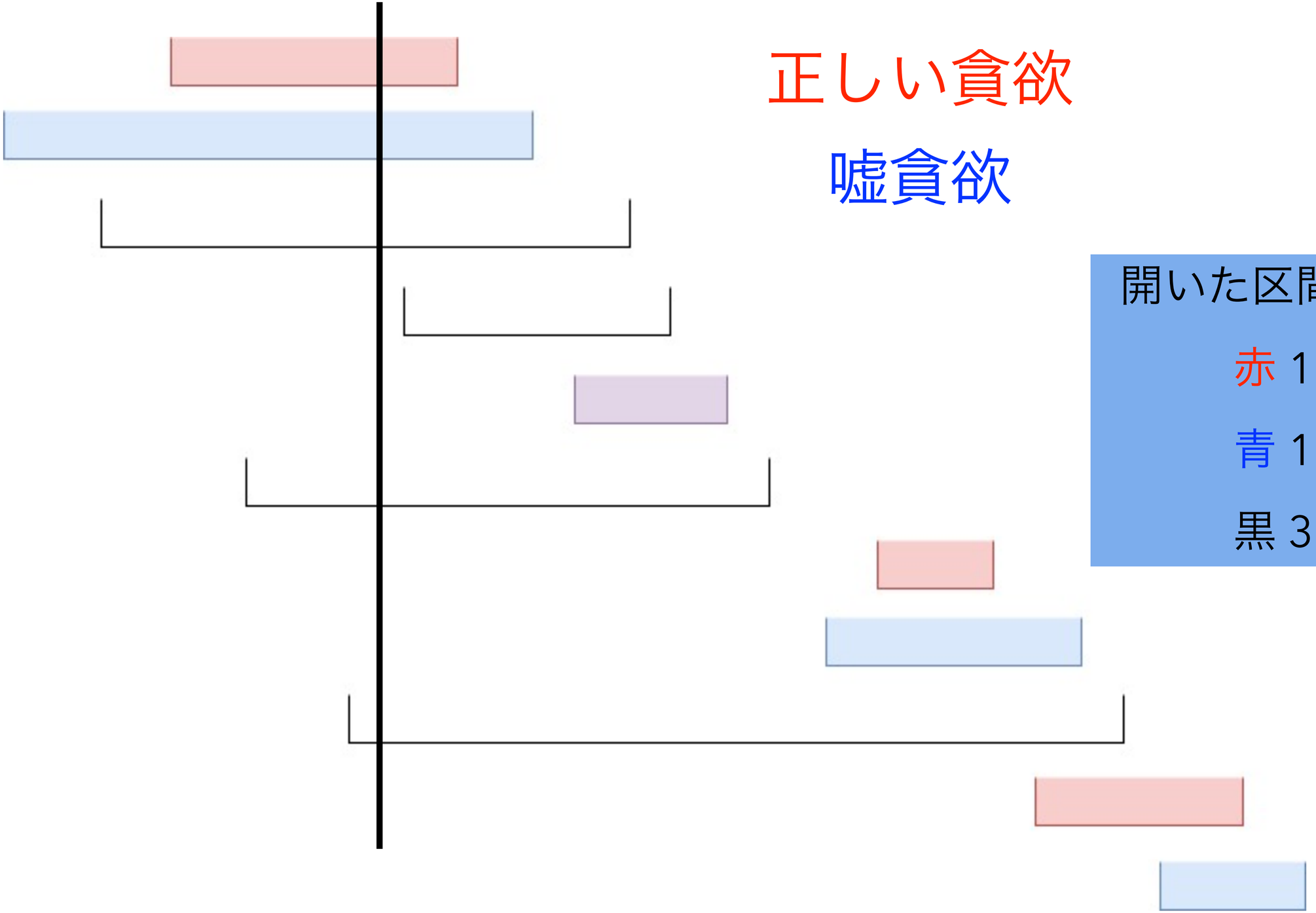
開いた区間の数

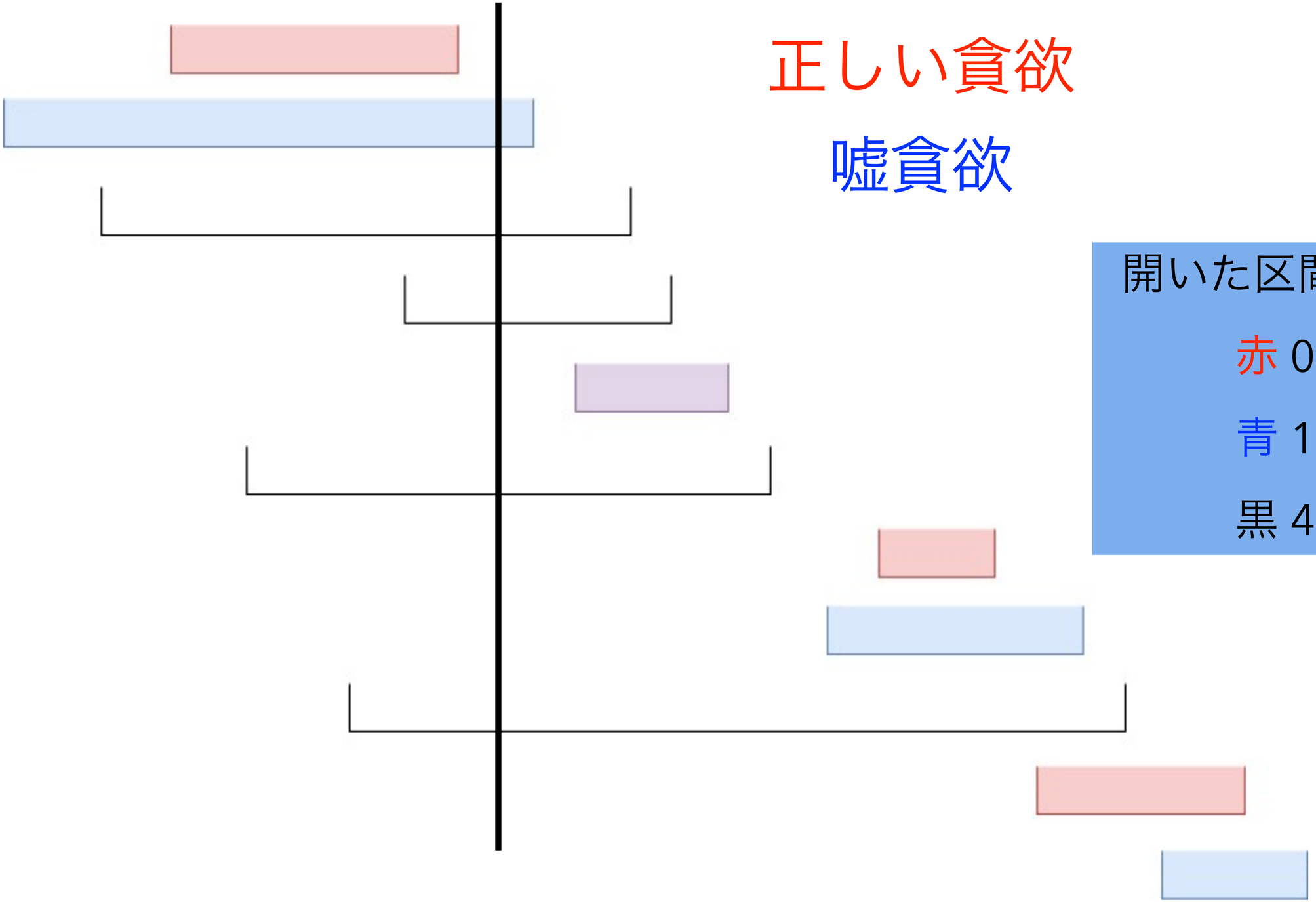
赤 0

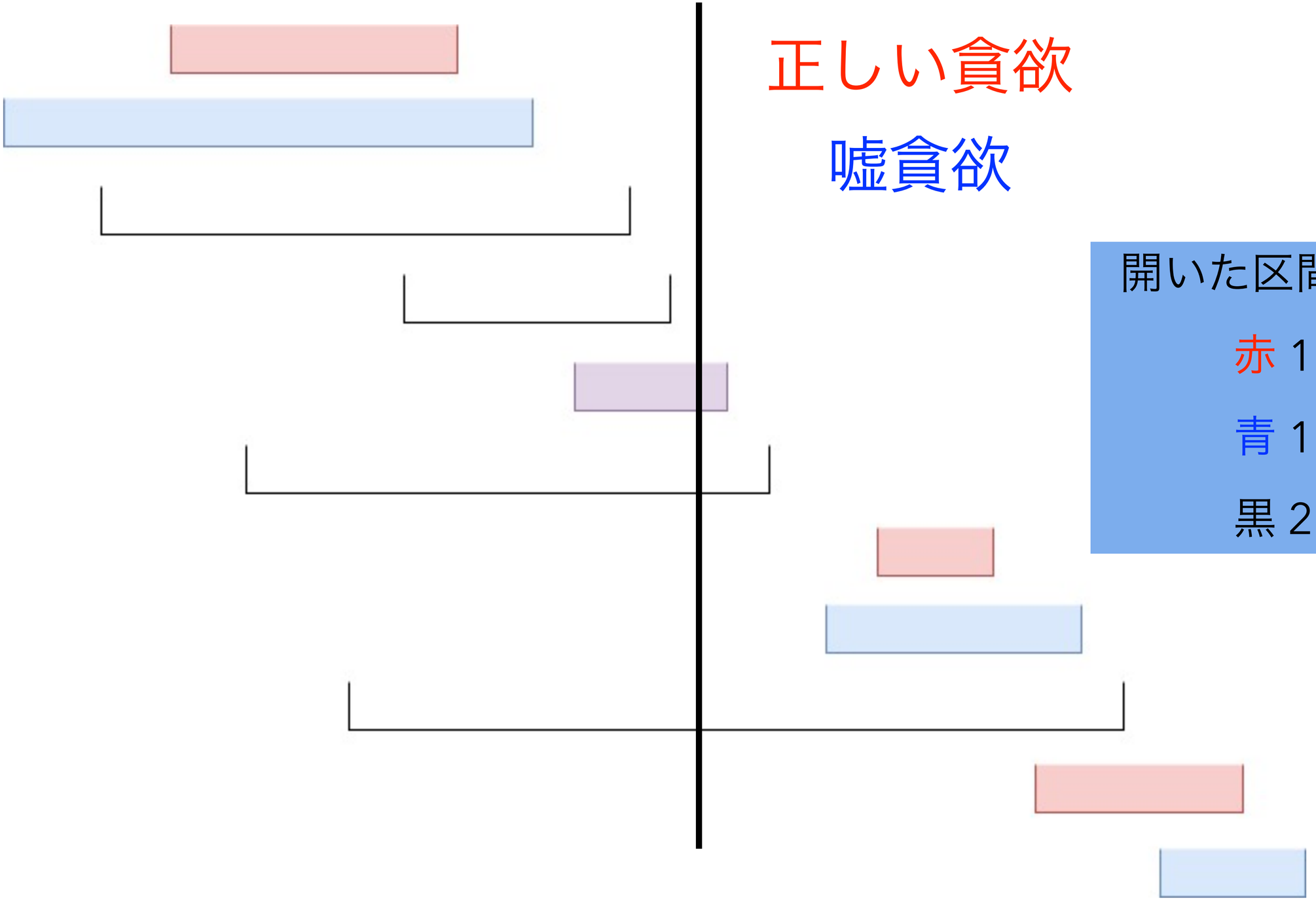
青 0

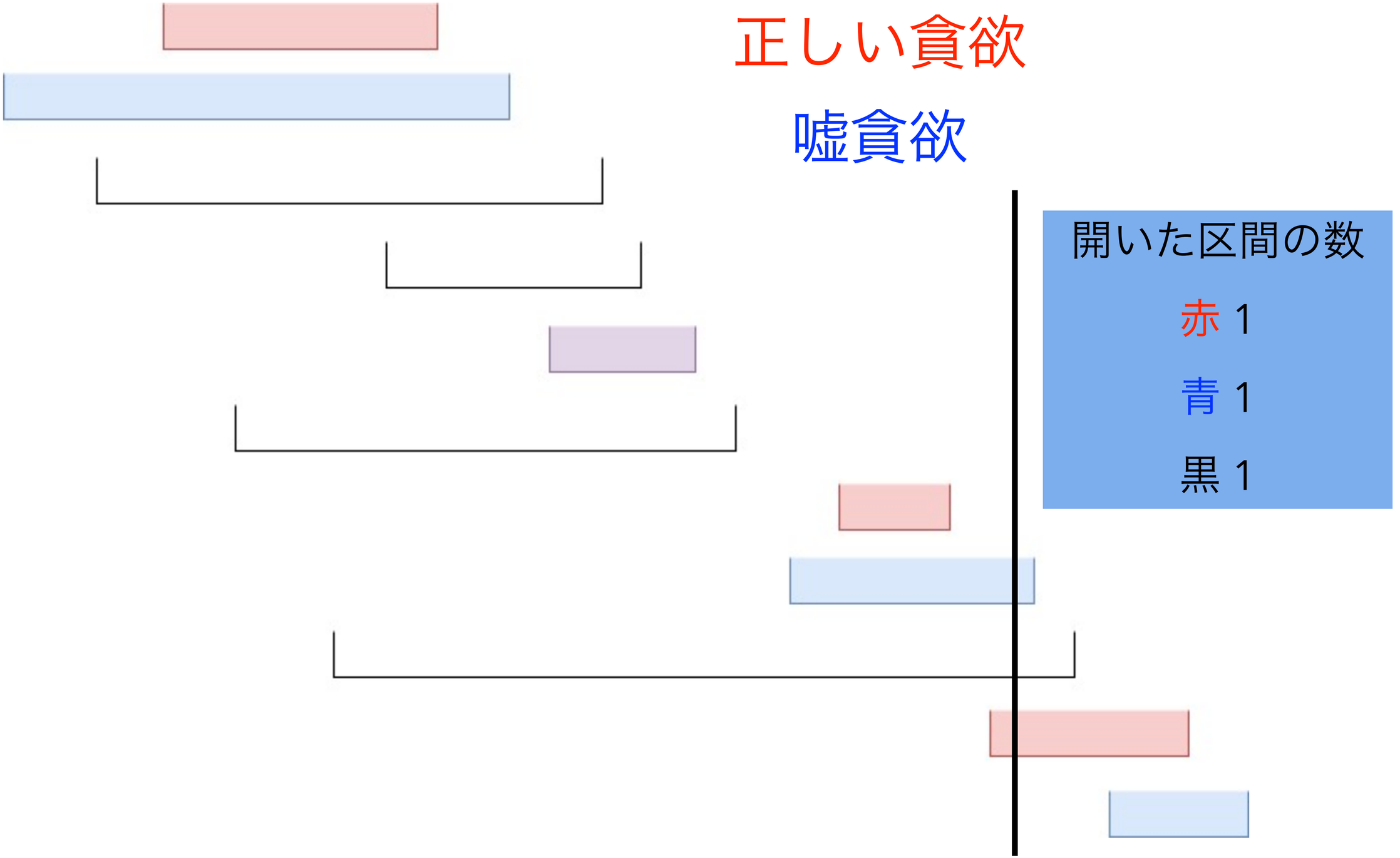
黒 0


正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 0


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 3


正しい貪欲
嘘貪欲
開いた区間の数
赤 0
青 1
黒 4


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 2


正しい貪欲
嘘貪欲
開いた区間の数
赤 1
青 1
黒 1

小課題 4 ($N \leqq 300$)

- 全部で 6 通りの状態がある
- 工夫すればもっと状態をまとめられるが、いずれにせよ場合分けが必要
- 焦らず丁寧に実装しましょう
- 計算量 $O(N^3)$

ここまでで 51 点

小課題 5 ($N \leqq 3000$)

- 再び挿入 DP に戻ります
- さっきは B_i の昇順に区間を見たが、今度は A_i の降順に区間を見る
- 反転してるだけで実質同じじゃないの？
  - 嘘貪欲は区間の始点をベースに動いているので、実は対称性がない
  - 実際、区間の始点に関する制約の方が終点より “厳しい”

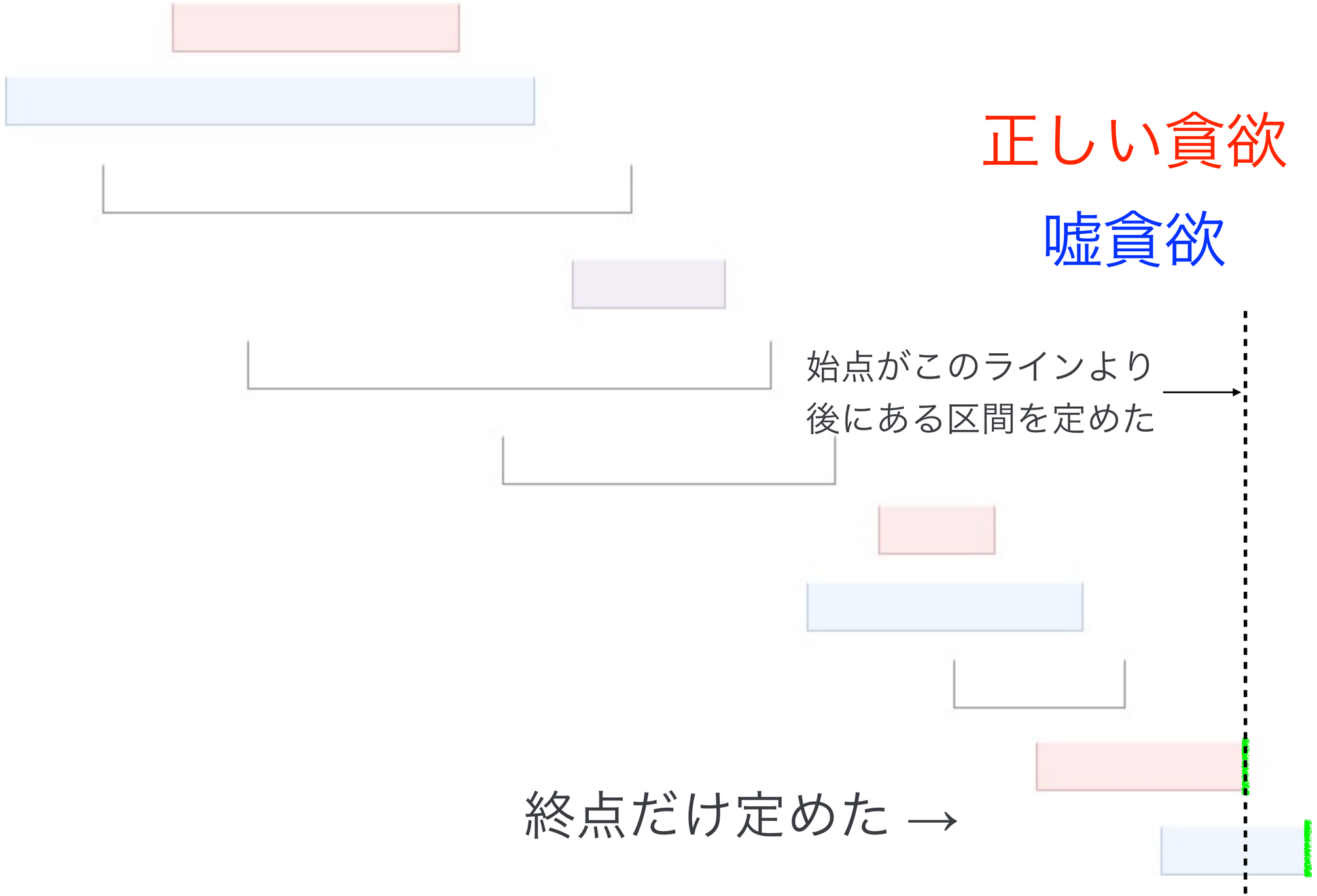
小課題 3 ($N \leqq 30$)

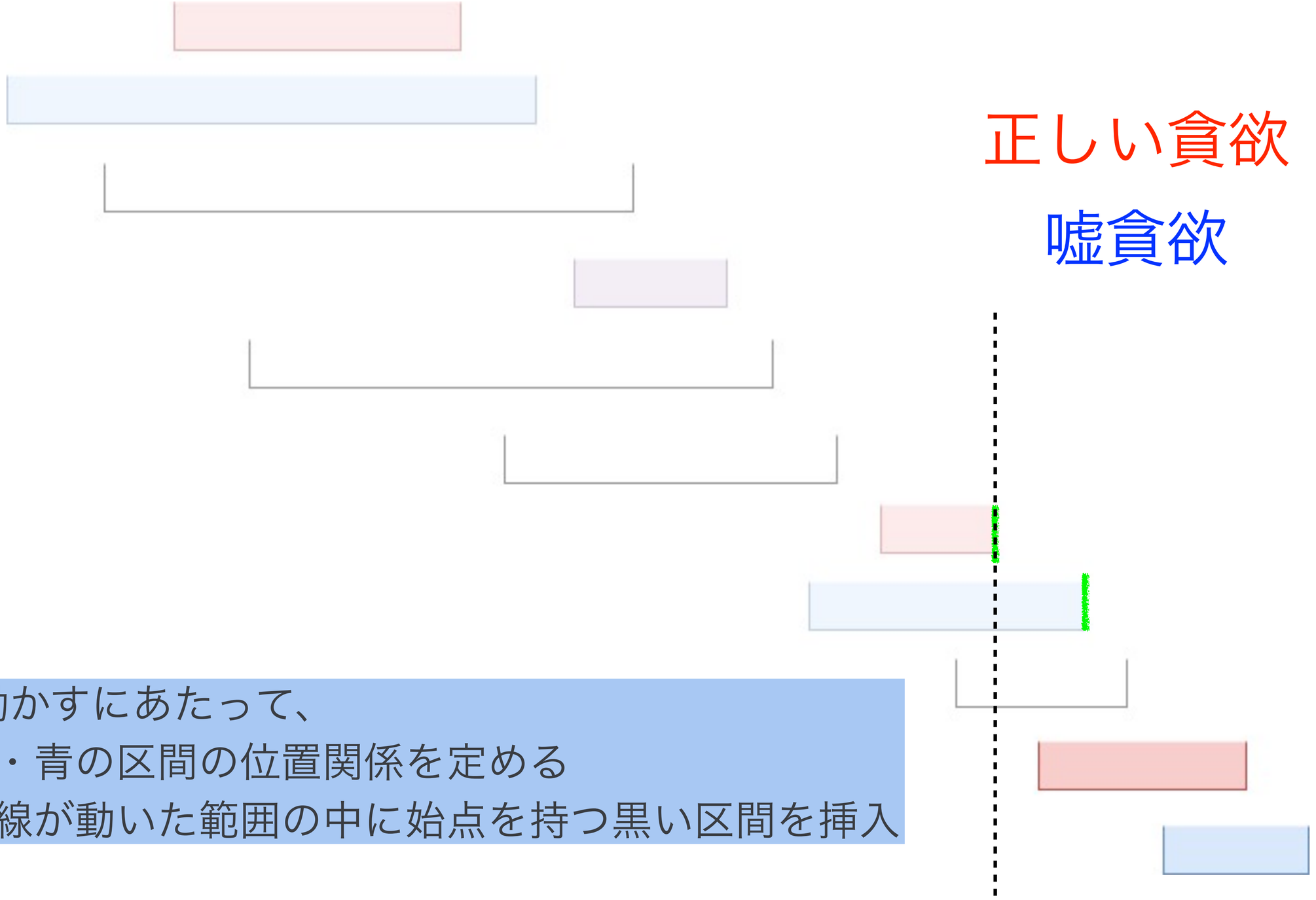
- B_i の昇順に区間を定める挿入 DP
- 以下の情報を持つ
  - 定めた区間の個数
  - 選んだ赤い区間の数 − 青い区間の数 (0 or 1)
  - 最後に選んだ赤い区間は何番目か
  - 最後に選んだ青い区間は何番目か
  - 挿入できる箇所のうち、青い区間の中に含まれるのは何箇所か

これがムダ

小課題 5 ($N \leqq 3000$)

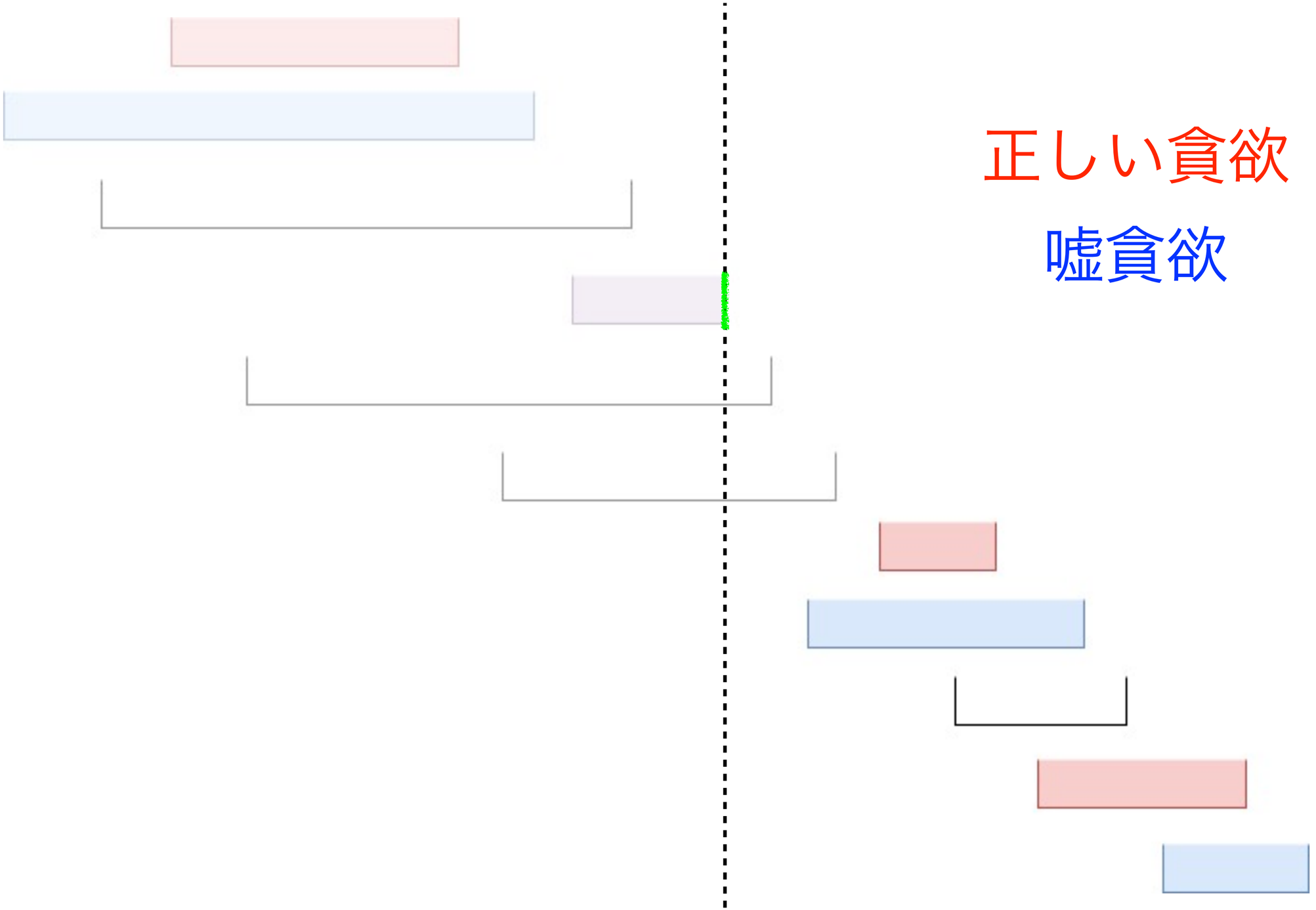
- 本当にすべて A_i の降順に区間を見るわけではなくて、赤い区間・青い区間の端点は特別扱いします
- 今回も図で説明します

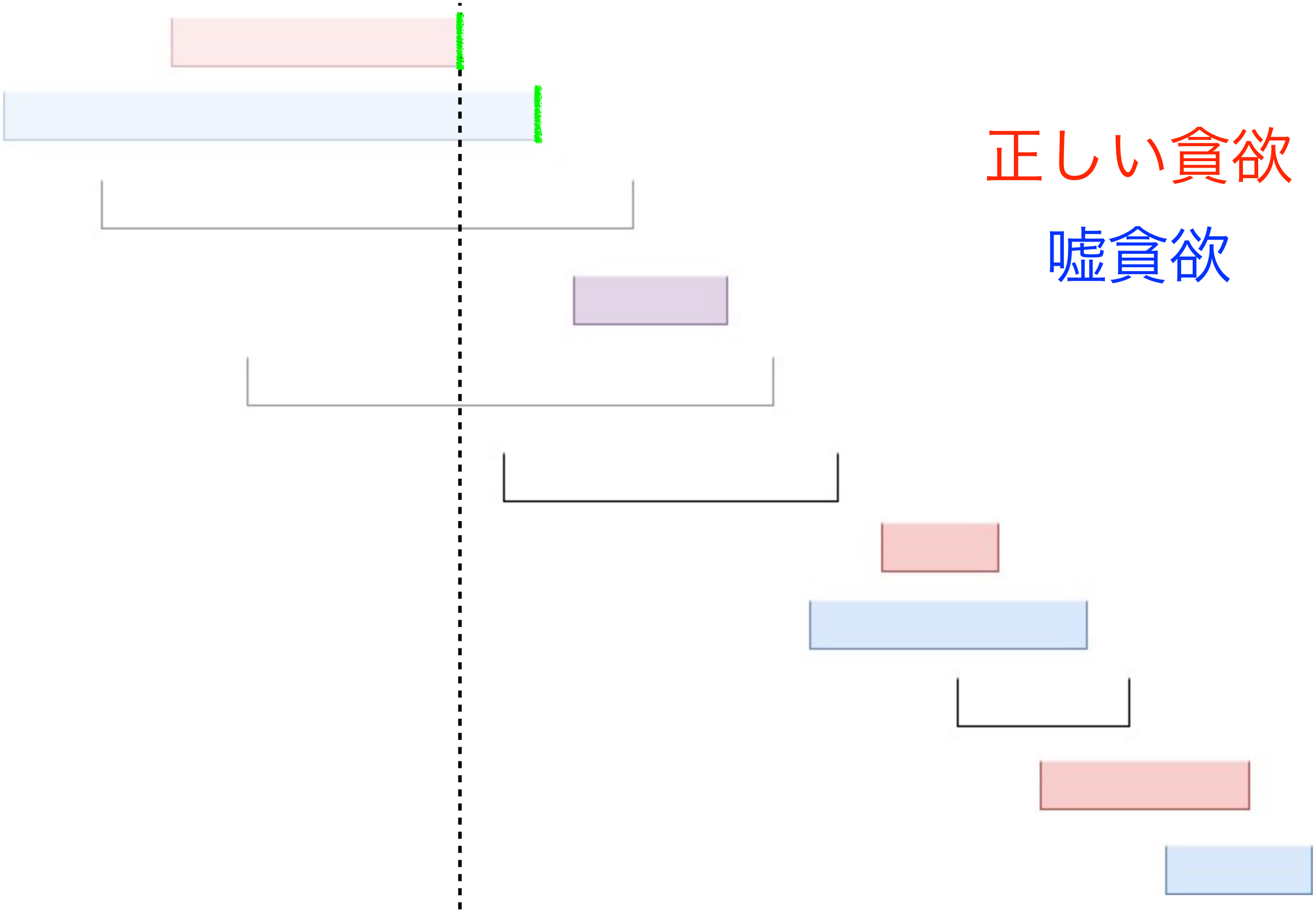

正しい貪欲
嘘貪欲
始点がこのラインより
後にある区間を定めた
終点だけ定めた →

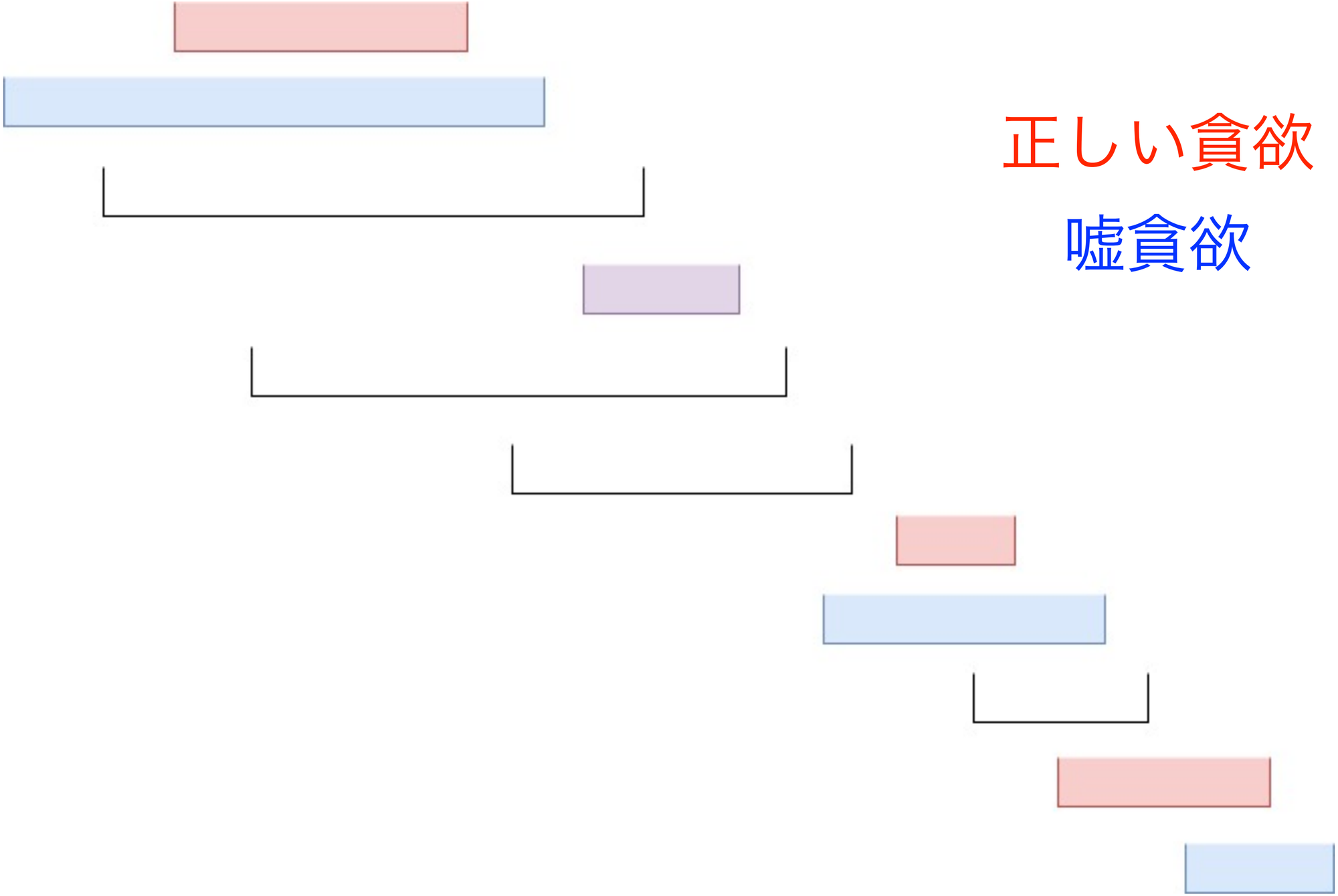


点線を動かすにあたって、

1. 赤・青の区間の位置関係を定める
2. 点線が動いた範囲の中に始点を持つ黒い区間を挿入

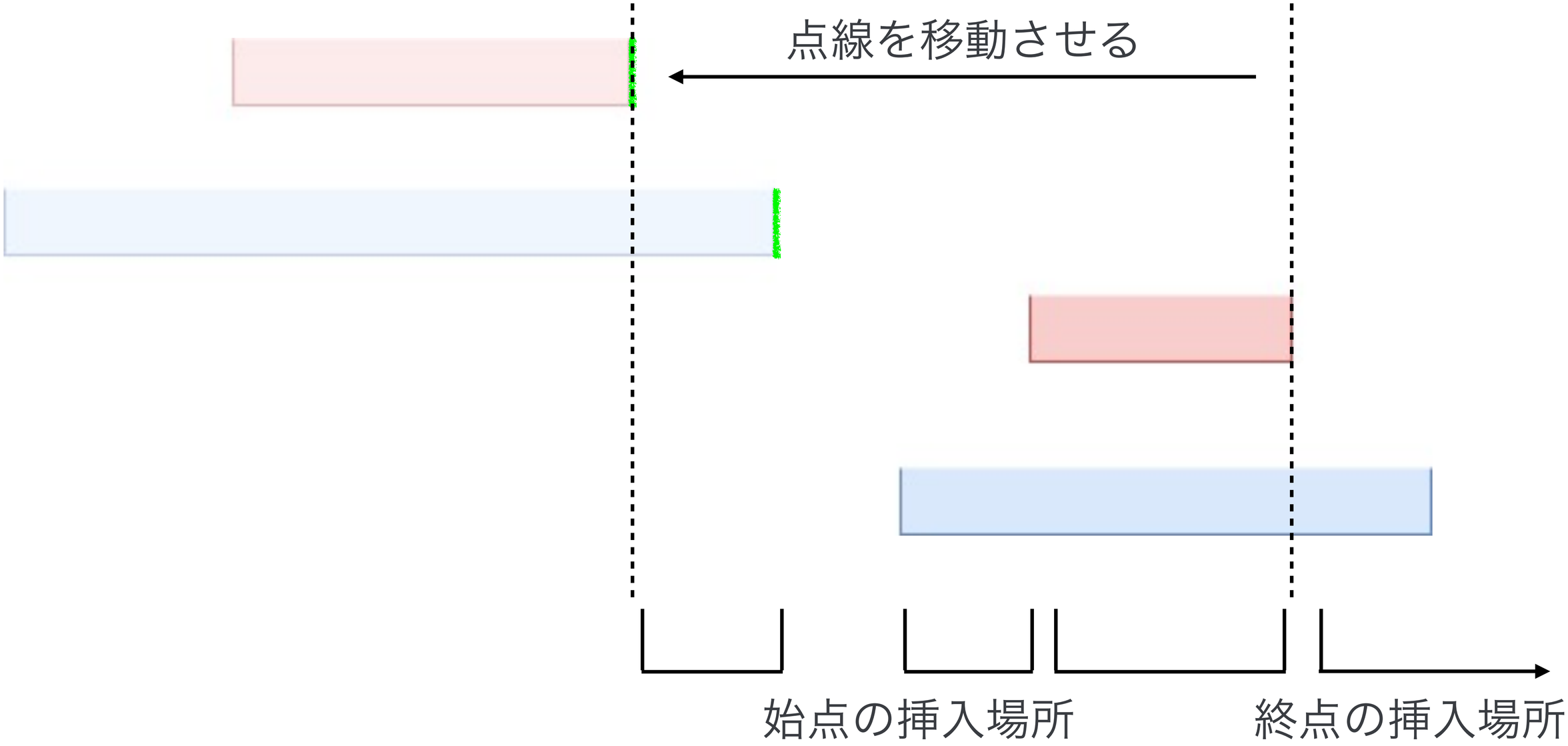

正しい貪欲
嘘貪欲

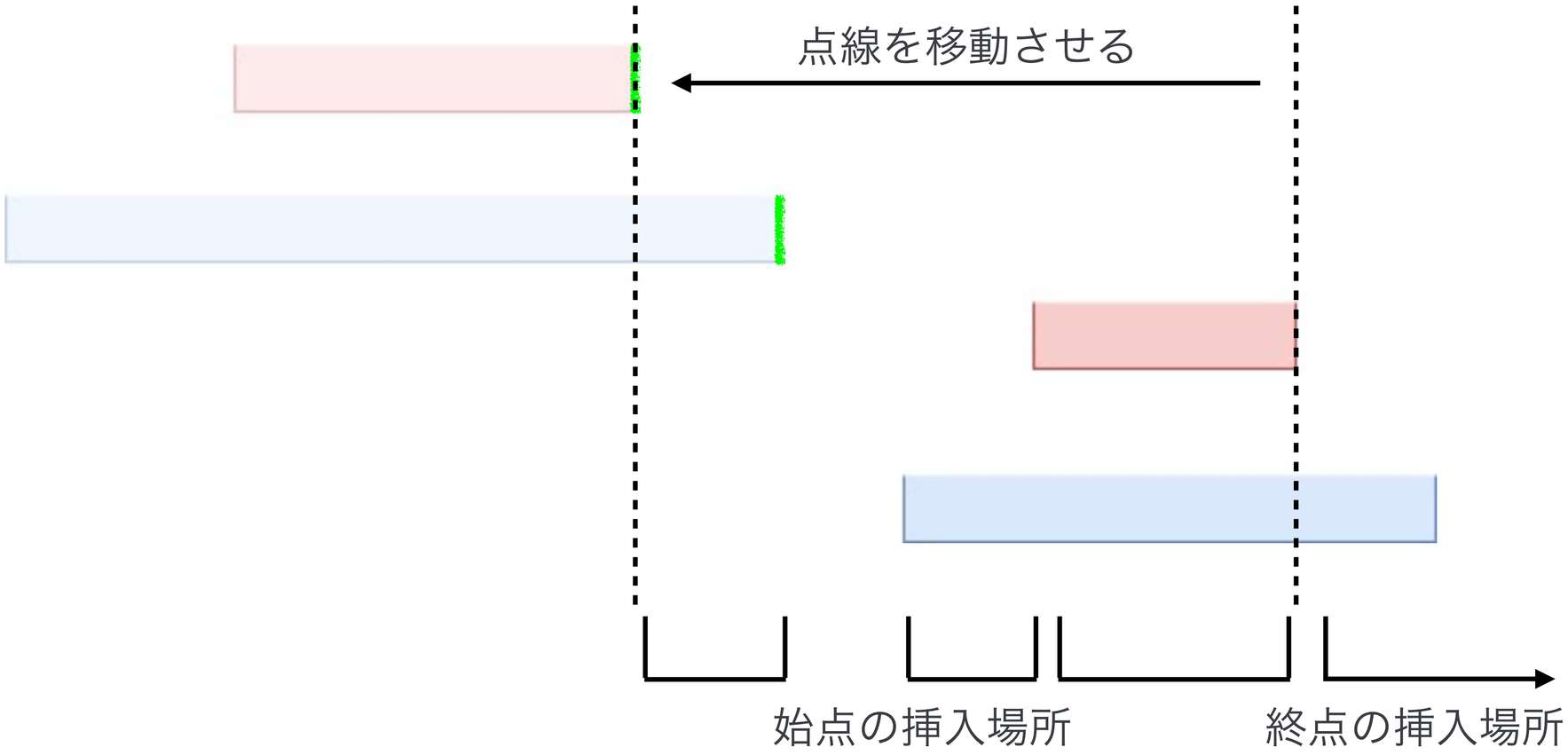

正しい貪欲
嘘貪欲


正しい貪欲
嘘貪欲

小課題 5 ($N \leqq 3000$)

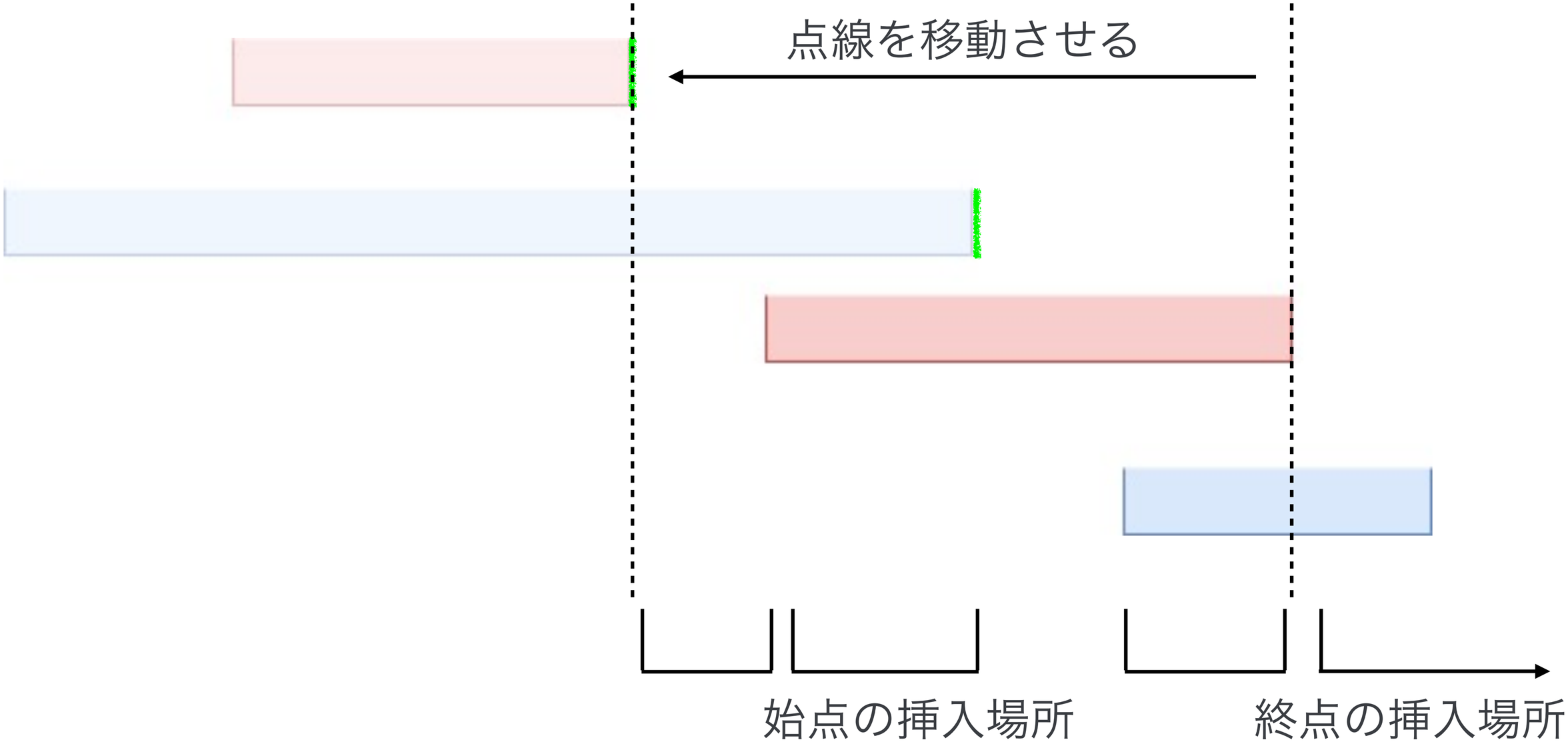
- 持つべき情報は、
  - 区間を何個定めたか
  - 終点だけ定まった赤・青の区間が一致したもの (図中における紫区間) かどうか
- 遷移の式をもうちょっと具体的に説明します


点線を移動させる
始点の挿入場所
終点の挿入場所



元々区間が i 個あって、新たに挿入する黒区間が j 個であるとき、

$$\mathrm{dp}[i+j+2][0] \leftarrow \mathrm{dp}[i+j+2][0] + \mathrm{dp}[i][0] \cdot {}_{j+2}C_2 \cdot (2i-2)(2i-1)\cdots(2i-2+(j-1))$$



※ この位置関係の場合もあることに注意

小課題 5 ($N \leqq 3000$)

- dp[*][0] ← dp[*][0] の形の遷移だけを扱ったが、他の 3 通りについても同様
- 状態 $O(N)$、 遷移 $O(N)$ なので $O(N^2)$ で解けた

ここまでで 87 点

小課題 6 ($N \leqq 20000$)

- そういえば使っていない制約があった
  - P は大きい素数
- 小課題 6 で初めて使います (人によってはもっと早く使うかもしれない)

小課題 6 ($N \leqq 20000$)

- さっきの遷移の式をもう一度よく眺めてみよう

$$\mathrm{dp}[i+j+2][0] \leftarrow \mathrm{dp}[i+j+2][0] + \mathrm{dp}[i][0] \cdot {}_{j+2}C_2 \cdot (2i-2)(2i-1)\cdots(2i-2+(j-1))$$

- $i+j+2$ を改めて j と置き直す

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot {}_{j-i}C_2 \cdot (2i-2)(2i-1)\cdots(i+j-5)$$

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot \frac{(j-i)(j-i-1)}{2} \cdot (2i-2)(2i-1)\cdots(i+j-5)$$

小課題 6 ($N \leqq 20000$)

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot \frac{(j-i)(j-i-1)}{2} \cdot (2i-2)(2i-1)\cdots(i+j-5)$$

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot \frac{(j-i)(j-i-1)}{2} \cdot \frac{(i+j-5)!}{(2i-3)!}$$

- ある配列 P, Q、定数 A, B, C が存在して、

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot P[i] \cdot Ai^2 \cdot Q[i+j]$$

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot P[i] \cdot Bij \cdot Q[i+j]$$

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot P[i] \cdot Cj^2 \cdot Q[i+j]$$

小課題 6 ($N \leqq 20000$)

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot P[i] \cdot Ai^2 \cdot Q[i+j]$$

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot P[i] \cdot Bij \cdot Q[i+j]$$

$$\mathrm{dp}[j][0] \leftarrow \mathrm{dp}[j][0] + \mathrm{dp}[i][0] \cdot P[i] \cdot Cj^2 \cdot Q[i+j]$$

$i, j, i+j$ に関する項しか出てこない

→ 畳み込める形になった🎉

畳み込みとは？

- 長さ N の配列 A, B が与えられる
- 以下の式で定義される配列 C を求めたい

$$C[i] = \sum_{j=0}^{i} A[j] \cdot B[i-j]$$

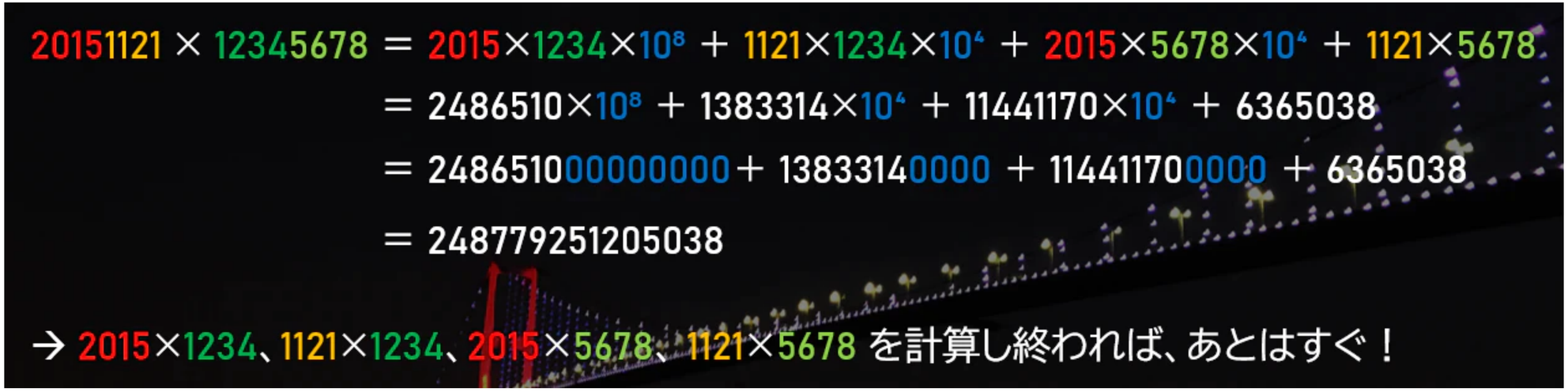
- ただし、出てくる値は全て $\mathrm{mod}\ P$ で考えるものとする
- 愚直にやると $O(N^2)$ だが、もっと速く求めたい

手法 1 : NTT (数論変換)

- 畳み込みを $O(N \log N)$ で求められる
- $\bmod P$ の世界でのフーリエ変換のようなもの
- 知らない人で中身が気になる人は自分で調べてください
- 今回の問題では P が任意なので、NTT に適した素数を 3 つほど持ってきてそれぞれにおいて NTT した上で、中国剰余定理から復元
- 任意 mod NTT を時間内に空書きするのは非現実的

手法 2 : Karatsuba 法

- 畳み込みを $O(N^{\log_2 3}) \approx O(N^{1.59})$ で求められる
- 以下、square1001 さんの記事 『超高速！多倍長整数の計算手法【前編：大きな数の四則計算を圧倒的な速度で！】』(Qiita) から引用します



手法 2 : Karatsuba 法

- NTT より遅い代わりに、実装がかなり楽
- 競技時間中に空書きするのも非現実的ではない

小課題 6 ($N \leqq 20000$)

- この問題における畳み込みはオンライン畳み込みとよばれるものなので、ただ畳み込めばいいというものでもない
- 分割統治と組み合わせる必要がある（オンライン・オフライン変換）
- 任意 mod NTT で $O(N(\log N)^2)$

  Karatsuba 法で $O(N^{1.59})$

ここまでで 100 点

得点分布

100　87　51　37　10　5　0