勇者ビ太郎３　解説

2025/3/23

米田優峻 (E869120)

問題概要

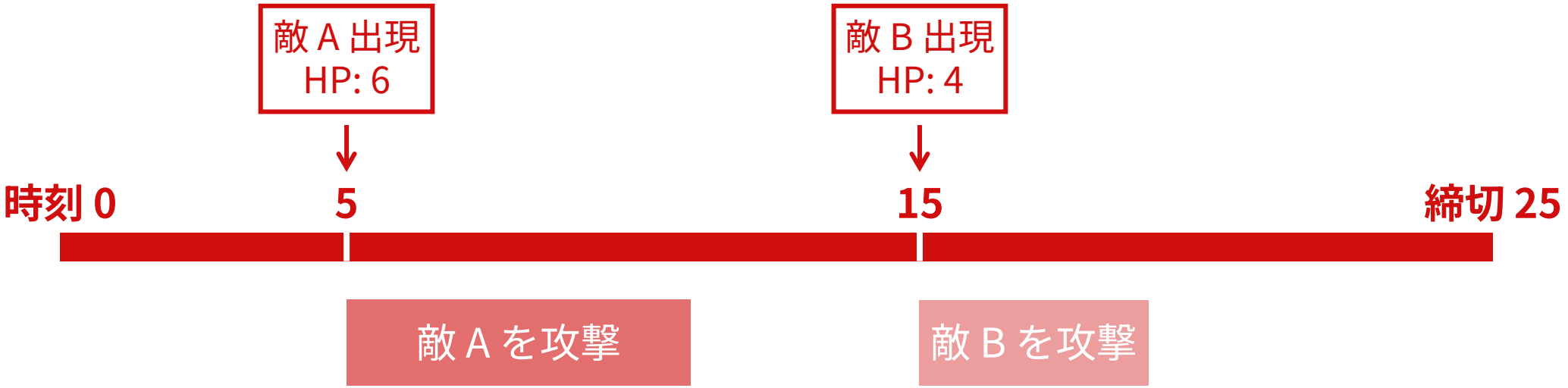
まず、次の問題を考えてみよう

問題概要

1 秒につきどれかの敵の HP を 1 減らすことができる

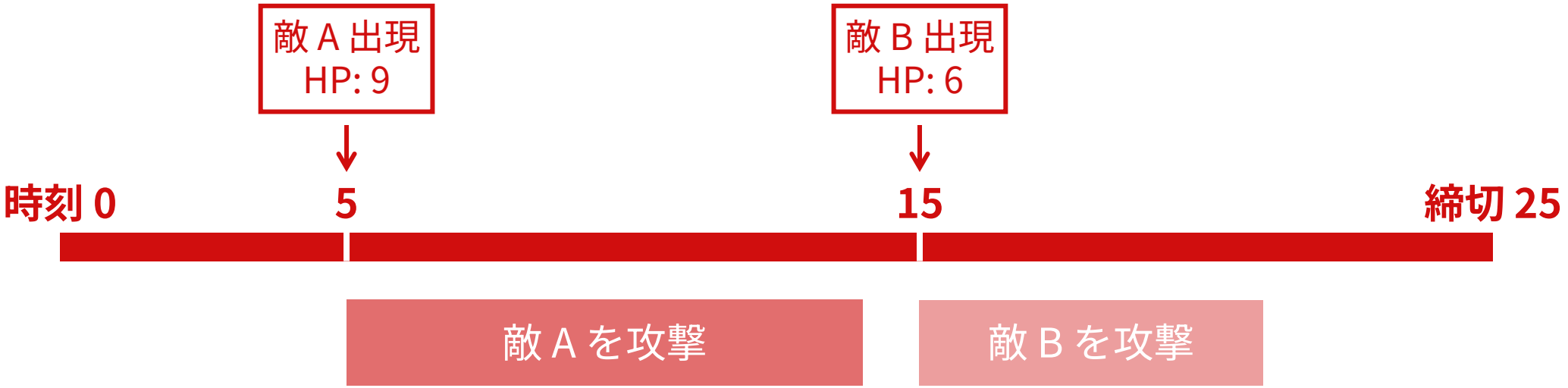
難易度いくつまでクリア (全部の敵を倒すこと) できるか？

問題概要



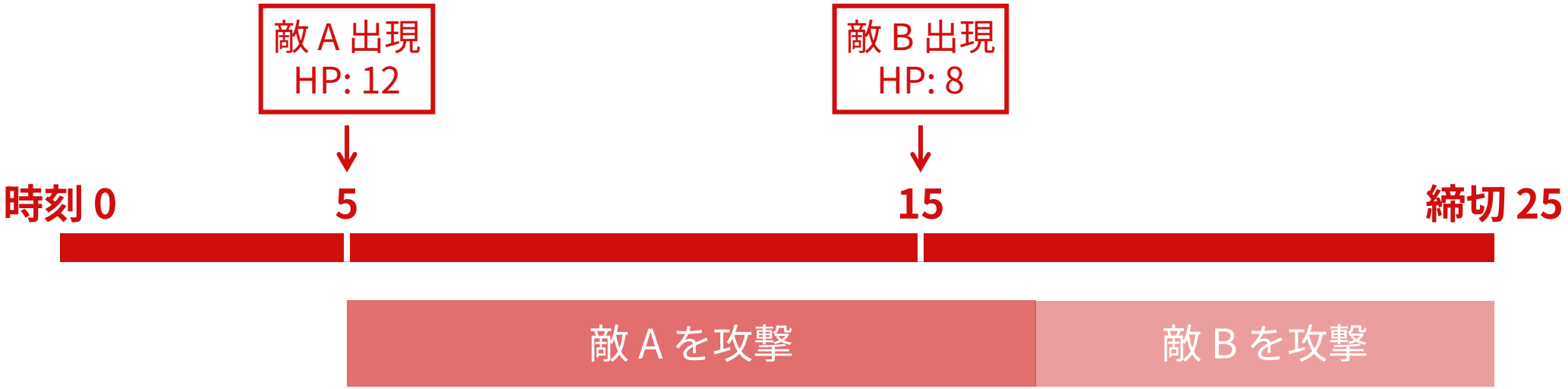
難易度 2 であれば、上記の戦略でクリア可能

問題概要



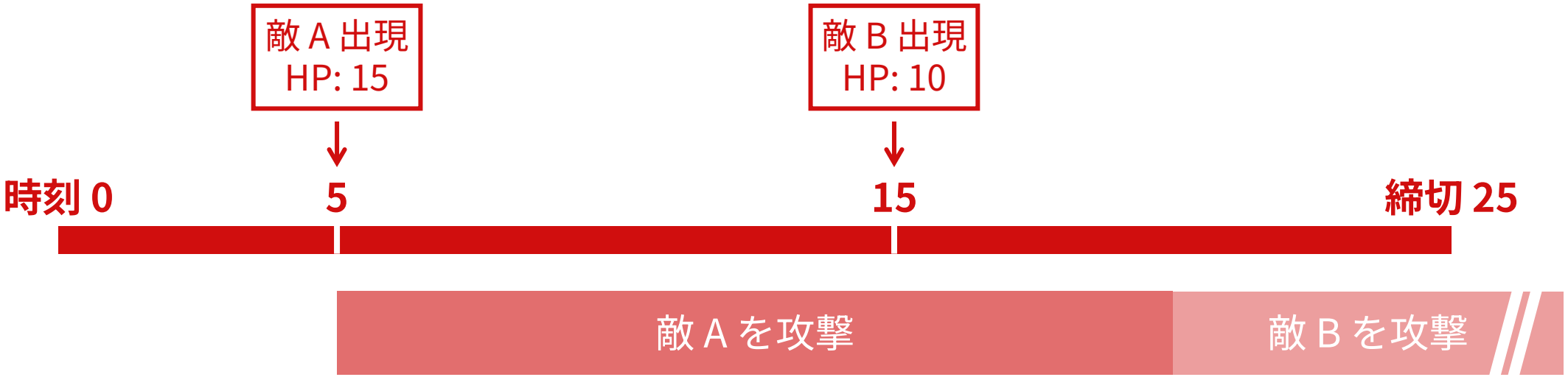
難易度 3 であれば、上記の戦略でクリア可能

問題概要



難易度 4 であれば、上記の戦略でクリア可能

問題概要



しかし、難易度 5 はクリアすることができない

よって答えは 4

問題概要

しかし、実際の問題では

敵を全部倒す必要はない

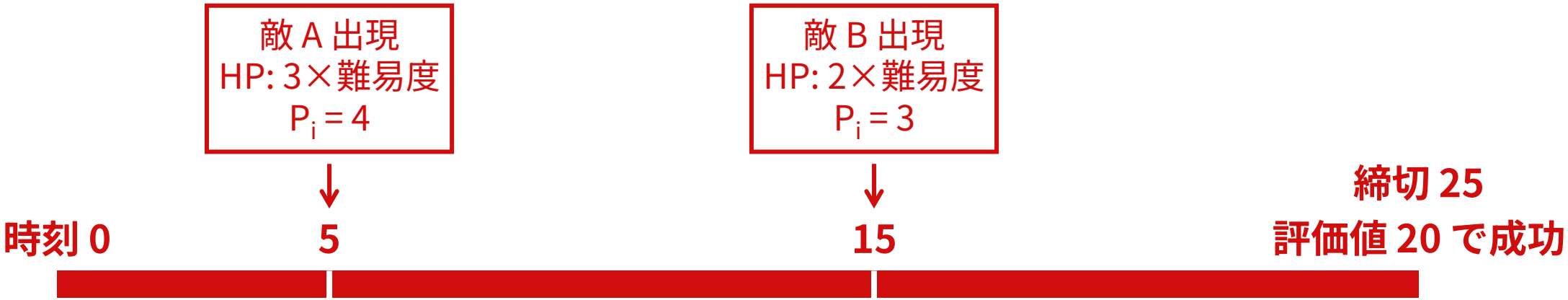
問題概要

各敵について強さ P_i が付けられていて

最後の HP× P_i の合計 (評価値という) が

一定の基準値以下であれば成功

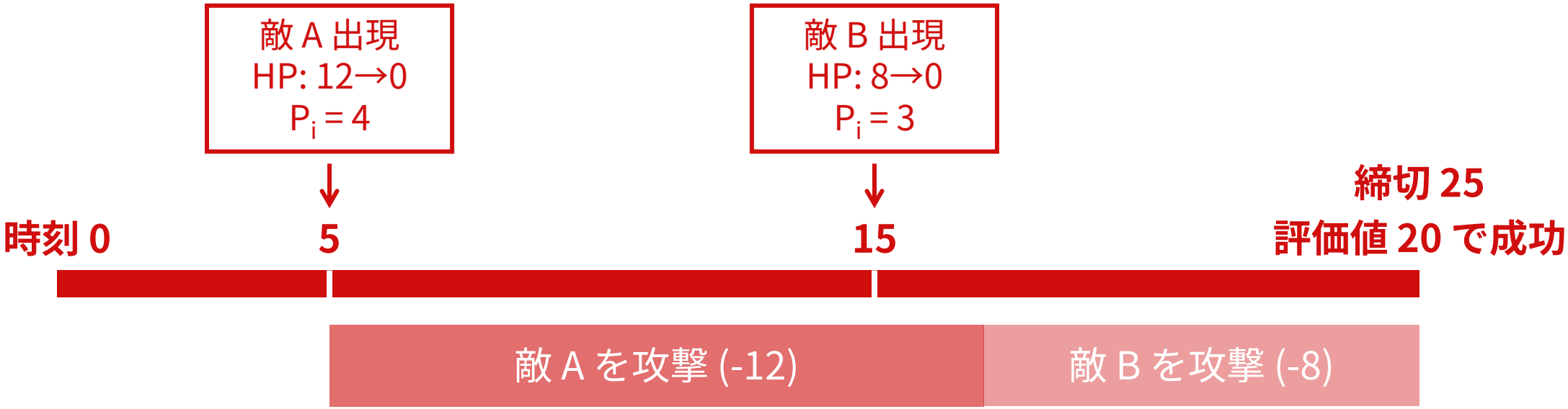
問題概要



たとえば上のようなケースの場合 (評価値 20 で成功)

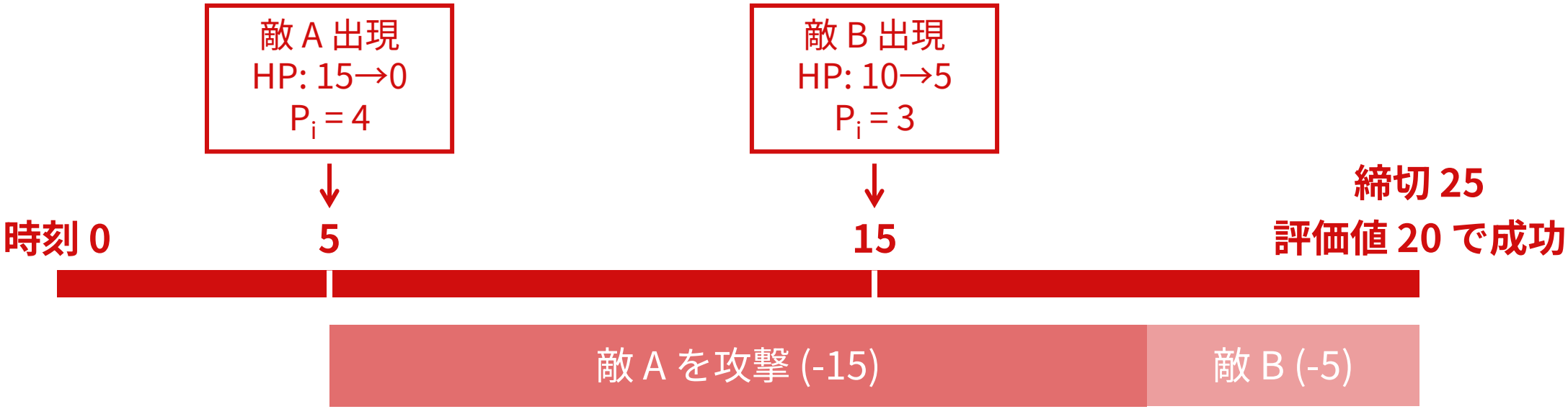
難易度いくつまでですることがクリアできるか？

問題概要



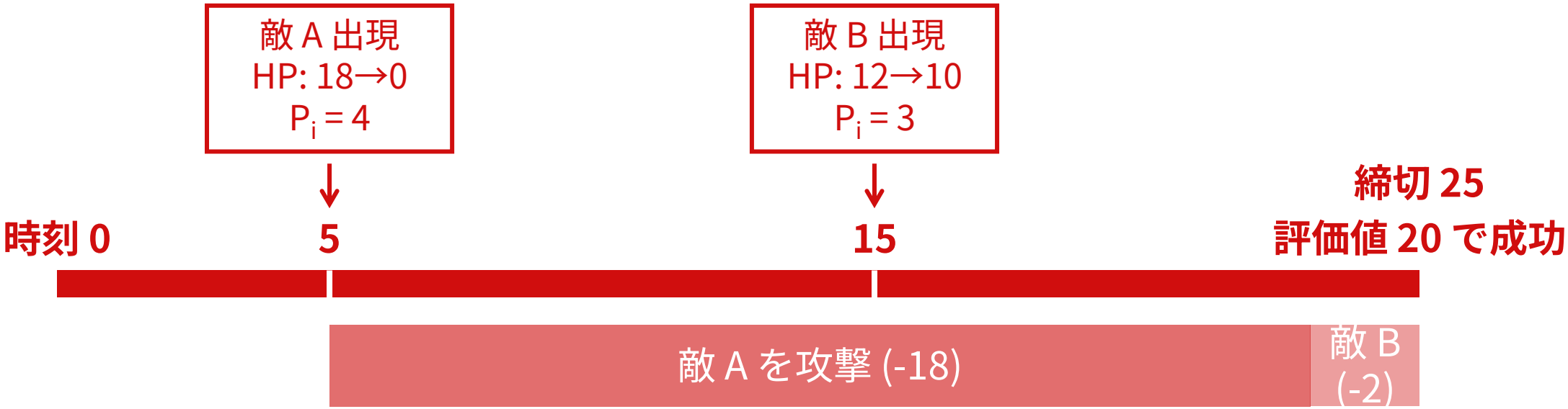
まず、難易度 4 はクリア可能 (評価値 0×4 + 0×3 = 0)

問題概要



また、難易度 5 もクリア可能 (評価値 $0 \times 4 + 5 \times 3 = 15$)

問題概要



しかし、難易度 6 は不可能 (評価値 $0 \times 4 + 10 \times 3 = 30$) → 答えは 6

問題概要

今回は基準値が「評価値 20」でしたが

このような問題を Q $(\leq 10^6)$ 個の基準値について解けますか？

というのが今回の問題です

制約

- モンスターの数 $N \leq 6000$
- クエリ (基準値) の数 $Q \leq 1000000$
- 締切時刻 $T \leq 10^{18}$

小課題 1

小課題 1

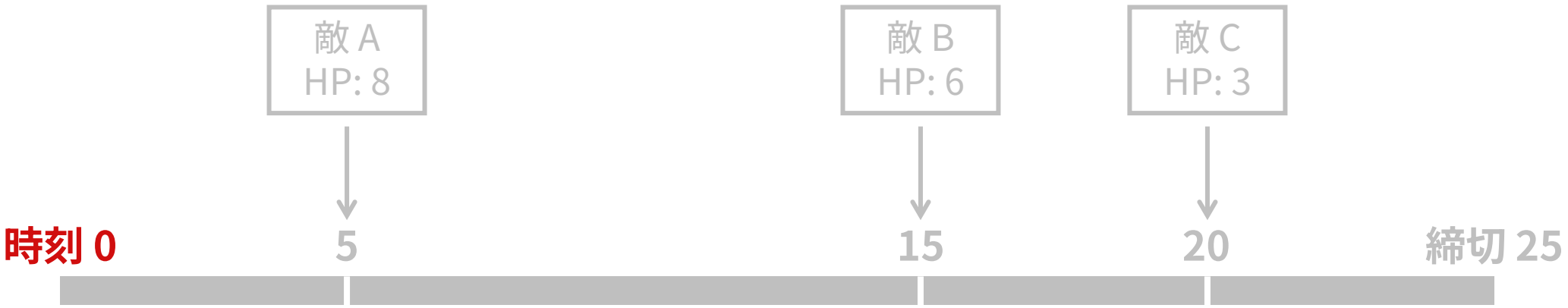
- $Q = 1, M_1 = 1, L = 0$
- つまり、難易度 1 のダンジョンで全部の敵を倒せるかを判定できれば良い

小課題 1: 解法

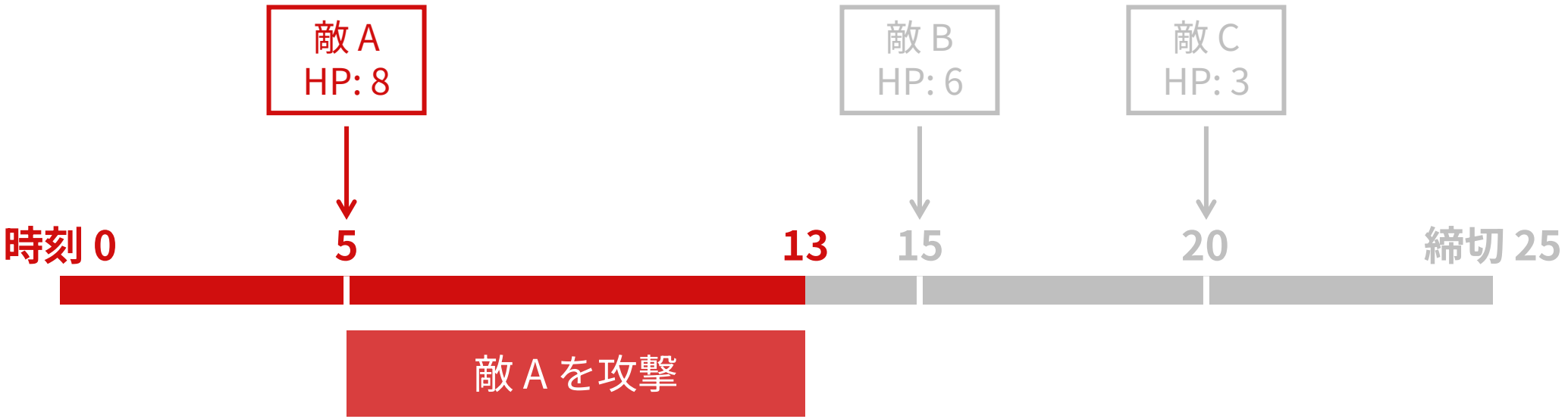
- 難易度が決まっているので、単にモンスターの出現時刻 S_i が小さい順に殺せばよい
- モンスターを S_i の小さい順にソートすると解ける

1 点

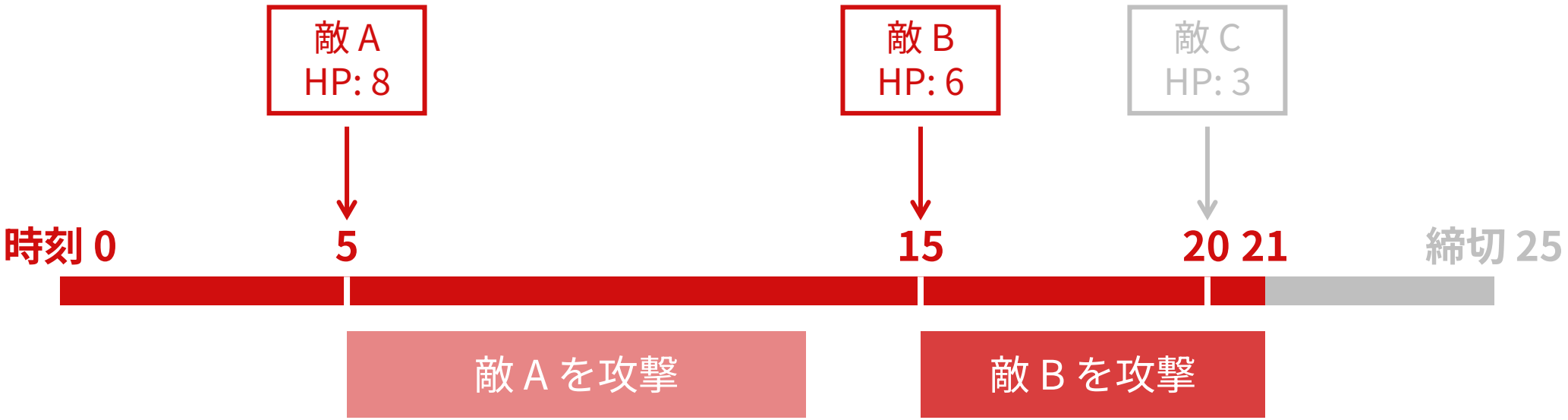
小課題 1: 例 1



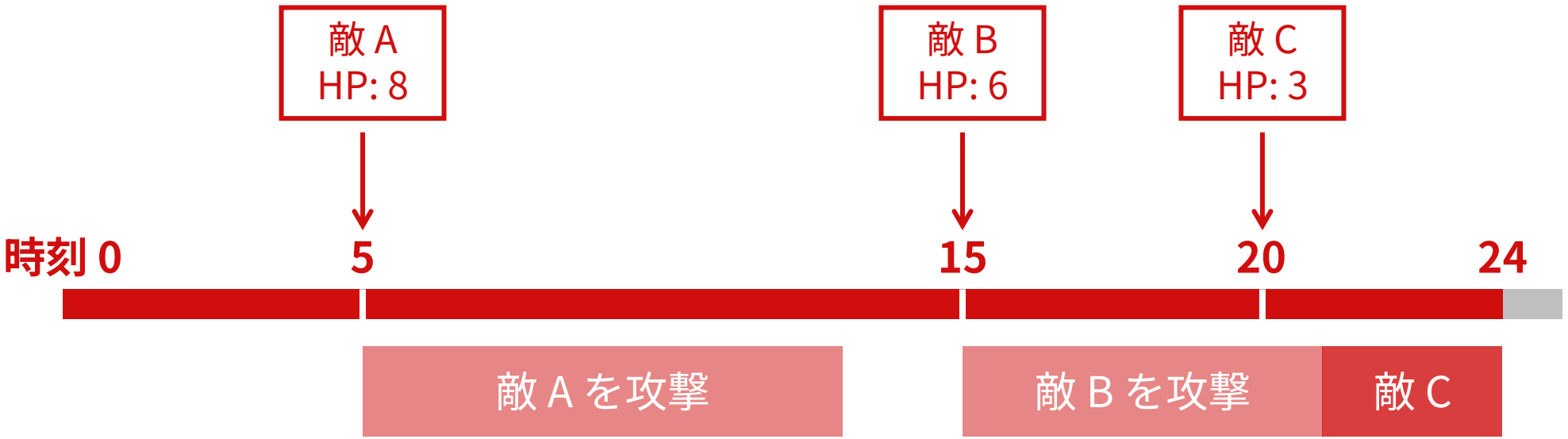
小課題 1: 例 1



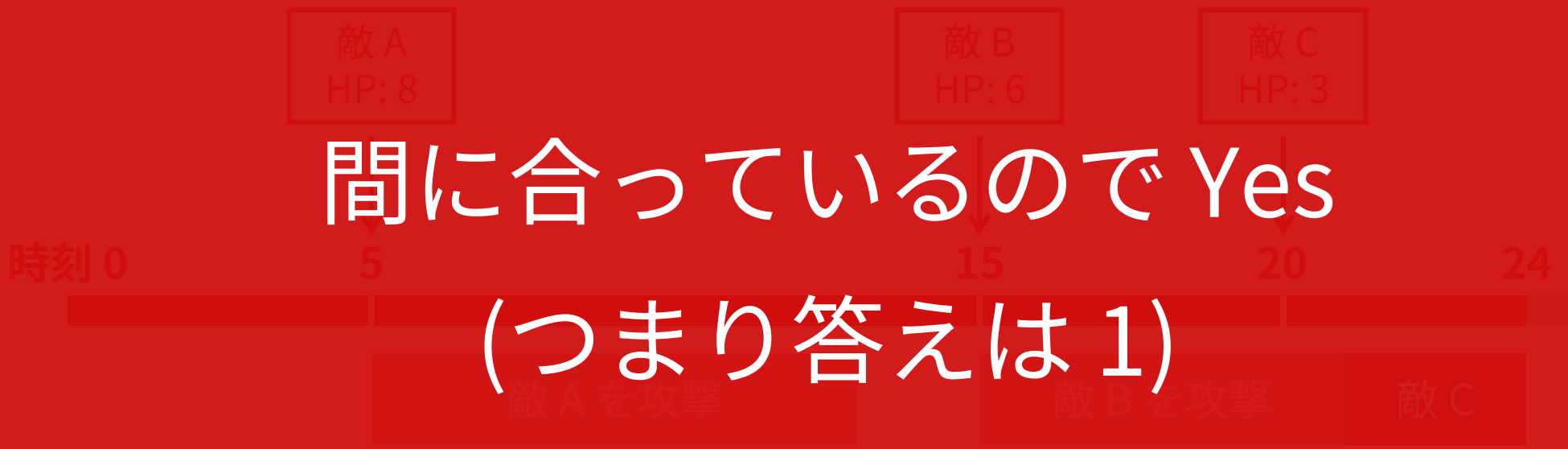
小課題 1: 例 1



小課題 1: 例 1



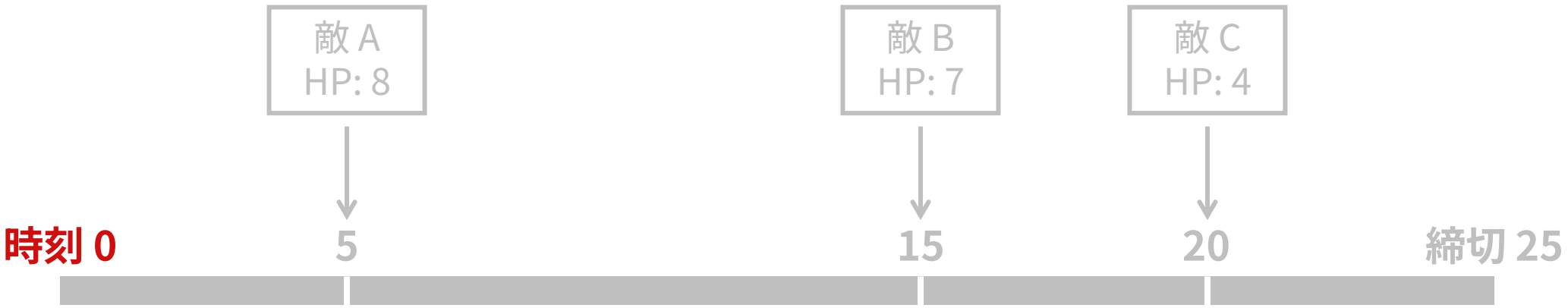
小課題 1: 例 1



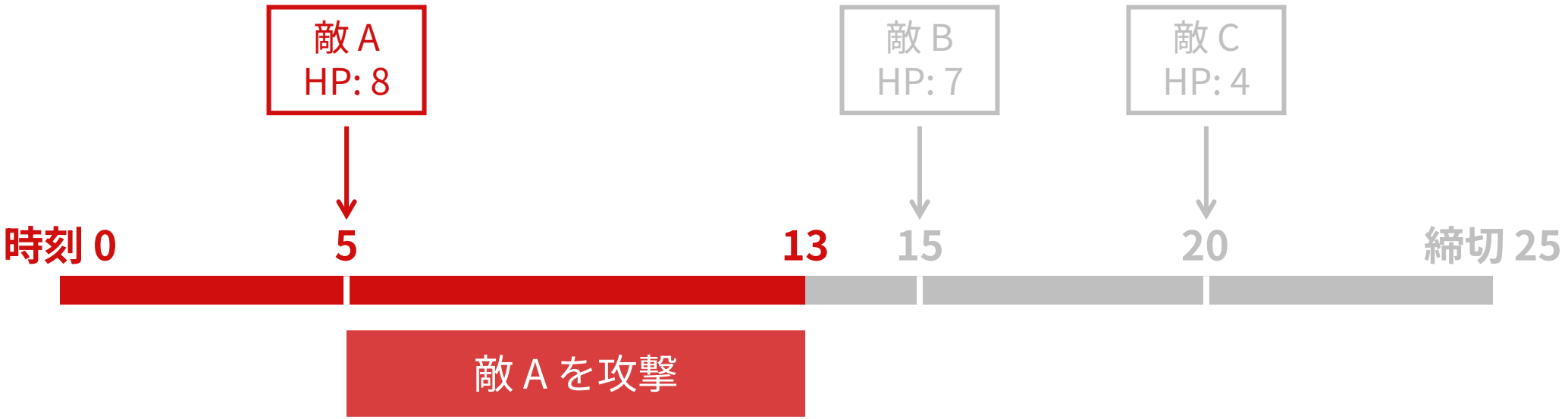
間に合っているので Yes

(つまり答えは 1)

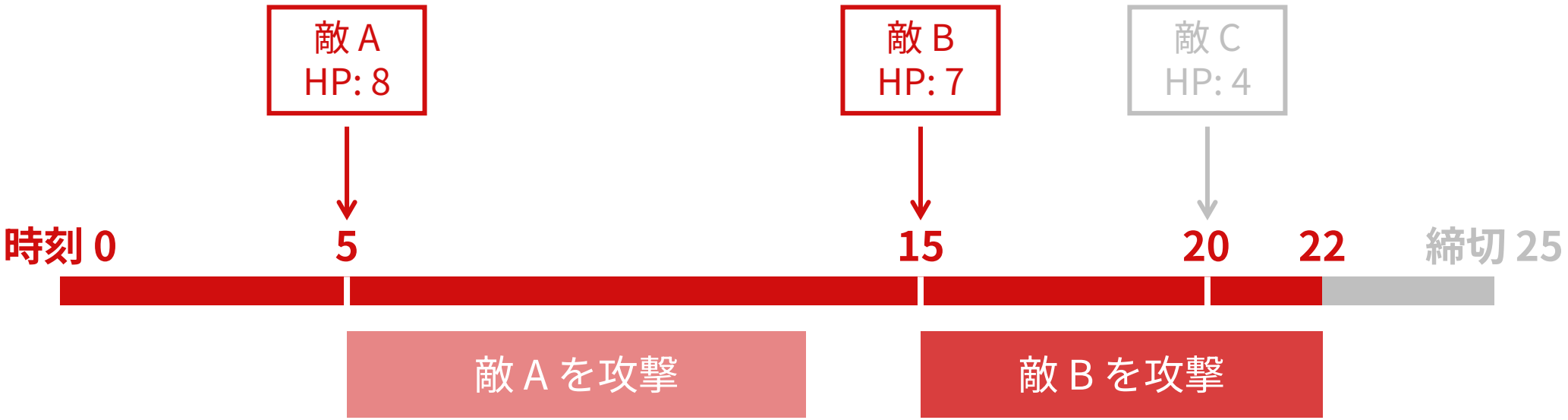
小課題 1: 例 2



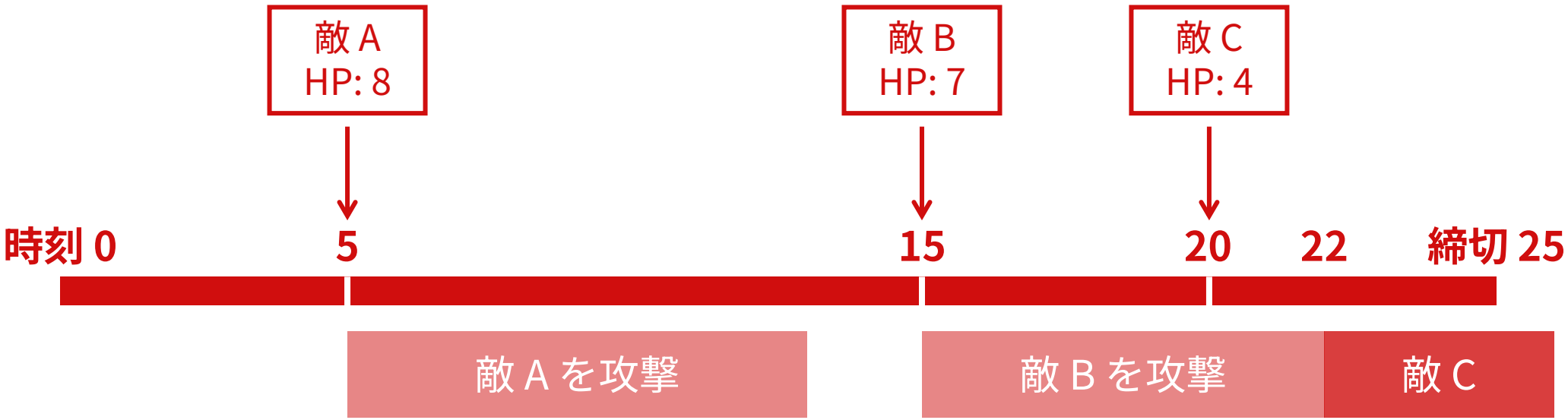
小課題 1: 例 2



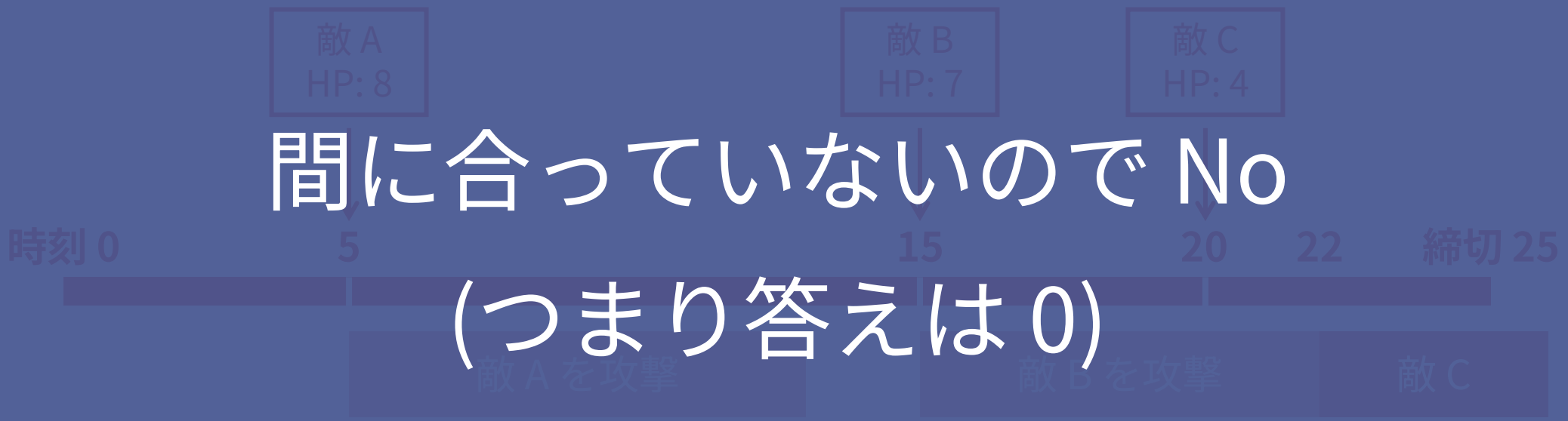
小課題 1: 例 2



小課題 1: 例 2



小課題 1: 例 2



間に合っていないので No

(つまり答えは 0)

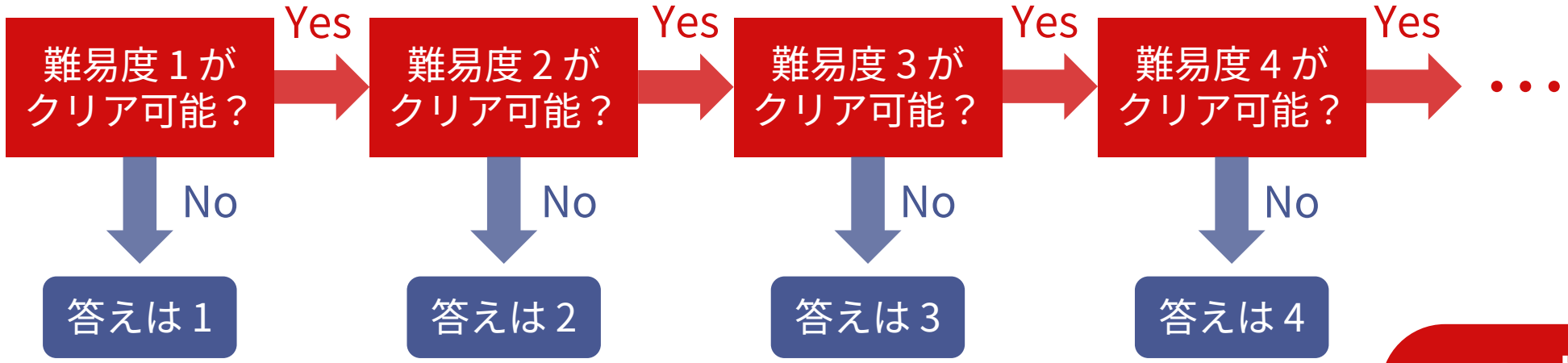
小課題 2

小課題 2

- $N \leq 30, Q = 1, M_1 = 0$
- つまり全敵の HP をゼロにできる最大の難易度を求めればよい
- 難易度 $L \leq 10^7$

解法 1: 全探索

以下のように全探索をすると、計算量が $O(NL)$ になる。制約が $N \leq 30, L = 10^7$ なので通る

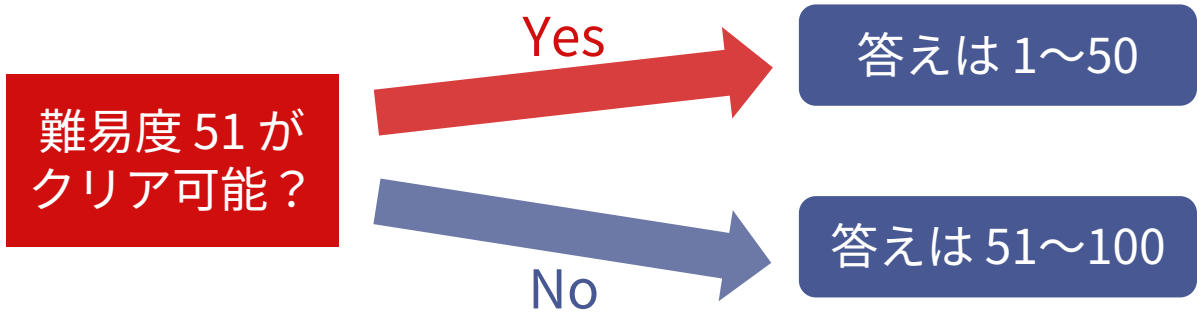


4 点

解法 2: 二分探索

しかし、二分探索をすると $O(N \log L)$ に改善される

二分探索のアイデアとしては、たとえば $L = 100$ として、最初に難易度 51 がクリア可能かどうかを聞く (それで選択肢が半分になるので、それを繰り返す)



4 点

小課題 3

小課題 3

- $N \leq 30, Q \leq 3$
- これまであった $M_i = 0$ の制約が消えているので、全部の敵を必ずしも倒す必要はない

小課題 3

ある難易度 l をクリアするための
最適な戦略を考えてみよう

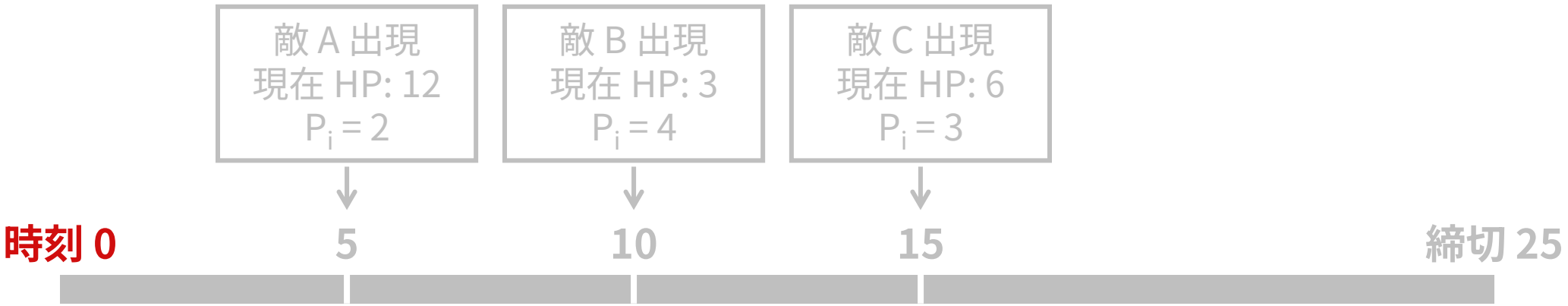
小課題 3

最小化したいのは (最終 HP) × P_i の合計

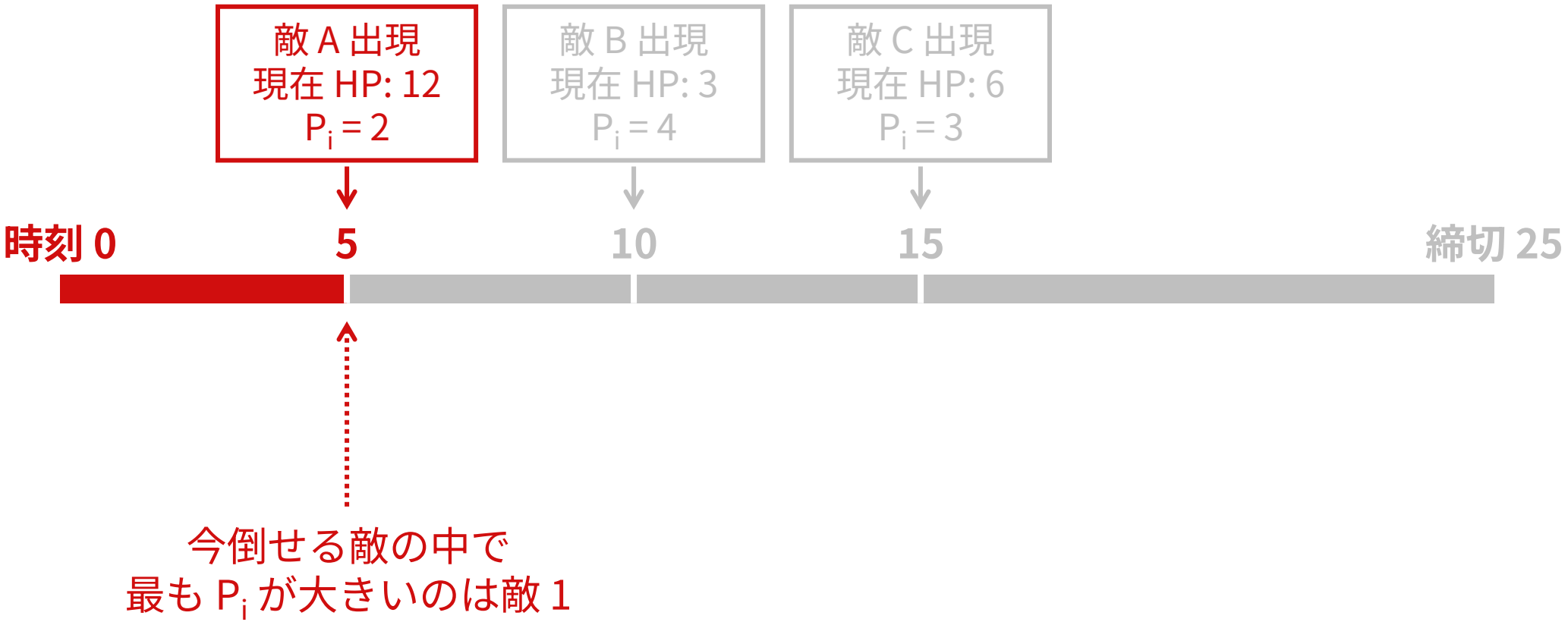
▼

今攻撃できる敵のうち最も P_i が大きいものを
攻撃すればよいのではないか？

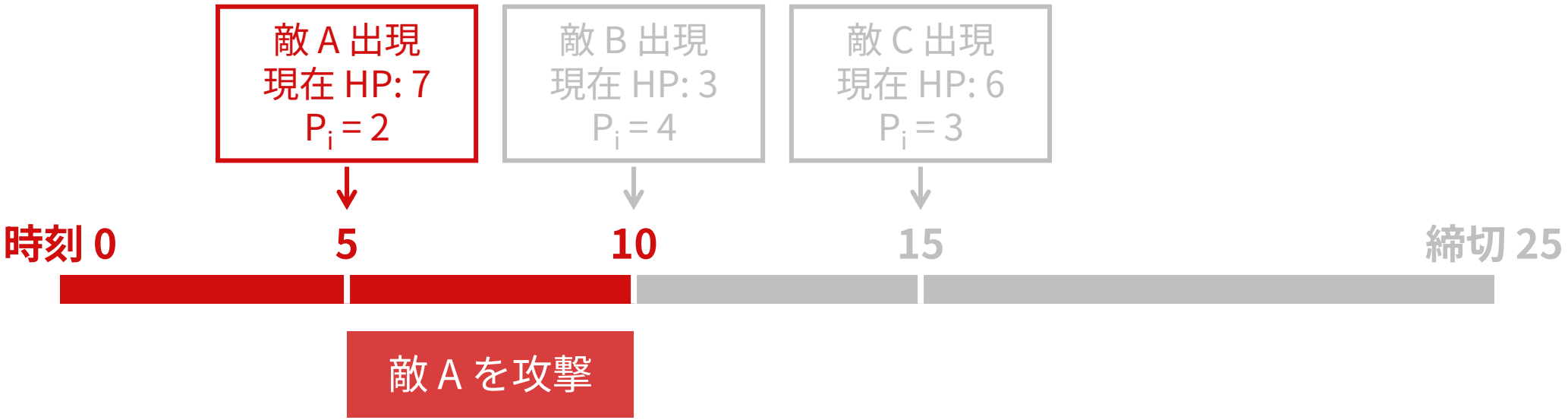
最適戦略: 例



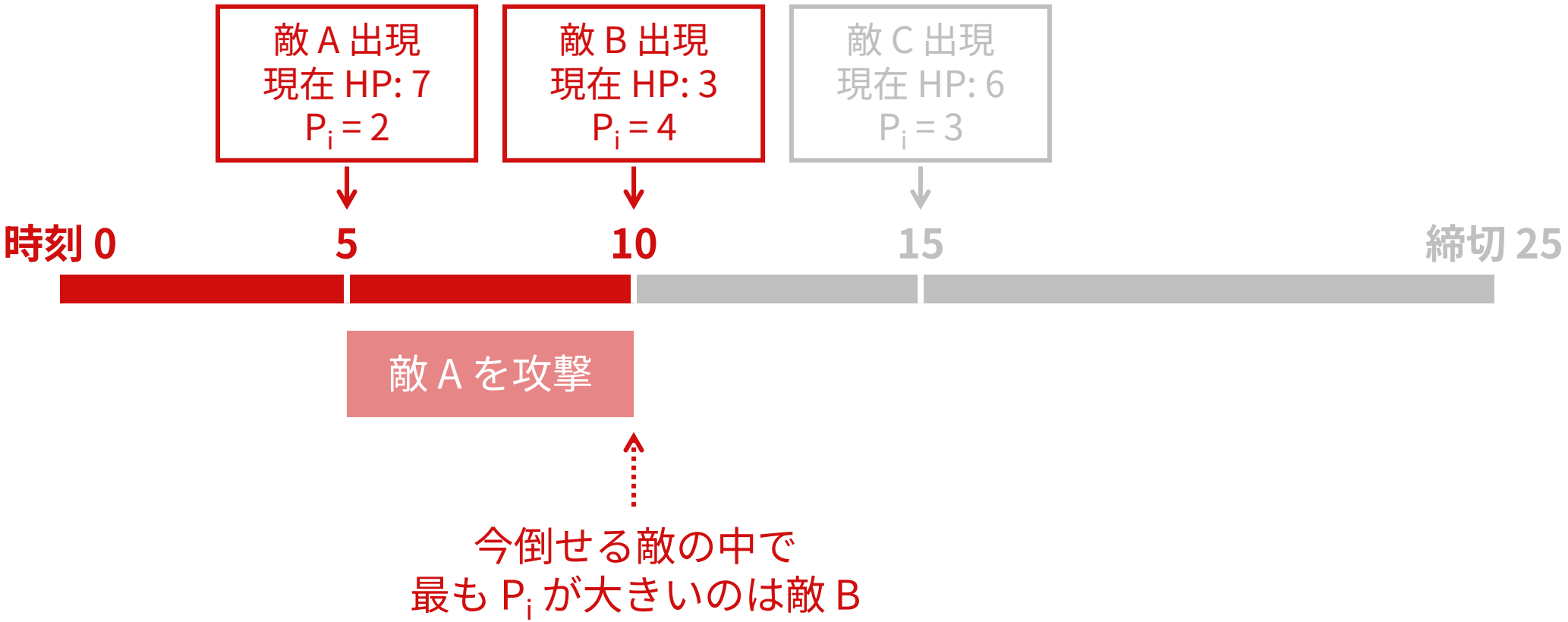
最適戦略: 例


敵 A 出現
現在 HP: 12
$P_i = 2$
敵 B 出現
現在 HP: 3
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
15
締切 25
今倒せる敵の中で
最も P_i が大きいのは敵 1

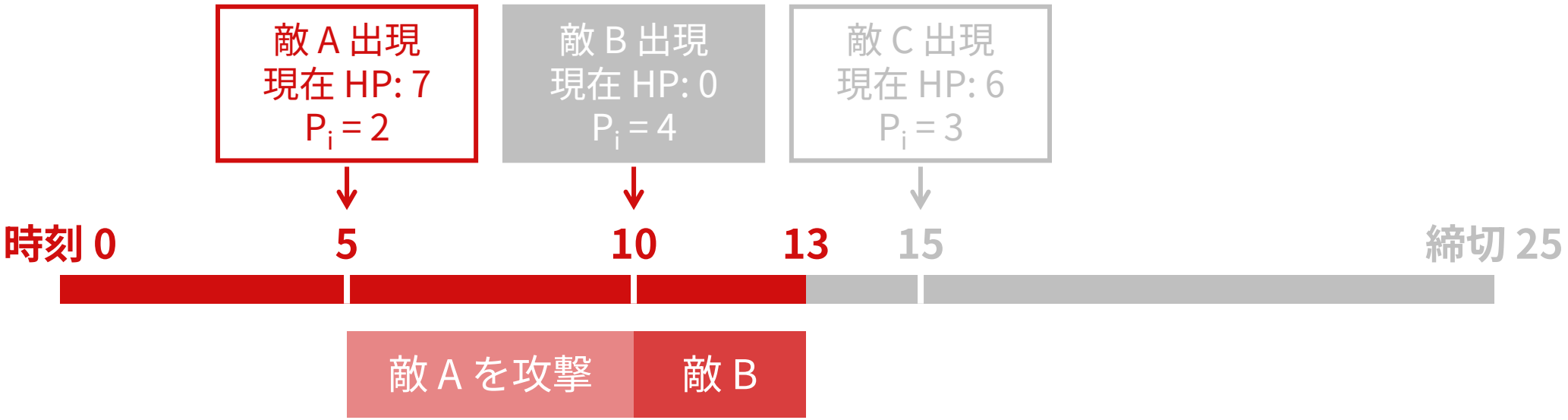
最適戦略: 例


敵 A 出現
現在 HP: 7
$P_i = 2$
敵 B 出現
現在 HP: 3
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
15
締切 25
敵 A を攻撃

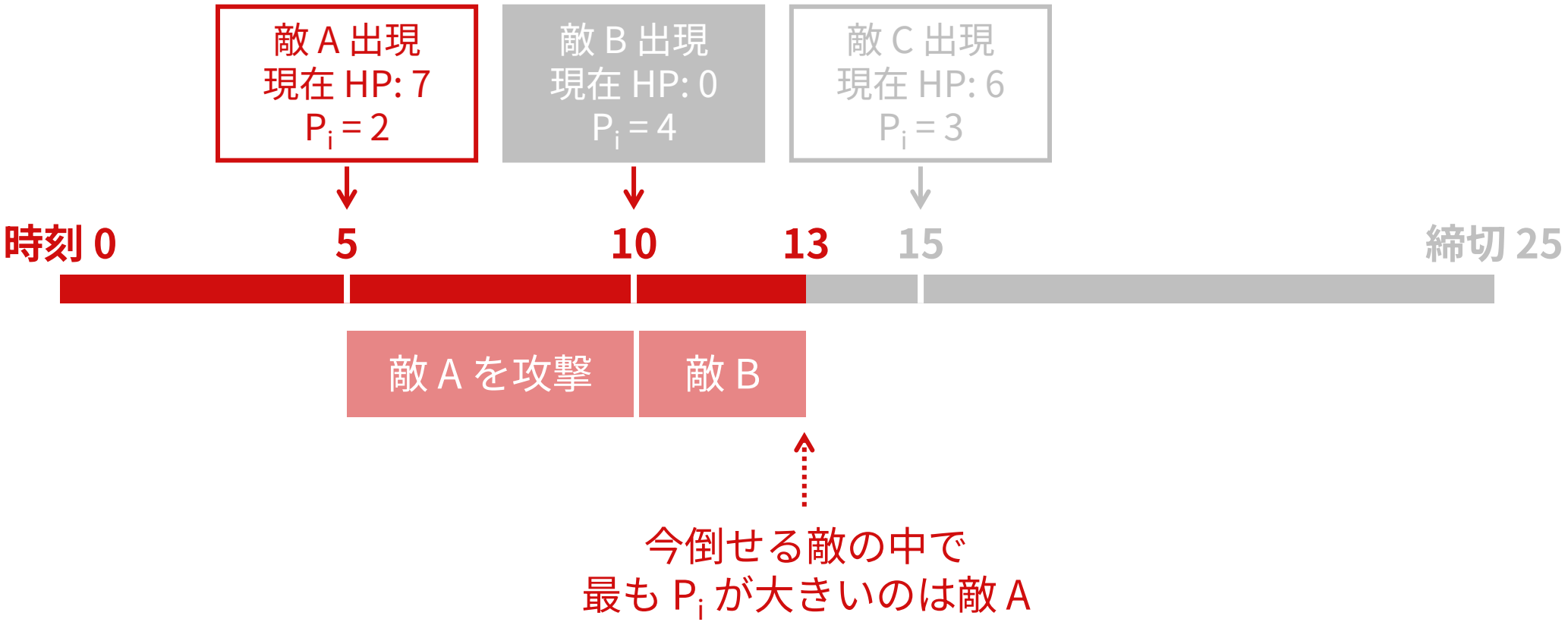
最適戦略: 例


敵 A 出現
現在 HP: 7
$P_i = 2$
敵 B 出現
現在 HP: 3
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
15
締切 25
敵 A を攻撃
今倒せる敵の中で
最も P_i が大きいのは敵 B

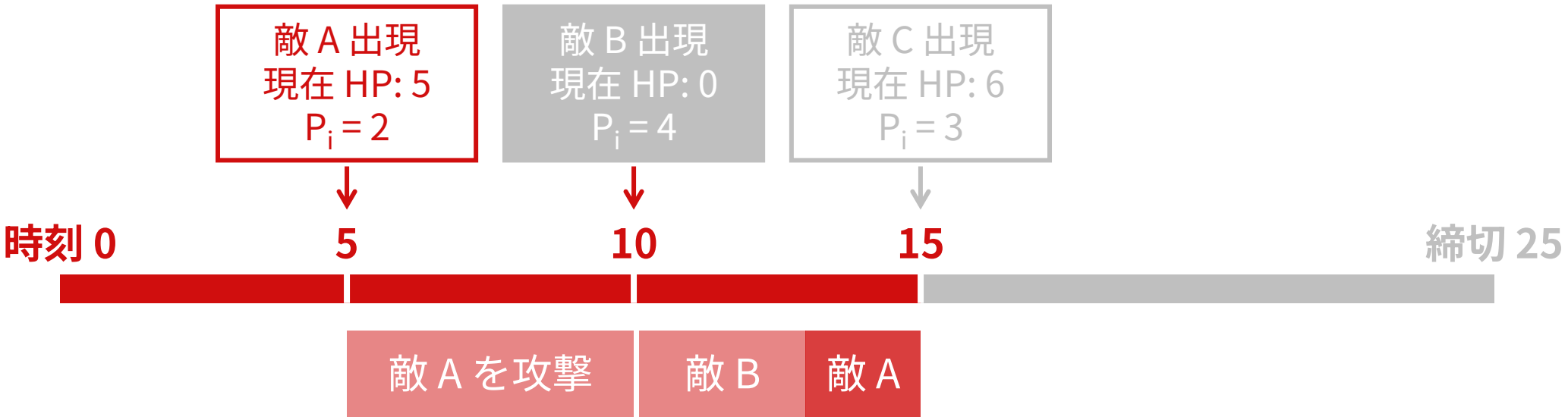
最適戦略: 例


敵 A 出現
現在 HP: 7
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
13
15
締切 25
敵 A を攻撃
敵 B

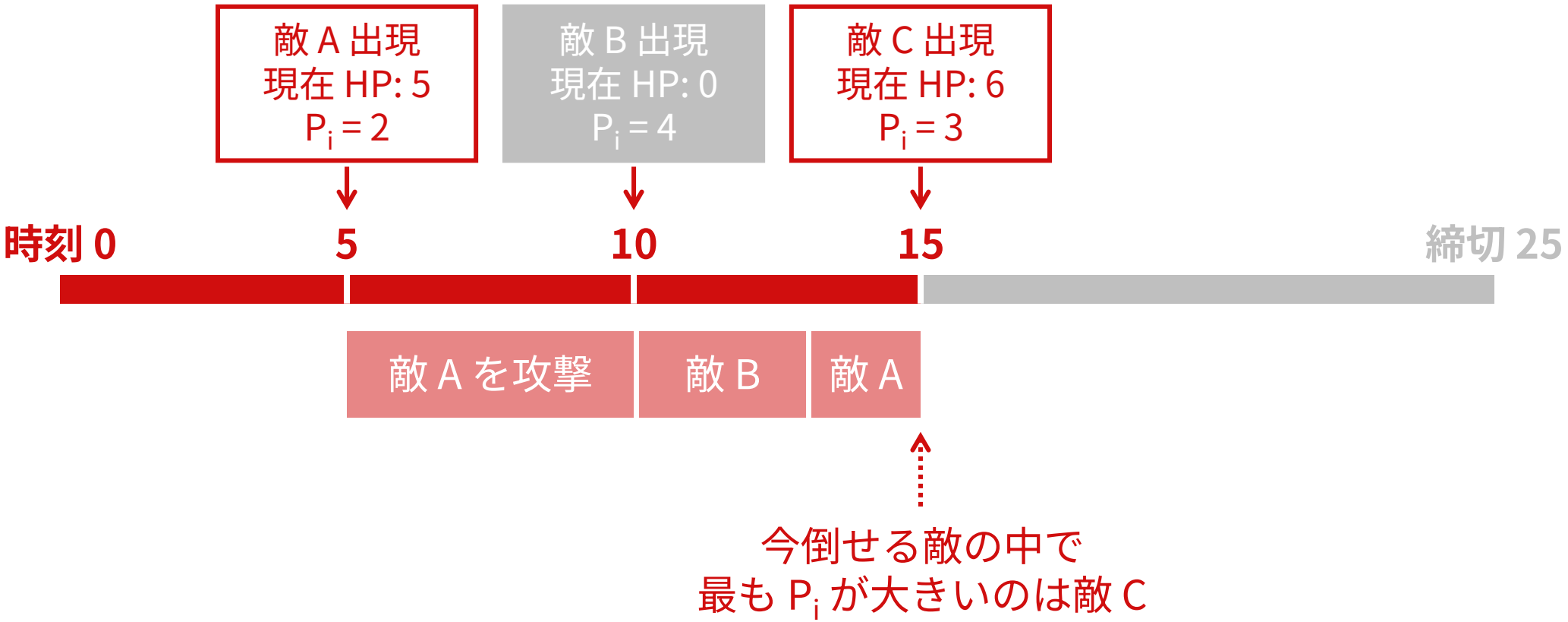
最適戦略: 例


敵 A 出現
現在 HP: 7
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
13
15
締切 25
敵 A を攻撃
敵 B
今倒せる敵の中で
最も P_i が大きいのは敵 A

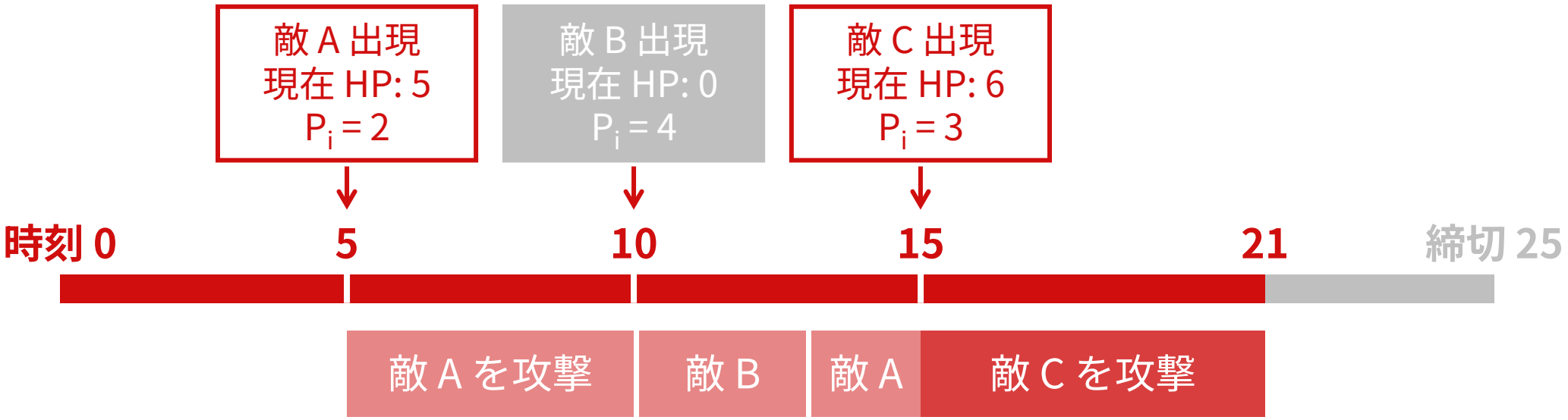
最適戦略: 例


敵 A 出現
現在 HP: 5
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
15
締切 25
敵 A を攻撃
敵 B
敵 A

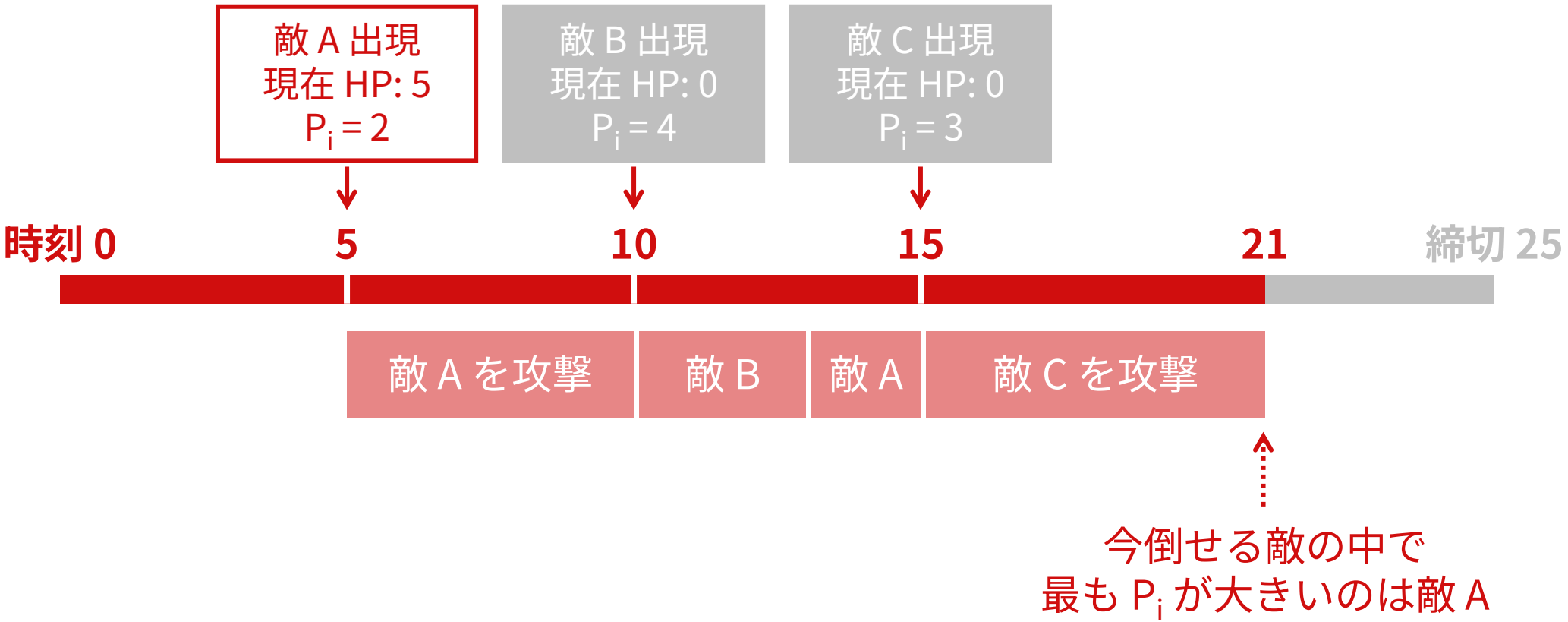
最適戦略: 例


敵 A 出現
現在 HP: 5
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
15
締切 25
敵 A を攻撃
敵 B
敵 A
今倒せる敵の中で
最も P_i が大きいのは敵 C

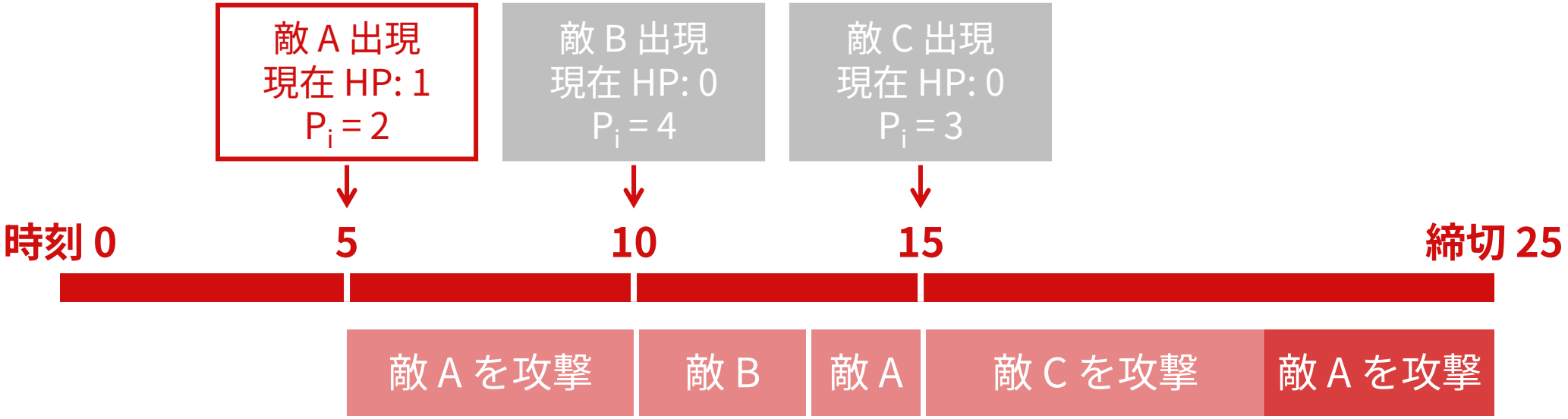
最適戦略: 例


敵 A 出現
現在 HP: 5
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 6
$P_i = 3$
時刻 0
5
10
15
21
締切 25
敵 A を攻撃
敵 B
敵 A
敵 C を攻撃

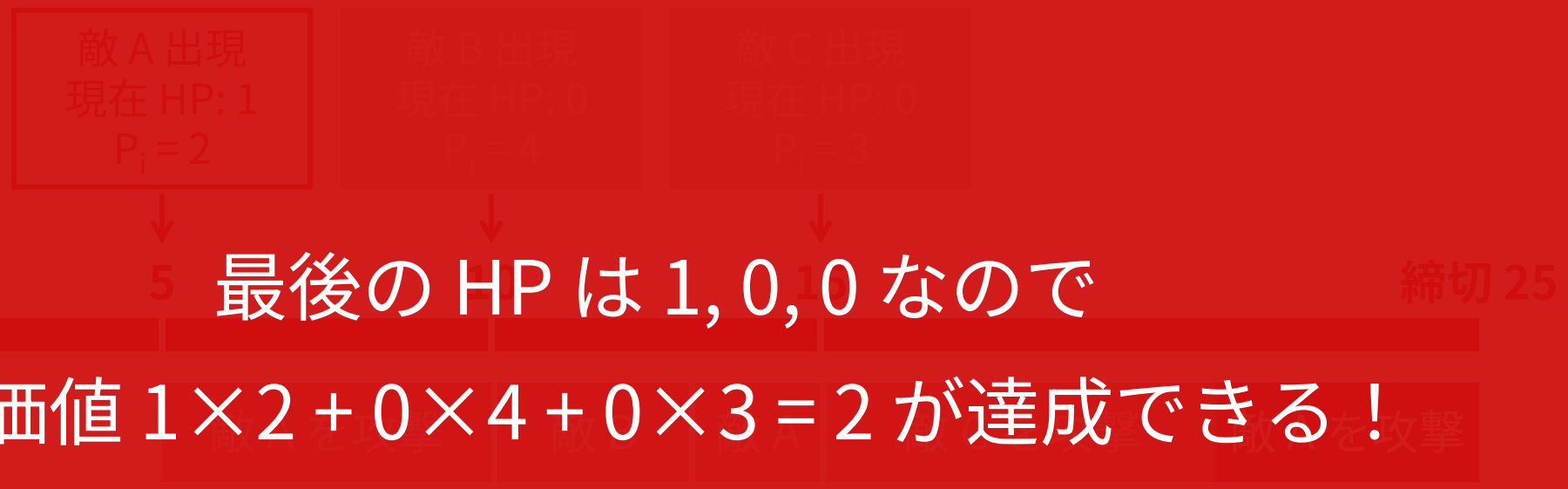
最適戦略: 例


敵 A 出現
現在 HP: 5
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 0
$P_i = 3$
時刻 0
5
10
15
21
締切 25
敵 A を攻撃
敵 B
敵 A
敵 C を攻撃
今倒せる敵の中で
最も P_i が大きいのは敵 A

最適戦略: 例


敵 A 出現
現在 HP: 1
$P_i = 2$
敵 B 出現
現在 HP: 0
$P_i = 4$
敵 C 出現
現在 HP: 0
$P_i = 3$
時刻 0
5
10
15
締切 25
敵 A を攻撃
敵 B
敵 A
敵 C を攻撃
敵 A を攻撃

最適戦略: 例



最後の HP は 1, 0, 0 なので

評価値 1×2 + 0×4 + 0×3 = 2 が達成できる！

小課題 3

攻撃する敵が変わる可能性のあるタイミングは以下

- 敵が出現したタイミング
- 敵の HP がゼロになったタイミング

合計 $2N$ 回しかなく、P_i の最大値を求めるのに $O(N)$ かかるので各難易度について計算量は $O(QN^2)$

答えで二分探索をして $O(QN^2 \log L)$ → **小課題 3 に AC**

14 点

小課題 4

小課題 4

- $N \leq 6000, Q = 3$
- N の制約が 30 から 6000 に上がったので、小課題 3 の解法 (計算量 $O(QN^2 \log L)$ が通らない

小課題３の見直し

小課題３では P_i の最大値を求めるのに $O(N)$ かかった

▼

優先度付きキューで高速化できないか？

小課題 3 の見直し

- 優先度付きキューを使うと、計算量が $O(QN \log N \log L)$ になり **小課題 4 まで通る**
- ただし、実装を考えるのがそこまで簡単ではない（※実装量自体は少ない)

24 点

小課題 5, 6

小課題 5, 6

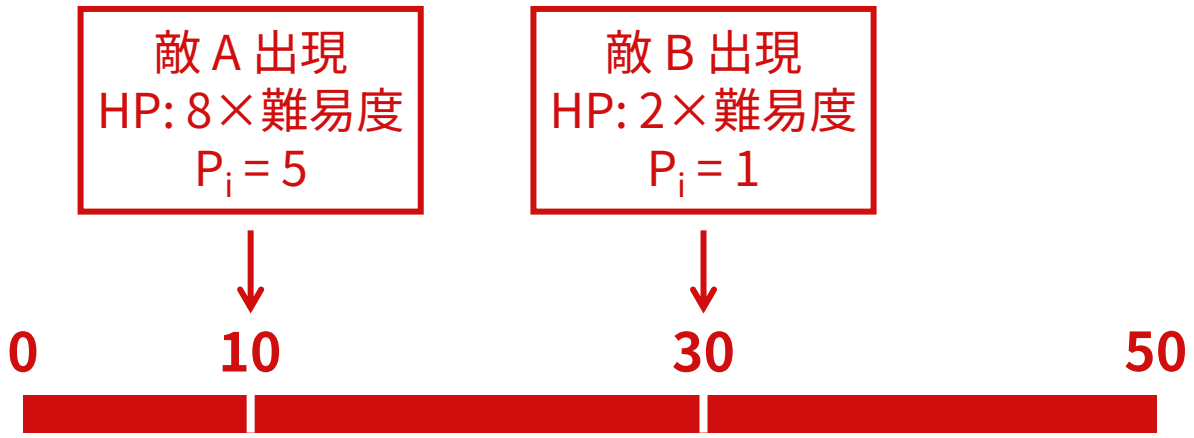
- $N \leq 30, Q \leq 1000000$
- Q に対して N が非常に小さい

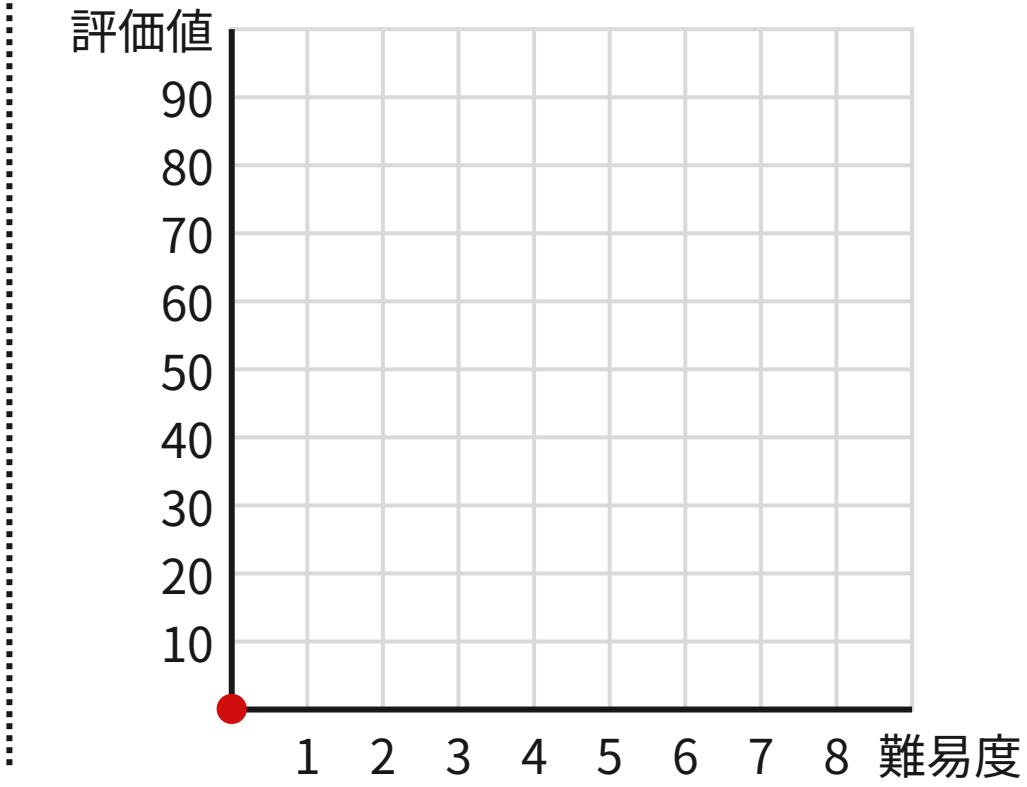
重要な考察

難易度 l が増えた時に可能な評価値の最大値 $f(l)$ が
どう変化するのかを観察してみよう！

重要な考察: 例

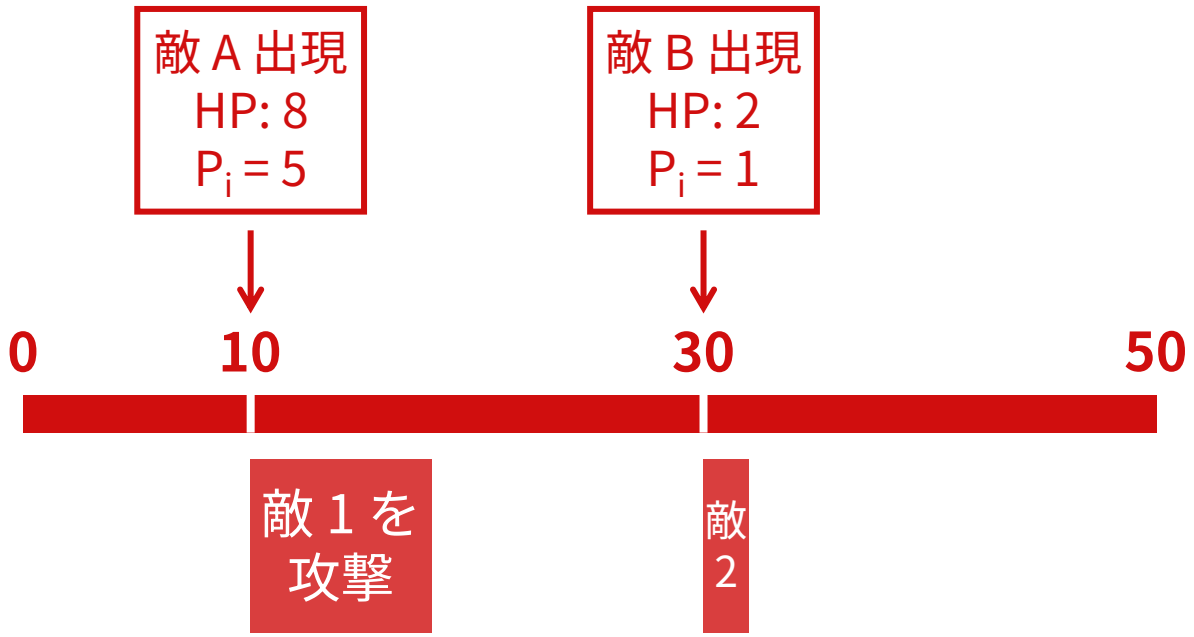
まずは以下の簡単なケースを考える

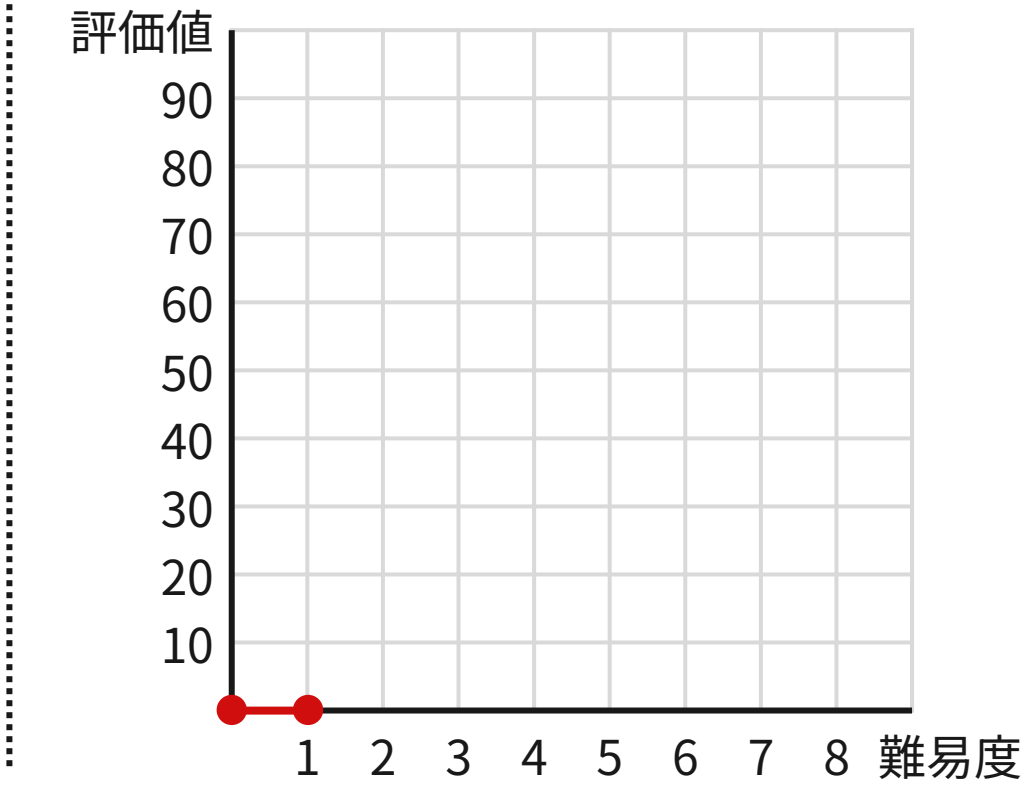




重要な考察: 例

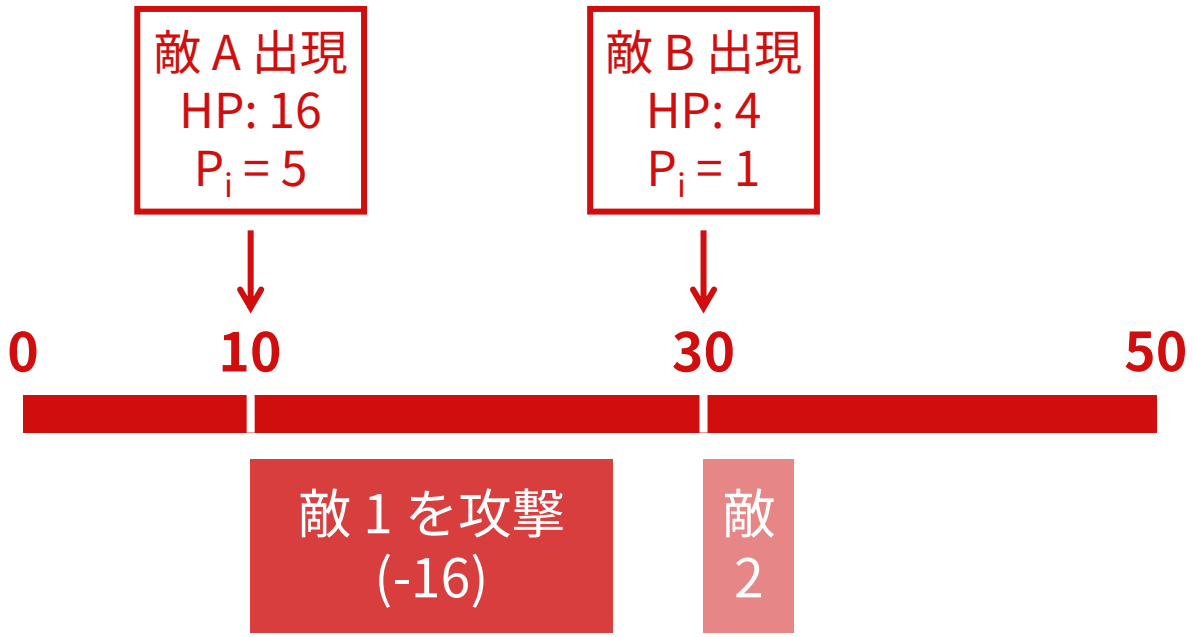
難易度 1 の場合、評価値 0 を達成することができる

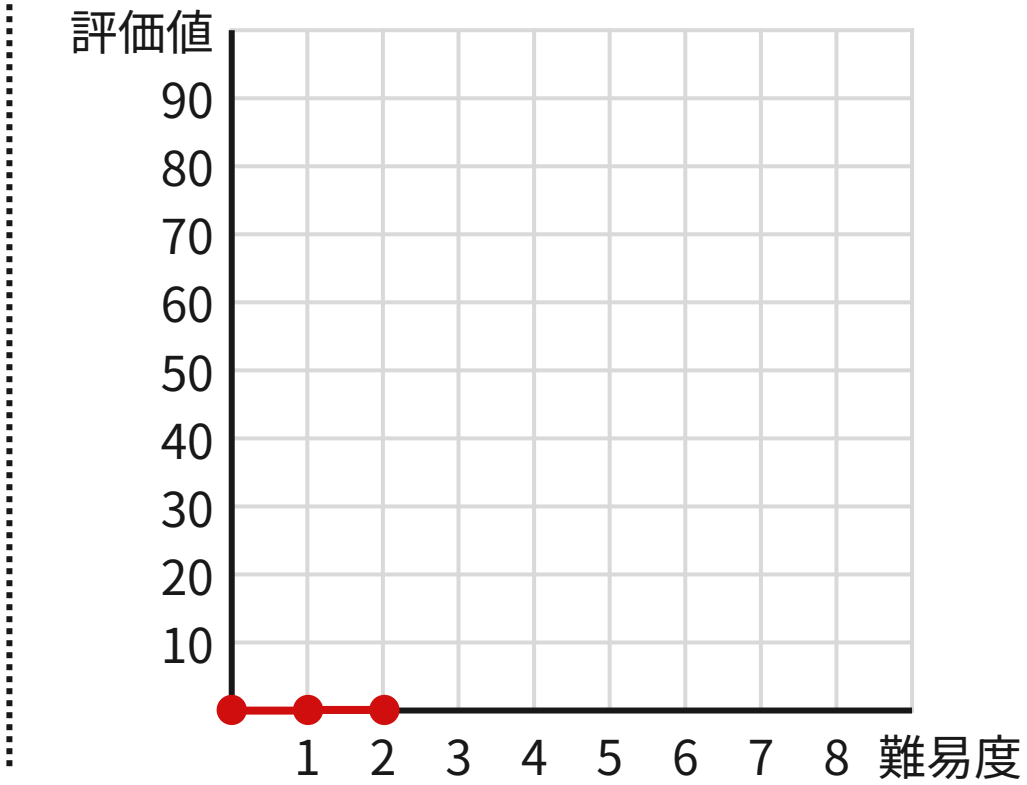




重要な考察: 例

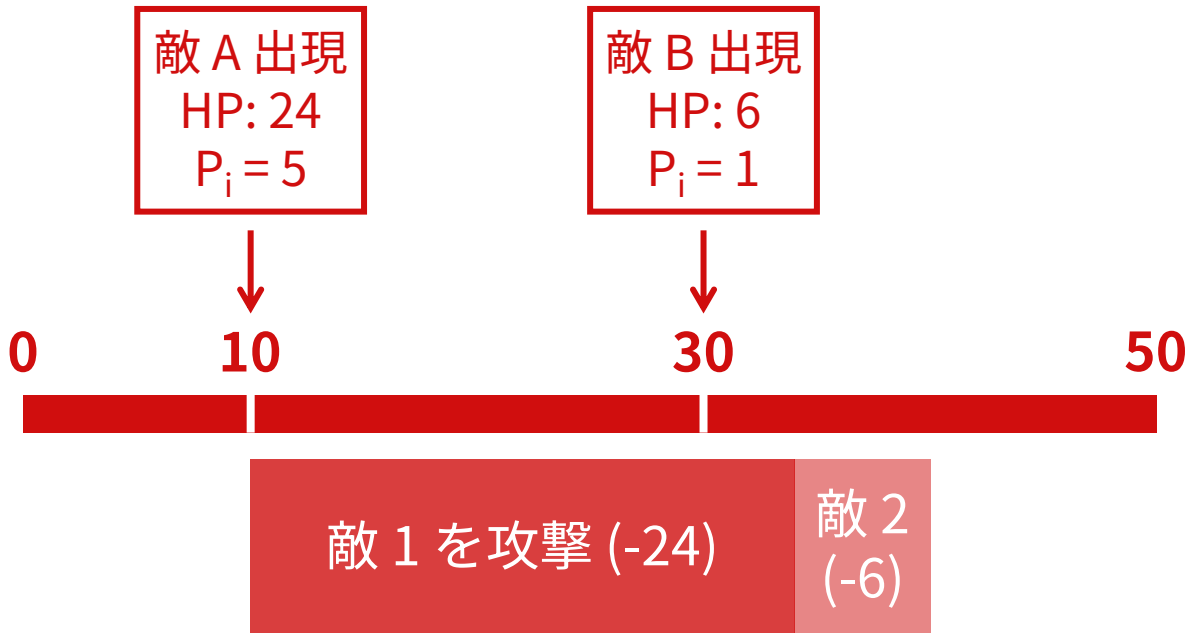
難易度 2 の場合も、評価値 0 を達成することができる

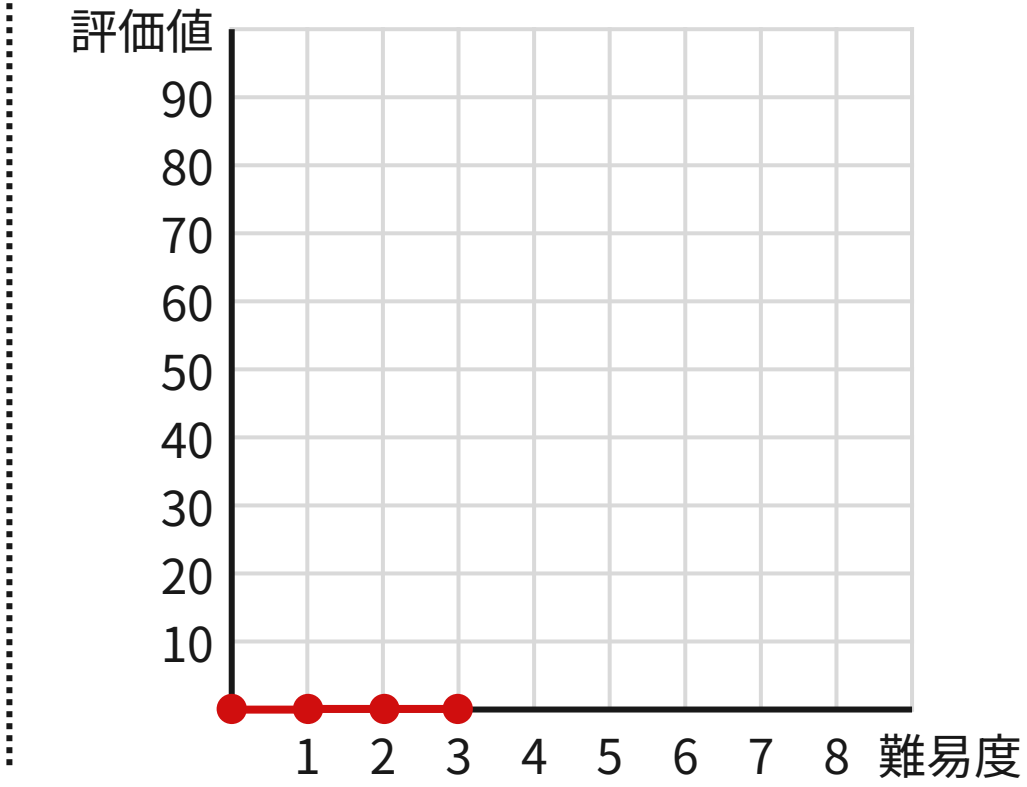




重要な考察: 例

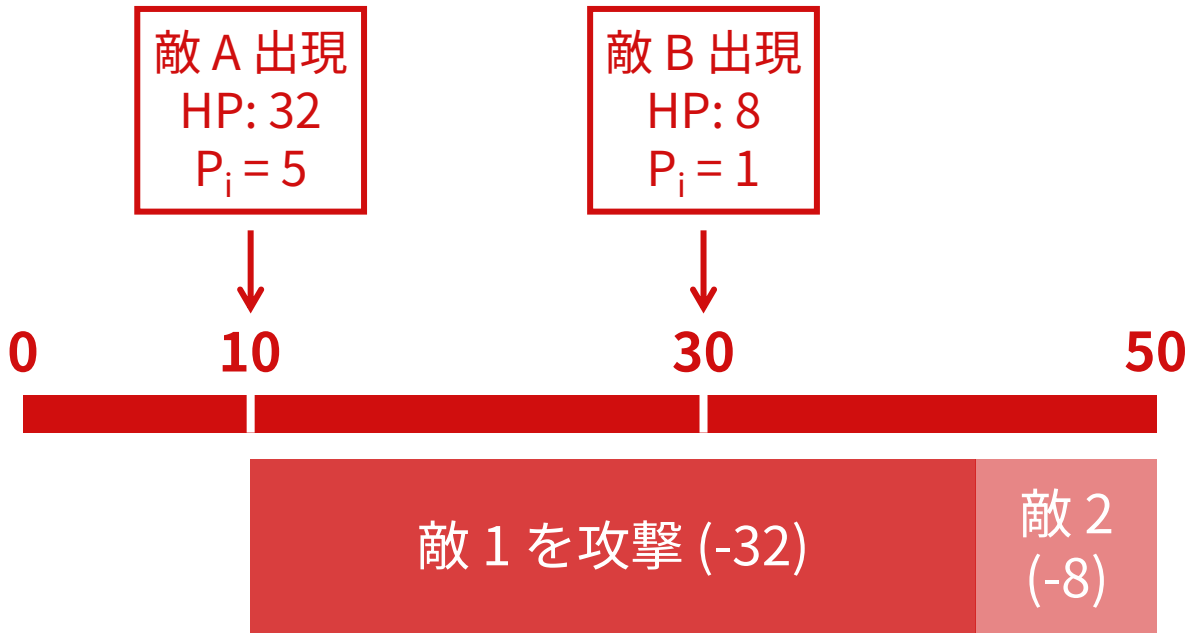
難易度 3 の場合も、評価値 0 を達成することができる

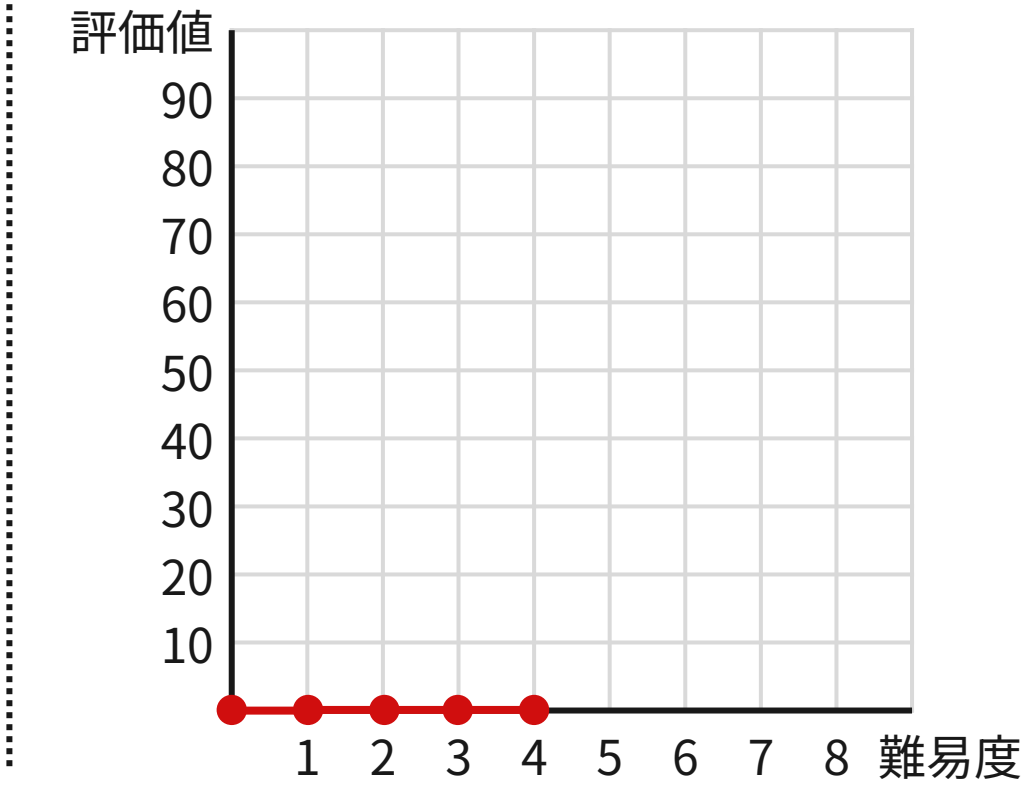




重要な考察: 例

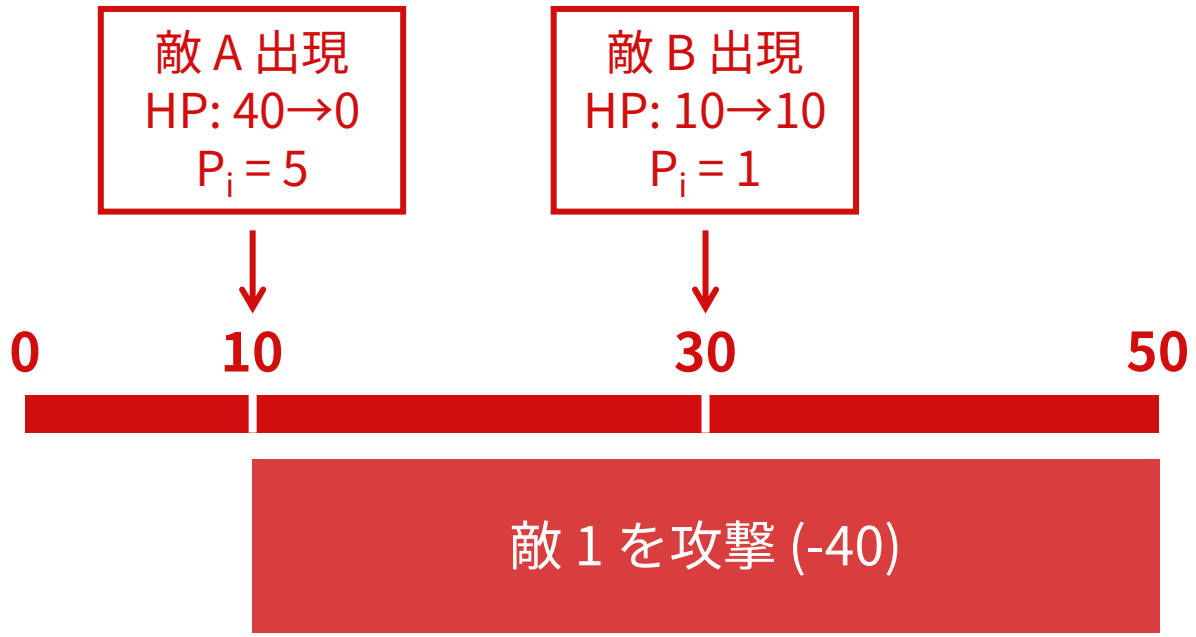
難易度 4 の場合も、評価値 0 を達成することができる

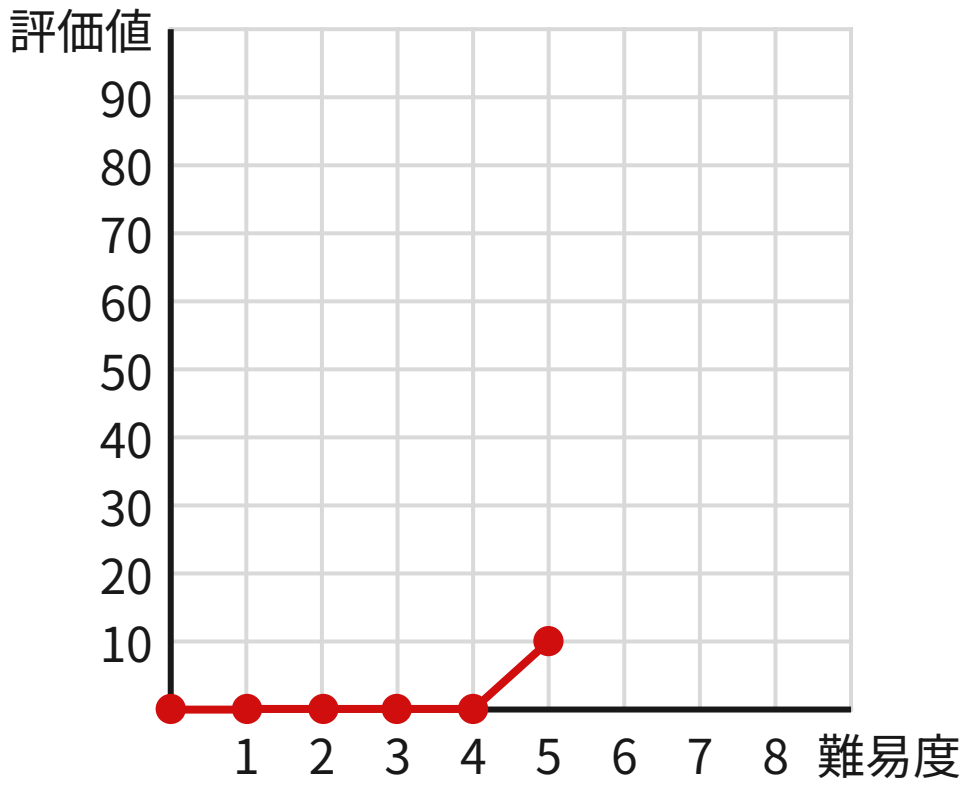




重要な考察: 例

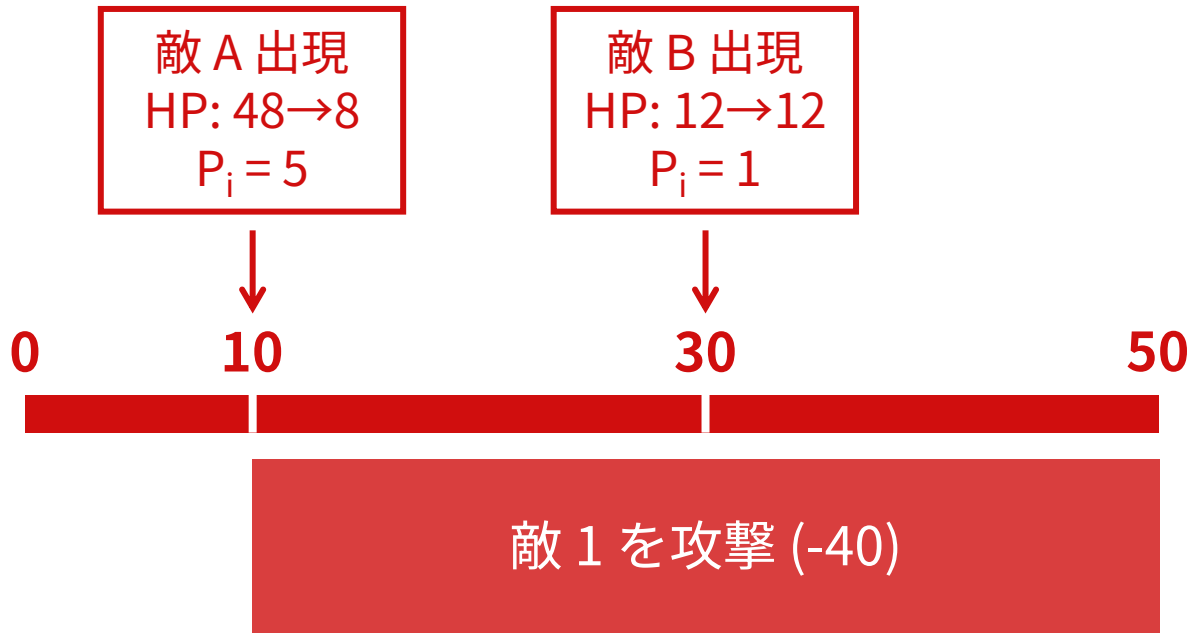
難易度 5 の場合、評価値 10 を達成することができる

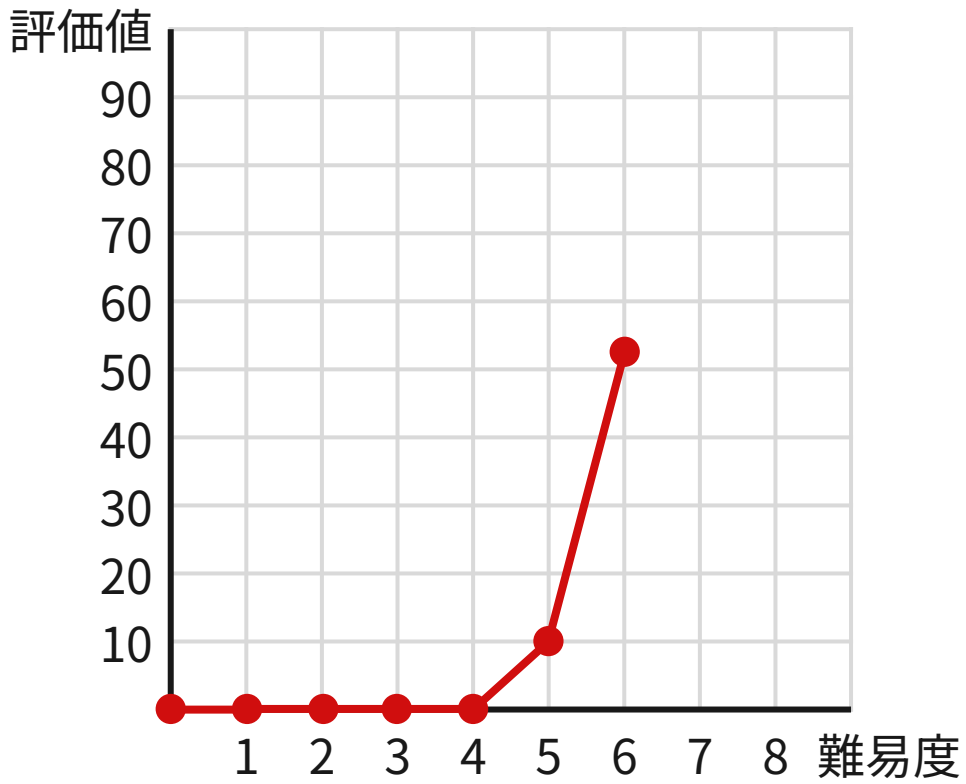




重要な考察: 例

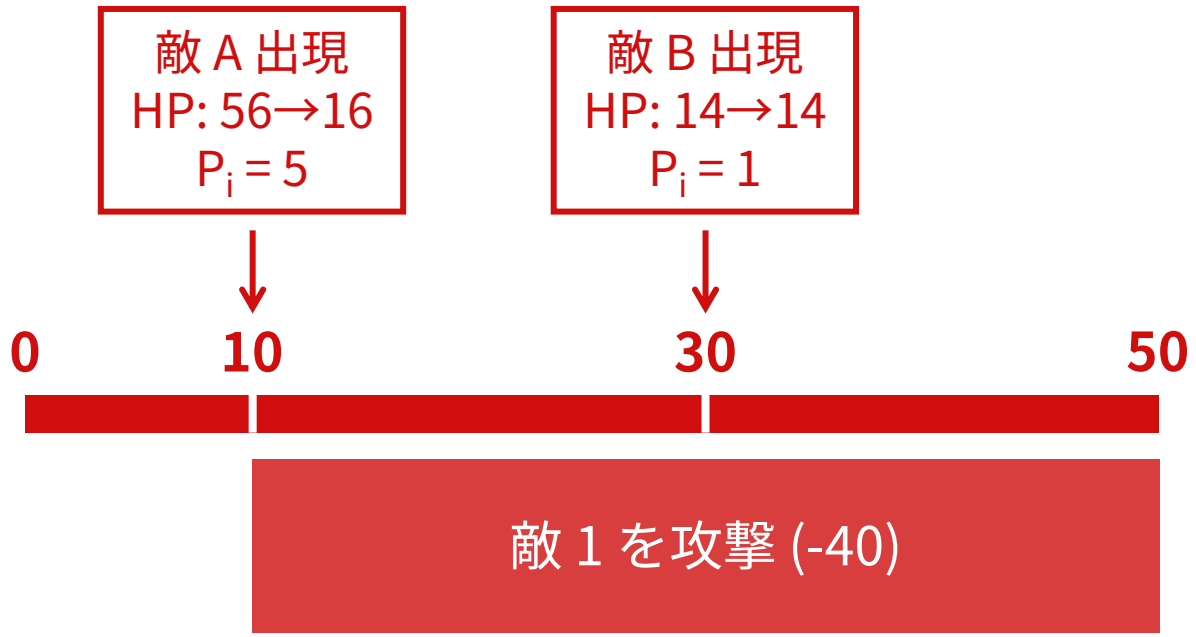
難易度 6 の場合、評価値 8×5 + 1×12 = 52 を達成可能

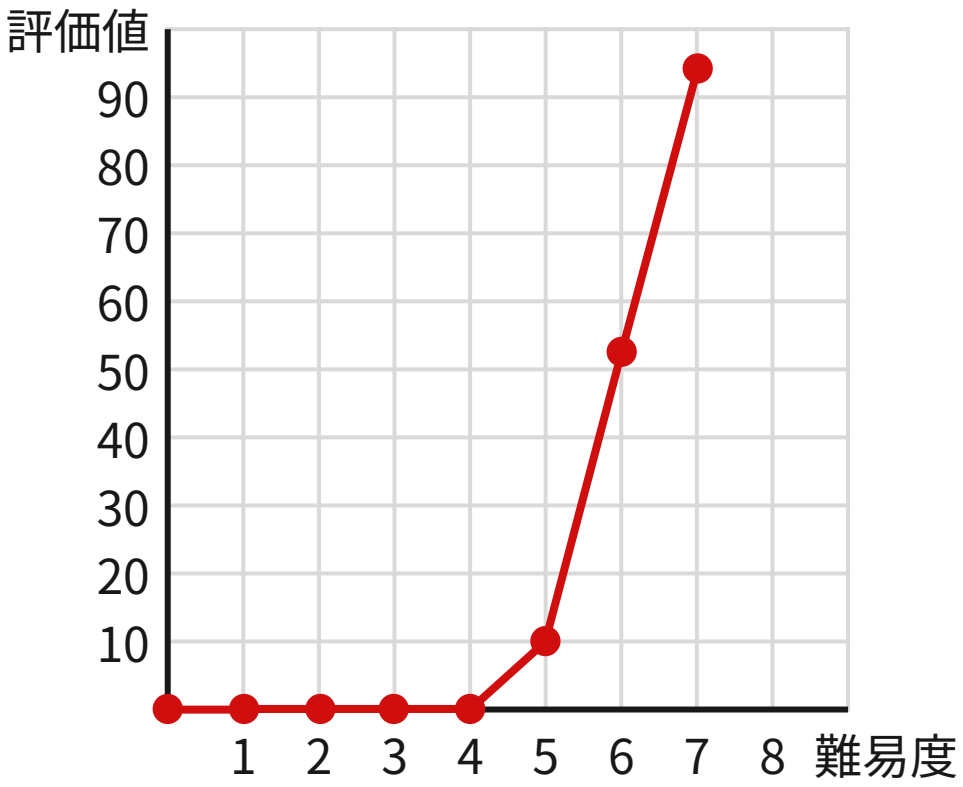




重要な考察: 例

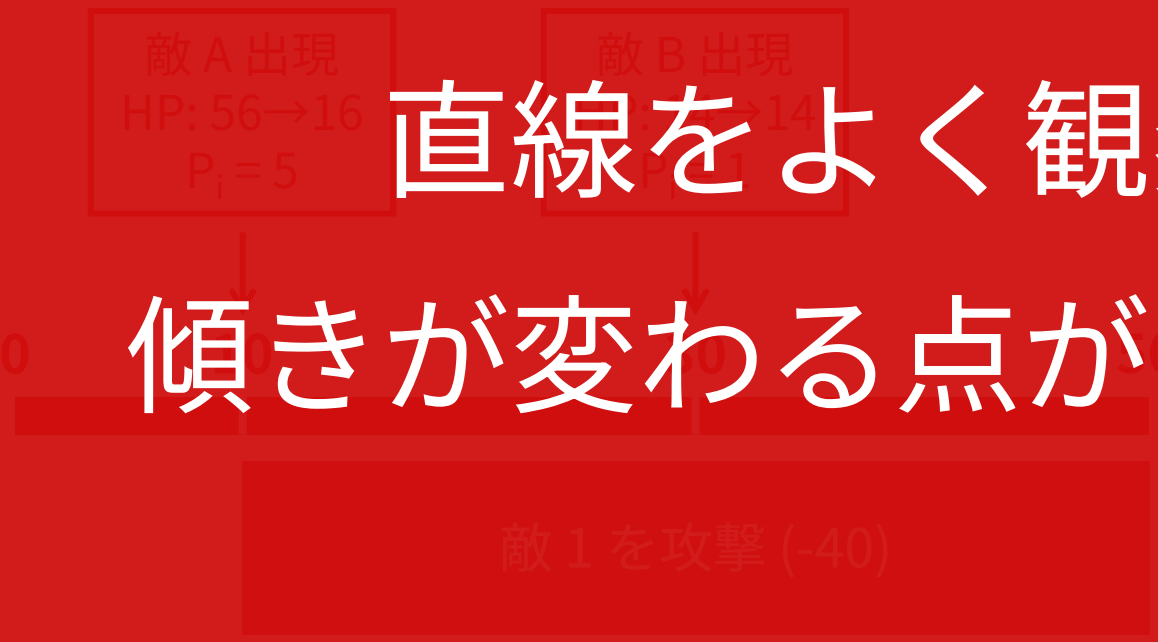
難易度 7 の場合、評価値 16×5 + 1×14 = 94 を達成可能

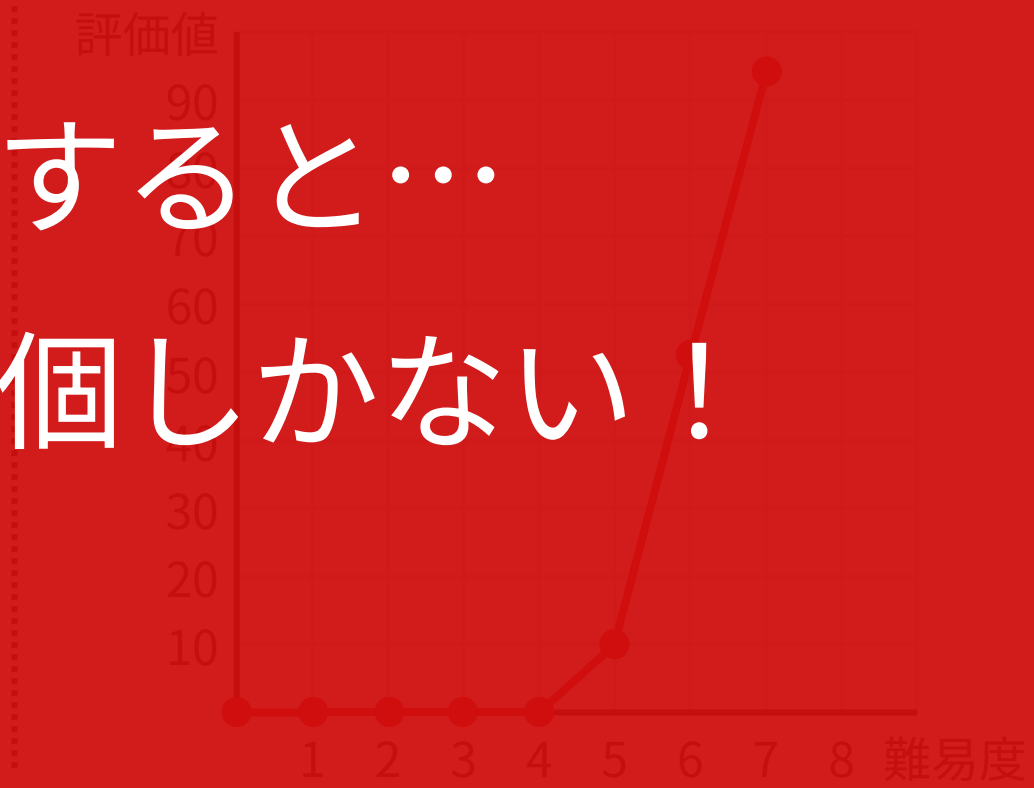




重要な考察: 例

難易度 7 の場合、評価値 16×5 + 1×14 = 94 を達成可能





直線をよく観察すると…

傾きが変わる点が 2 個しかない！

重要な考察

実はどんなケースでも傾きが変わる点が
多くないのでは？

重要な考察

- 実は、傾きが変わる回数が $O(N^2)$ 以下であることが証明できる (証明は満点解法の考察を参照)
- つまり、傾きが変わる点を上手く求めれば $N \leq 30$ に正解することができそう

➡ **傾きが変わる点をどう求めるか？**

重要な考察

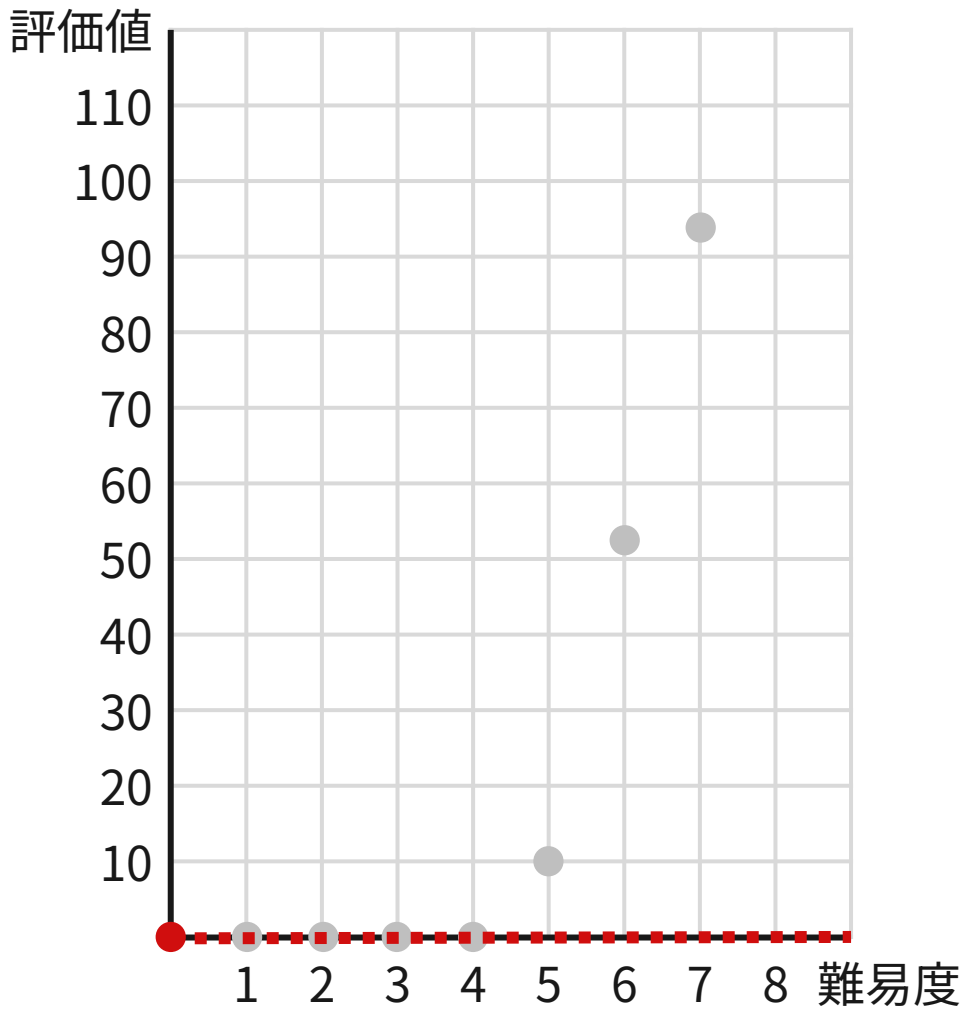
実は、グラフは凸※ なので

二分探索できる

※傾きが単調増加

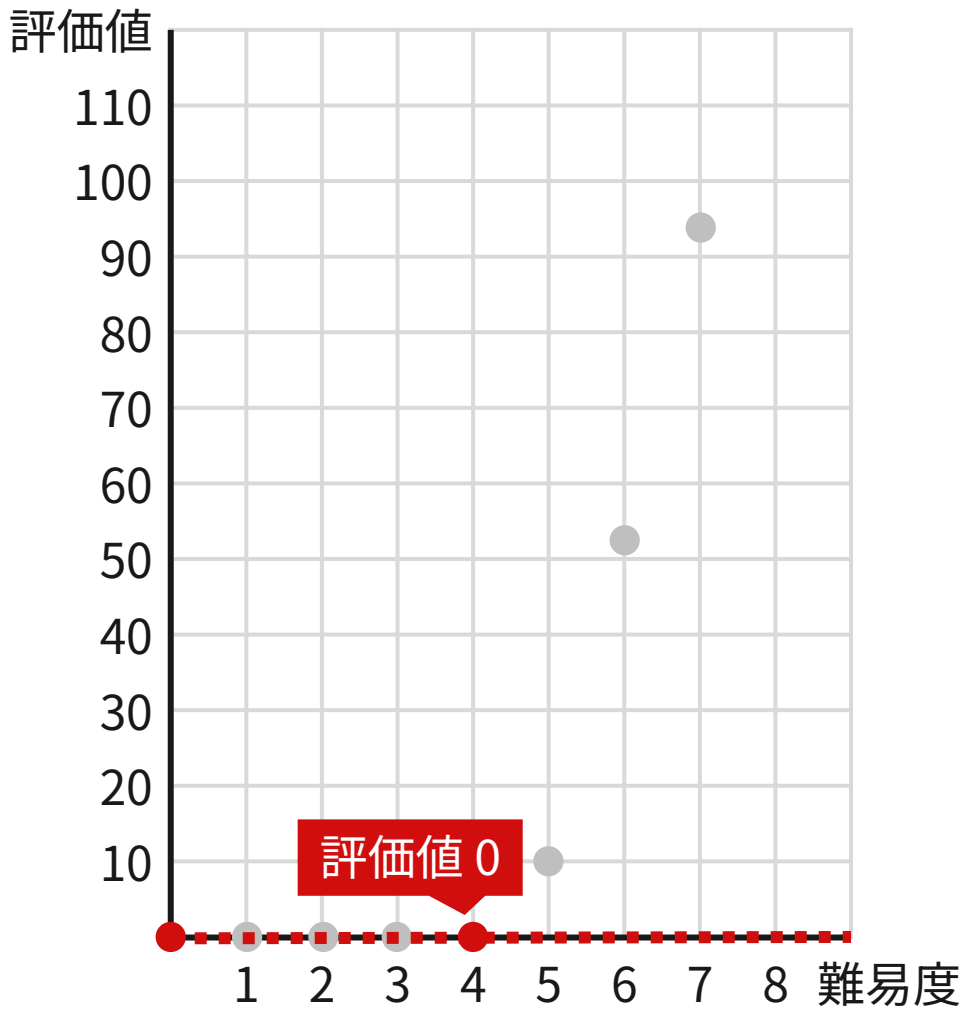
二分探索の例

- 最初の傾きは 0
- 次に傾きが変わる点はいつか？



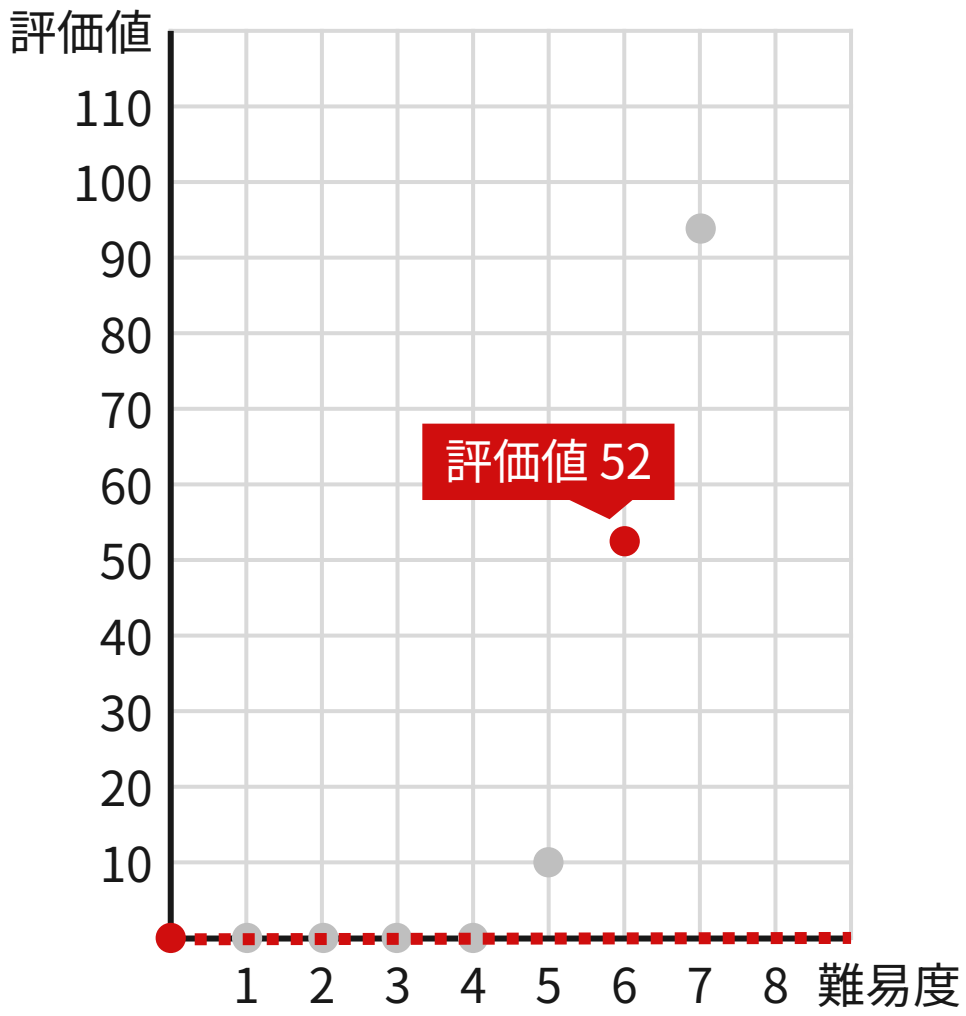
二分探索の例

- 最初の傾きは 0
- 次に傾きが変わる点はいつか？
  - 難易度 4 では変わらない



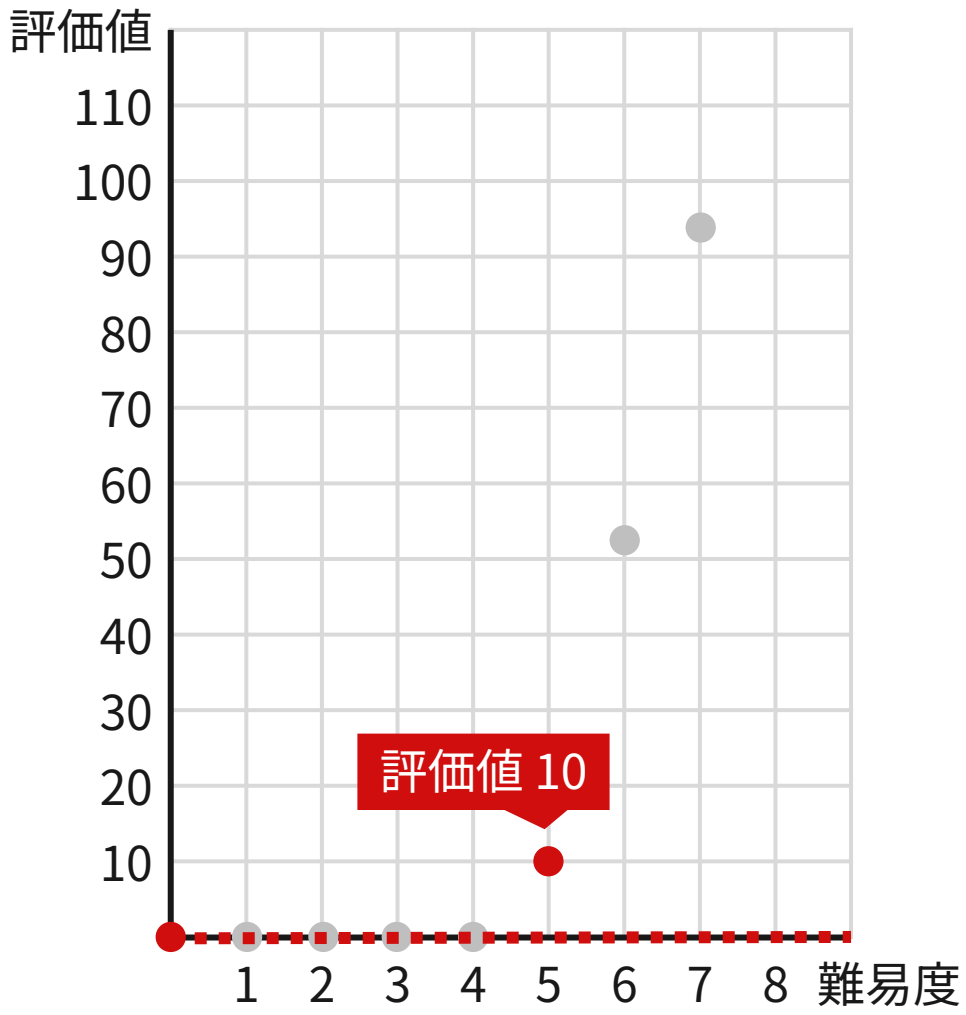
二分探索の例

- 最初の傾きは 0
- 次に傾きが変わる点はいつか？
  - 難易度 4 では変わらない
  - 難易度 6 では変わる



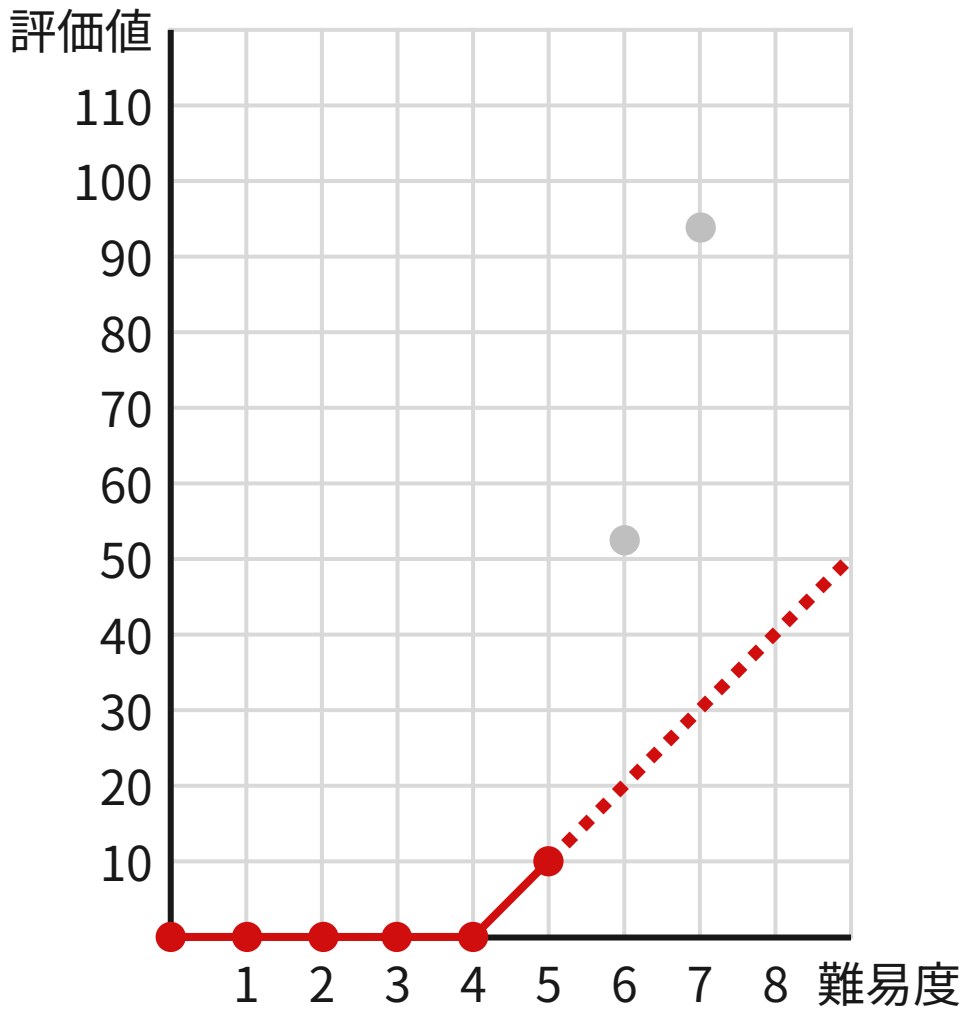
二分探索の例

- 最初の傾きは 0
- 次に傾きが変わる点はいつか？
  - 難易度 4 では変わらない
  - 難易度 6 では変わる
  - 難易度 5 でも変わる
  - → 次に傾きが変わるのは難易度 5



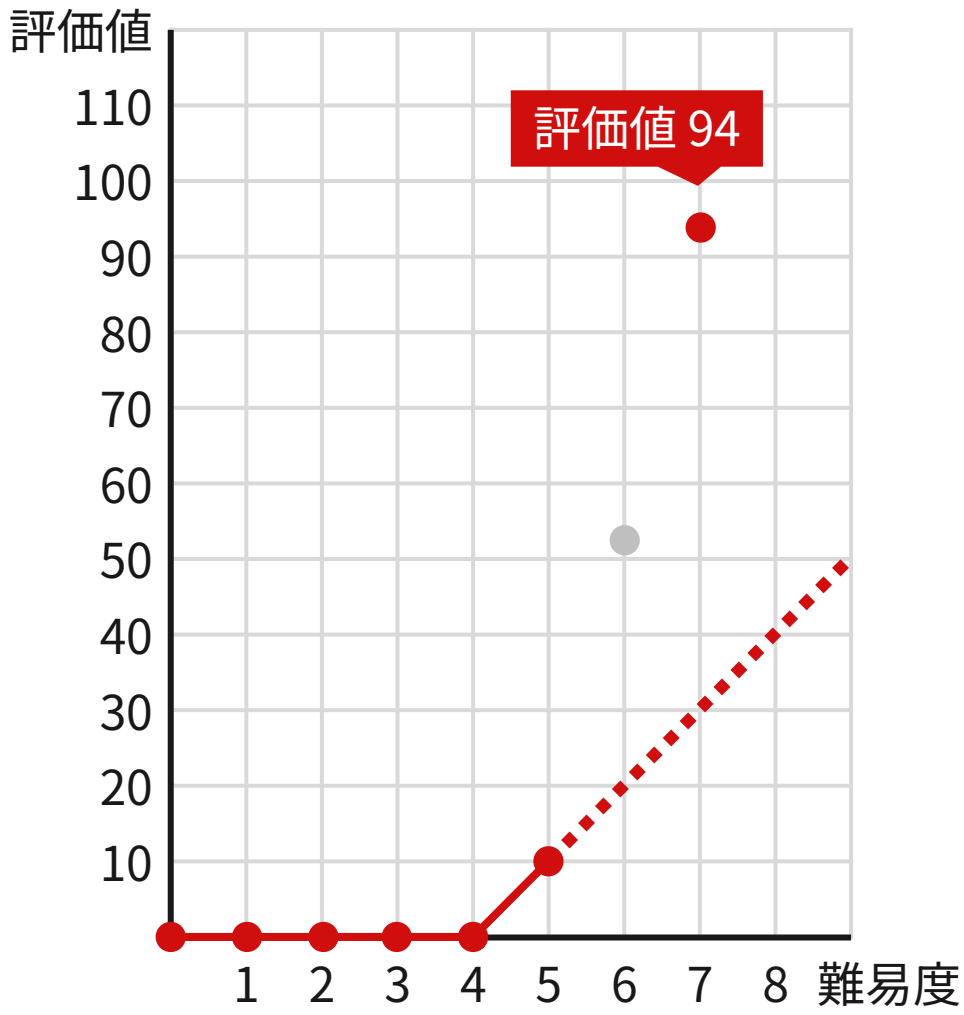
二分探索の例

- 現在の傾きは 10
- 次に傾きが変わる点はいつか？



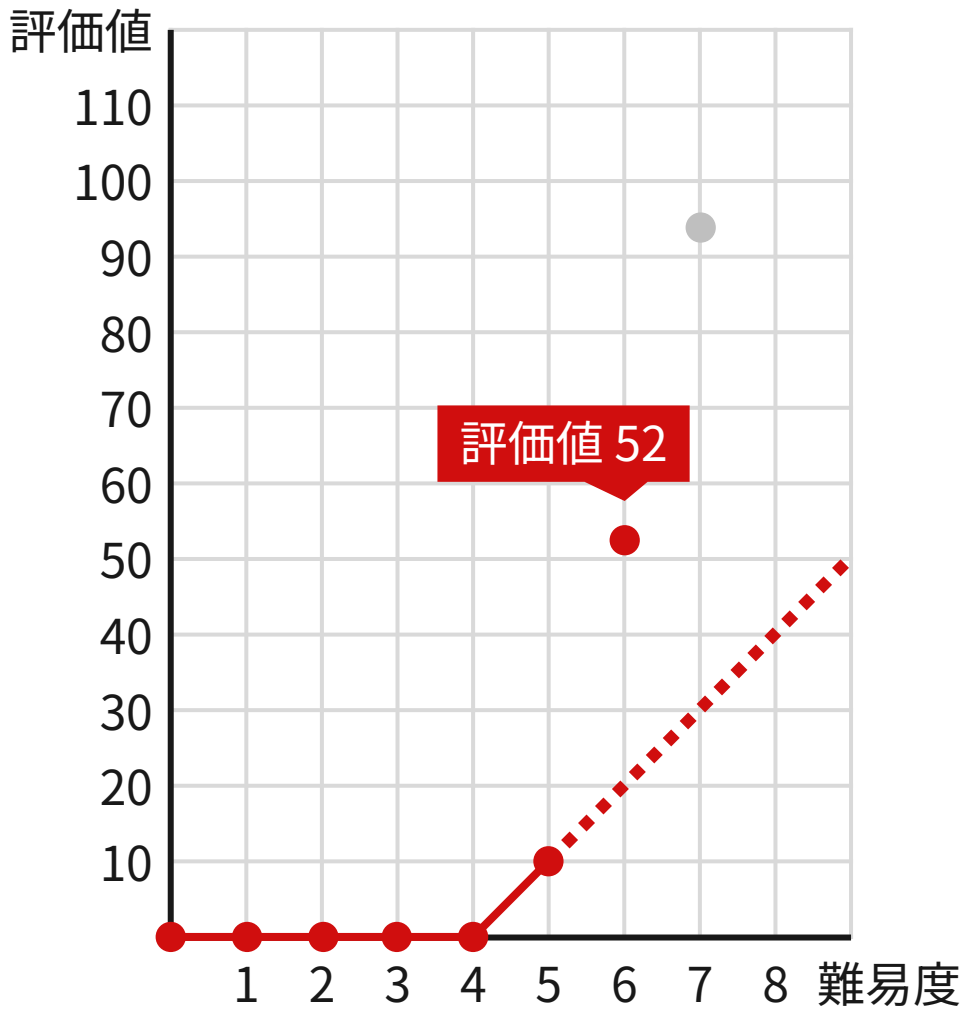
二分探索の例

- 現在の傾きは 10
- 次に傾きが変わる点はいつか？
  - 難易度 7 では変わる



二分探索の例

- 現在の傾きは 10
- 次に傾きが変わる点はいつか？
  - 難易度 7 では変わる
  - 難易度 6 でも変わる
  - → 次に傾きが変わるのは難易度 6



小課題 5

- このように二分探索をすると、合計で $O(N^2 \log L)$ 個の難易度について※ 最善評価値を求めればよい
- 最善評価値の計算に $O(N \log N)$ かかるので、全体で $O(N^3 \log N \log L)$ かかる
- $N \leq 30$ では間に合う

59 点

※傾きが変わる点が $O(N^2)$ 個であるため

補足

- 実は傾きが変わる点は $O(N)$ 個なのではないか？
- 証明はできていないが、解説担当者はそうだと思っている
- 定数倍によっては小課題 6 に通るかもしれない

67 点

小課題 7

小課題 7

- 難易度 l のときの最善評価値 $f(l)$ の関数を高速に求めたい
- もし以下の値がわかれば、関数 $f(l)$ もわかる
  - 最適な戦略で動いた時の、P_i が 1 番目に大きいものの攻撃時間
  - 最適な戦略で動いた時の、P_i が 2 番目に大きいものの攻撃時間
  - 最適な戦略で動いた時の、P_i が 3 番目に大きいものの攻撃時間
  - :
- 一体どうすればこれらの値が求まるのか？

小課題 7

- 難易度 l のときの最善評価値 $f(l)$ の関数を高速に求めたい
- もし以下の値がわかれば、関数 $f(l)$ もわかる
  - 最適な戦略で動いた時の、P_i が上位 1 個のものの合計攻撃時間
  - 最適な戦略で動いた時の、P_i が上位 2 個のものの合計攻撃時間
  - 最適な戦略で動いた時の、P_i が上位 3 個のものの合計攻撃時間
  - :
- 一体どうすればこれらの値が求まるのか？

重要な言い換え

最適な戦略における
P_i が上位 k 個のものの
合計攻撃時間

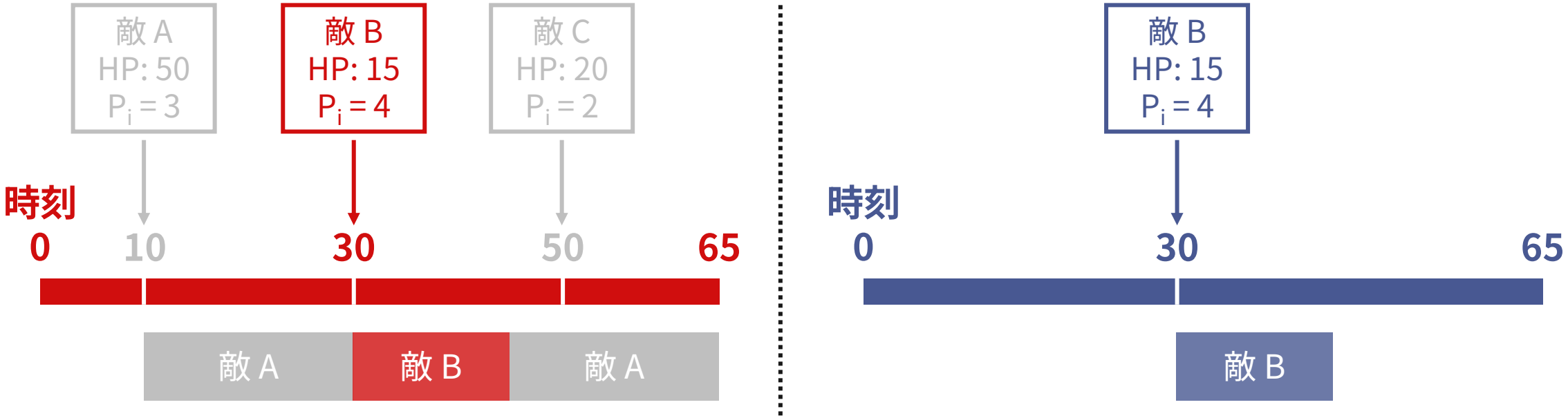
=

その k 体の敵だけがいるときの
(つまり $N = k$ のときの)
合計攻撃時間

重要な言い換え

よくわからないと思うので
例を説明します

重要な言い換え: 例 (k=1)



N = 3 のときの敵 B の合計攻撃時間 (15) が

敵 B だけを残したとき (N = 1) の合計攻撃時間 (15) と同じ

重要な言い換え

なぜそんなことが起こるのか？

重要な言い換え

なぜそんなことが起こるのか？

▼

小課題４の貪欲法において、P_i が上位 k 個の敵への攻撃は
他のどの敵よりも優先されるため

小課題 7

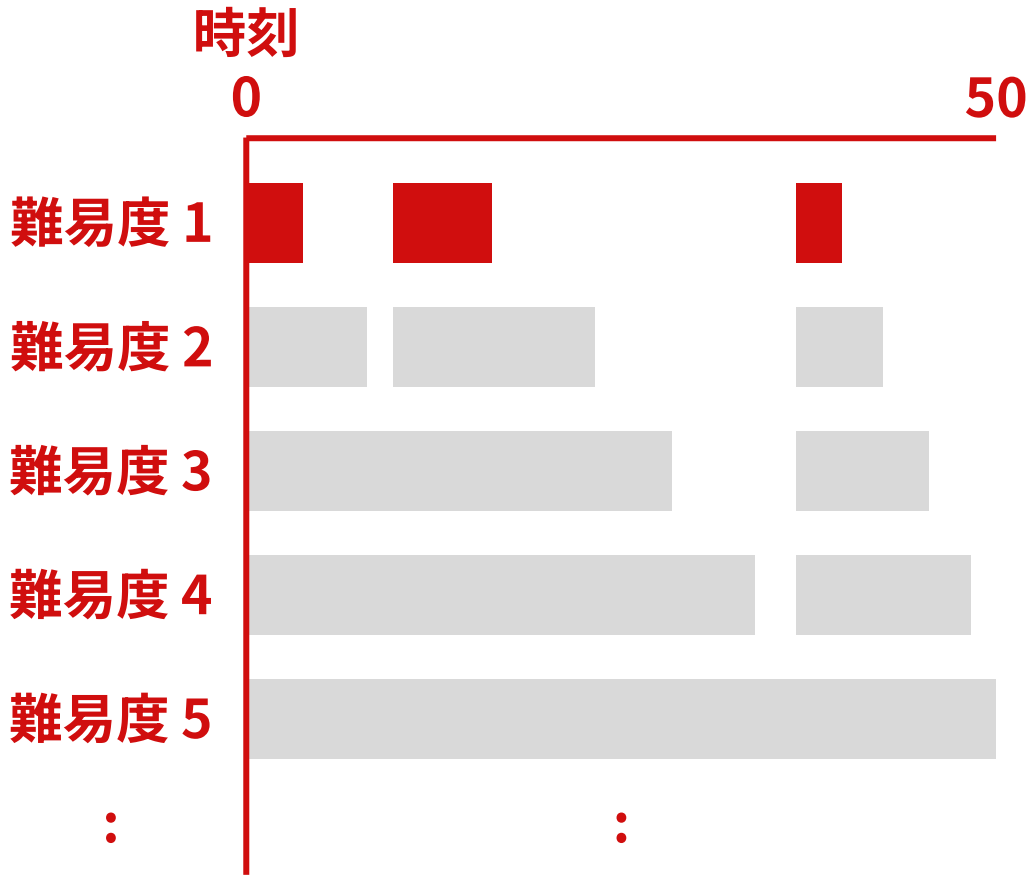
では、どのようにして

合計攻撃時間の関数を計算するか？

小課題 7

- これは単に、優先度付きキューで（難易度を増やしていった時の）敵の攻撃時間の衝突を右図のようにシミュレーションすればよい

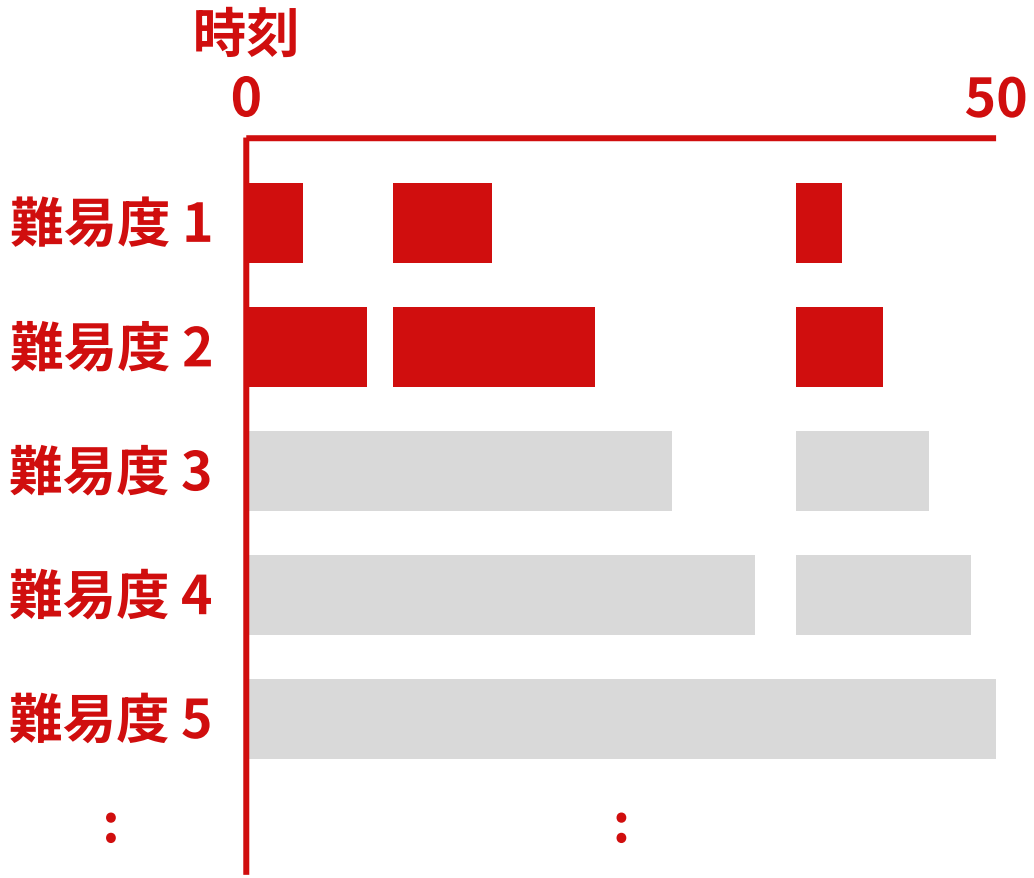
- 計算量は $O(N^2 \log N)$ となり小課題 7 に通る



小課題 7

- これは単に、優先度付きキューで（難易度を増やしていった時の）敵の攻撃時間の衝突を右図のようにシミュレーションすればよい

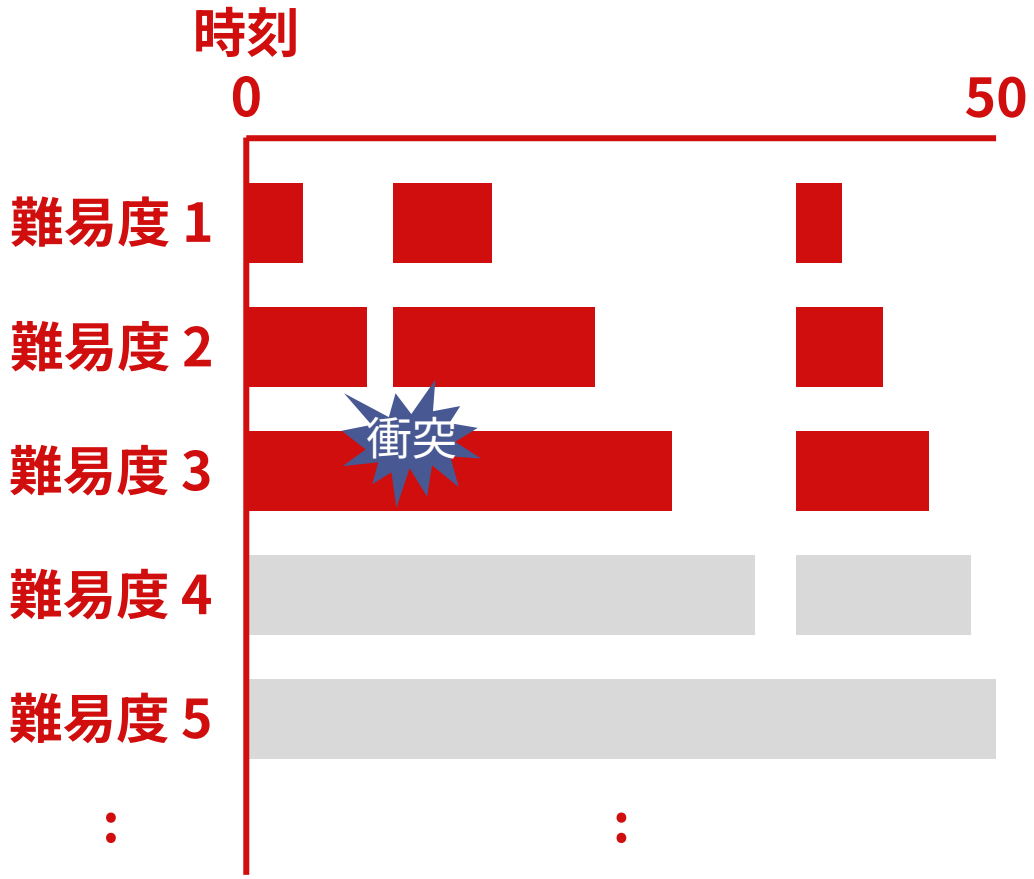
- 計算量は $O(N^2 \log N)$ となり小課題 7 に通る



小課題 7

- これは単に、優先度付きキューで（難易度を増やしていった時の）敵の攻撃時間の衝突を右図のようにシミュレーションすればよい

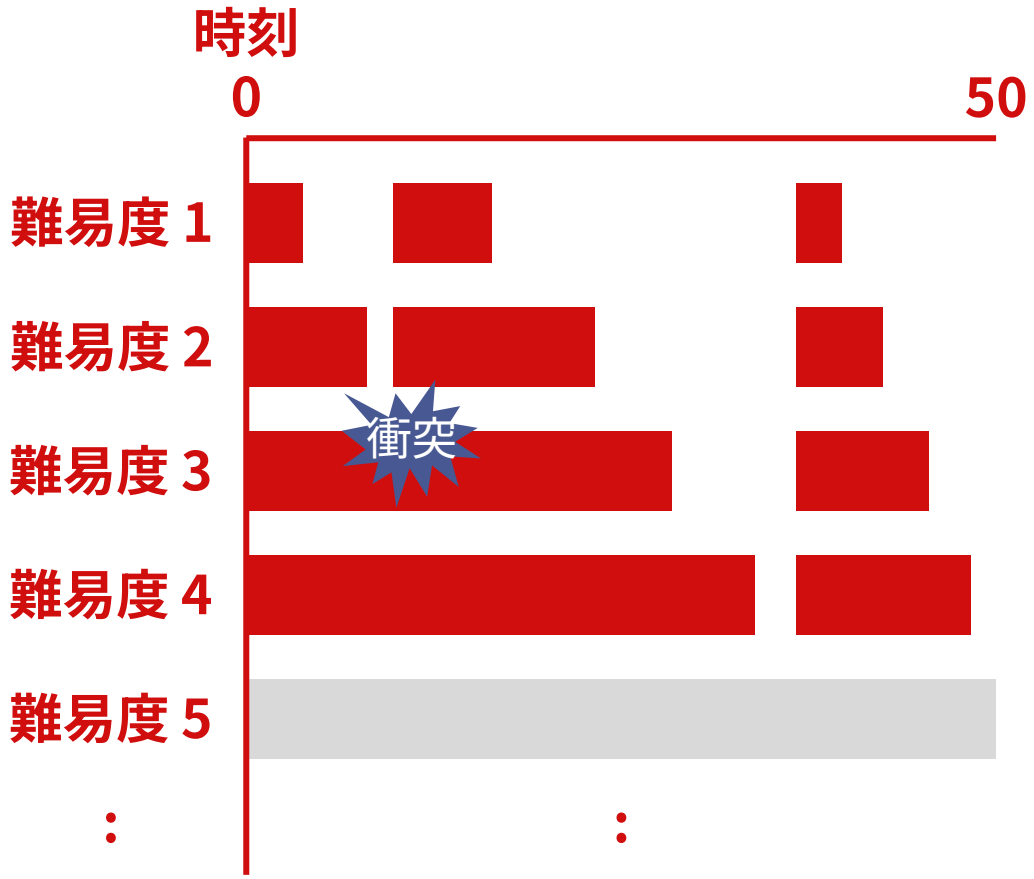
- 計算量は $O(N^2 \log N)$ となり小課題 7 に通る



小課題 7

- これは単に、優先度付きキューで（難易度を増やしていった時の）敵の攻撃時間の衝突を右図のようにシミュレーションすればよい

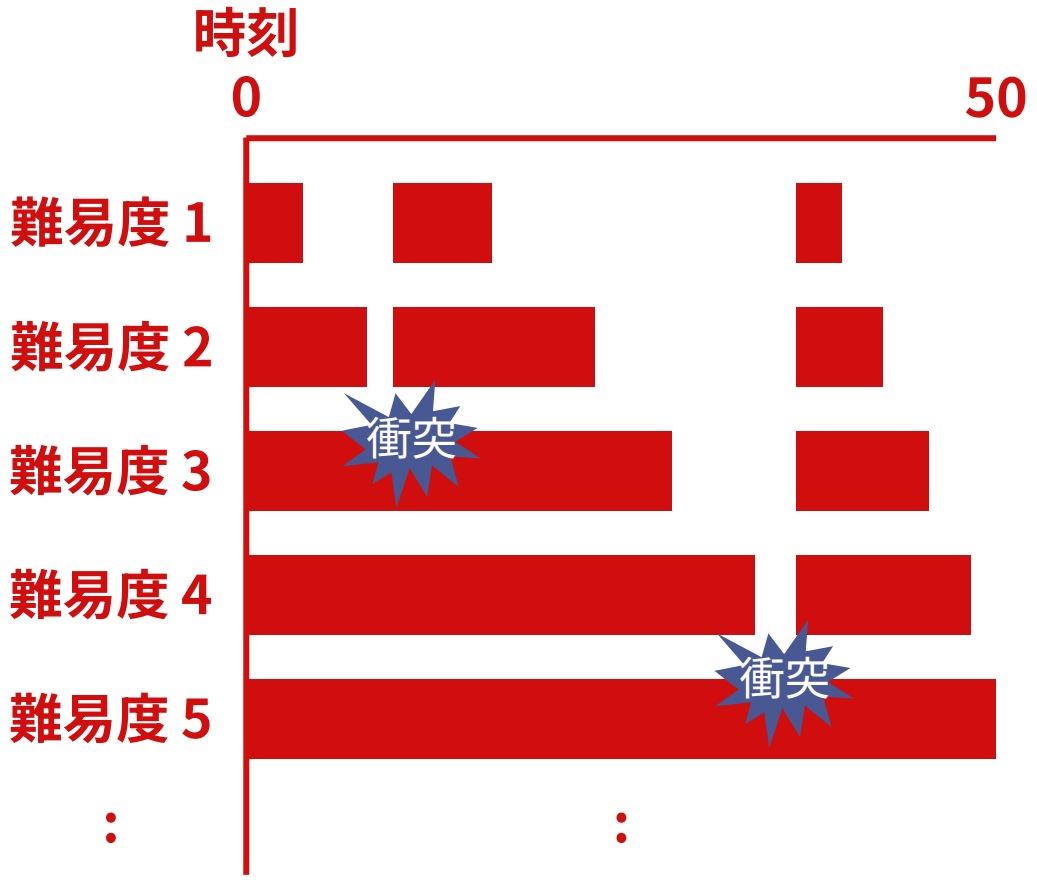
- 計算量は $O(N^2 \log N)$ となり小課題 7 に通る



小課題 7

- これは単に、優先度付きキューで（難易度を増やしていった時の）敵の攻撃時間の衝突を右図のようにシミュレーションすればよい

- 計算量は $O(N^2 \log N)$ となり小課題 7 に通る



満点

満点解法

- 実は、後ろから（終了時刻ギリギリの敵から）考えてスタックを使うと、$O(N \log N)$ でシミュレーションしていたところが $O(N)$ になる

- 全体の計算量が $O(N^2)$ になり、満点まで通る

100 点

得点分布

得点分布 (理想)