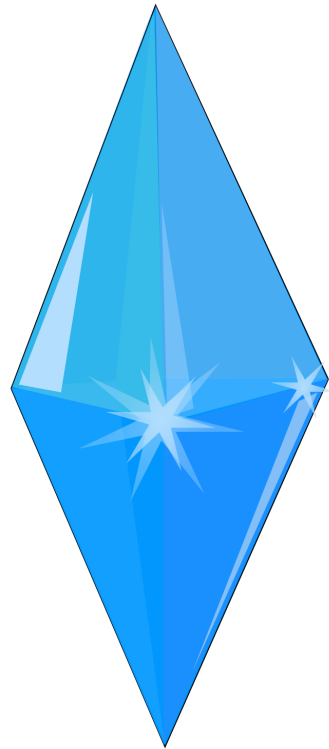


KUPC2017 I

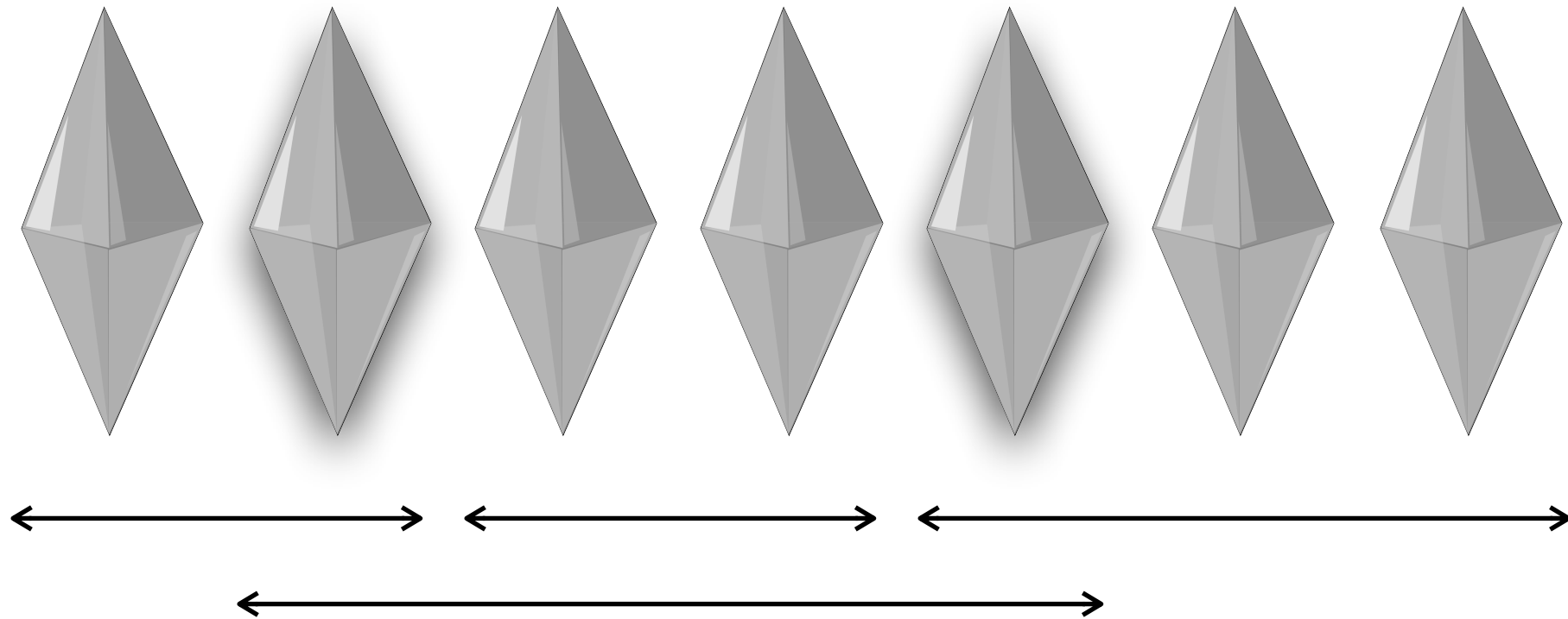
Activate It!!



drafear

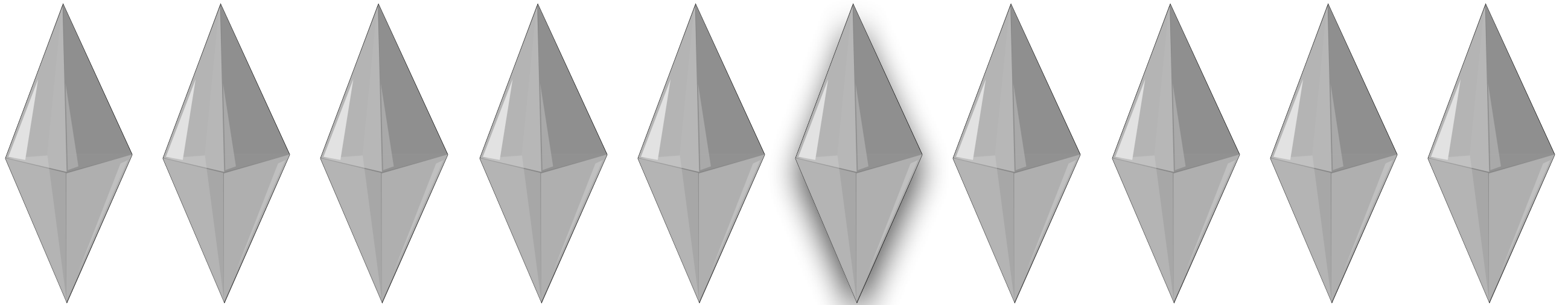
問題概要

- n 個の魔法石が一行に並んでいる
- 魔法 i を詠唱すると $[l_i, r_i]$ を活性化させる
- ただし x_1, \dots, x_n 番目の魔法石
(以降、特別な魔法石と呼ぶことにします)
を全て活性化させた後はこれ以上新たに活性化されない
- 活性化できる魔法石の個数を最大化せよ



例

10	4	1
6		
1	2	
4	5	
2	7	
5	10	



← 魔法1 →

← 魔法2 →

← 魔法3 →

← 魔法4 →

例

10 4 1

6

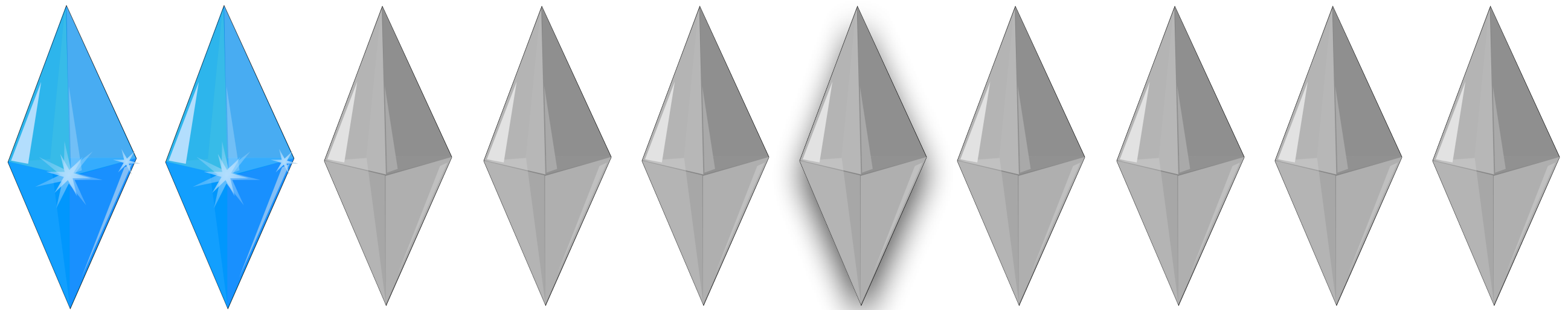
1 2

4 5

2 7

5 10

魔法1を詠唱する



魔法1

魔法2

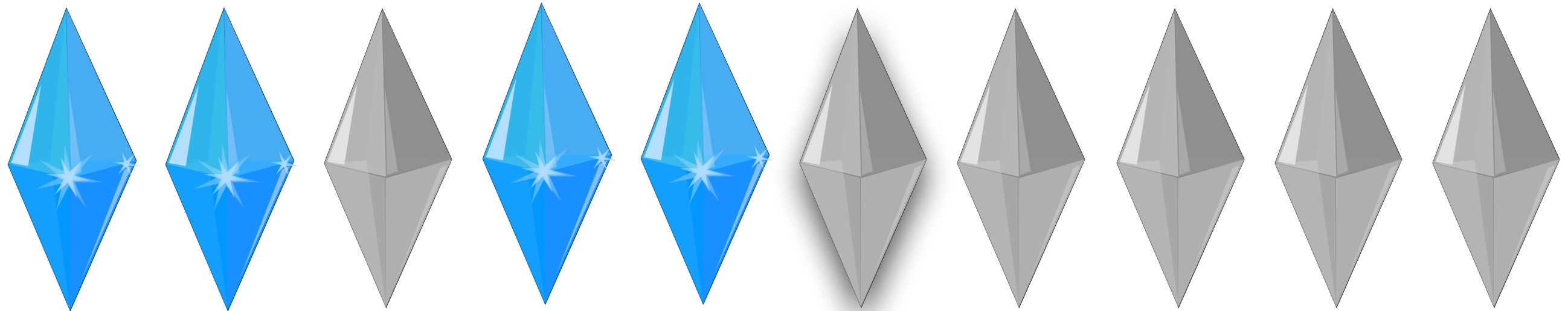
魔法3

魔法4

例

10 4 1
6
1 2
4 5
2 7
5 10

魔法2を詠唱する



魔法1

魔法2

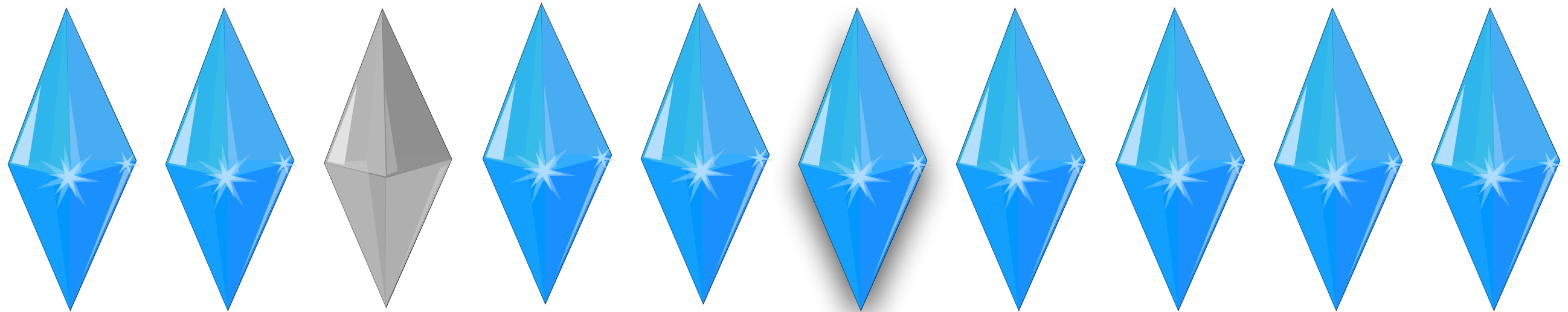
魔法3

魔法4

例

10 4 1
6
1 2
4 5
2 7
5 10

魔法4を詠唱する



← 魔法1 →

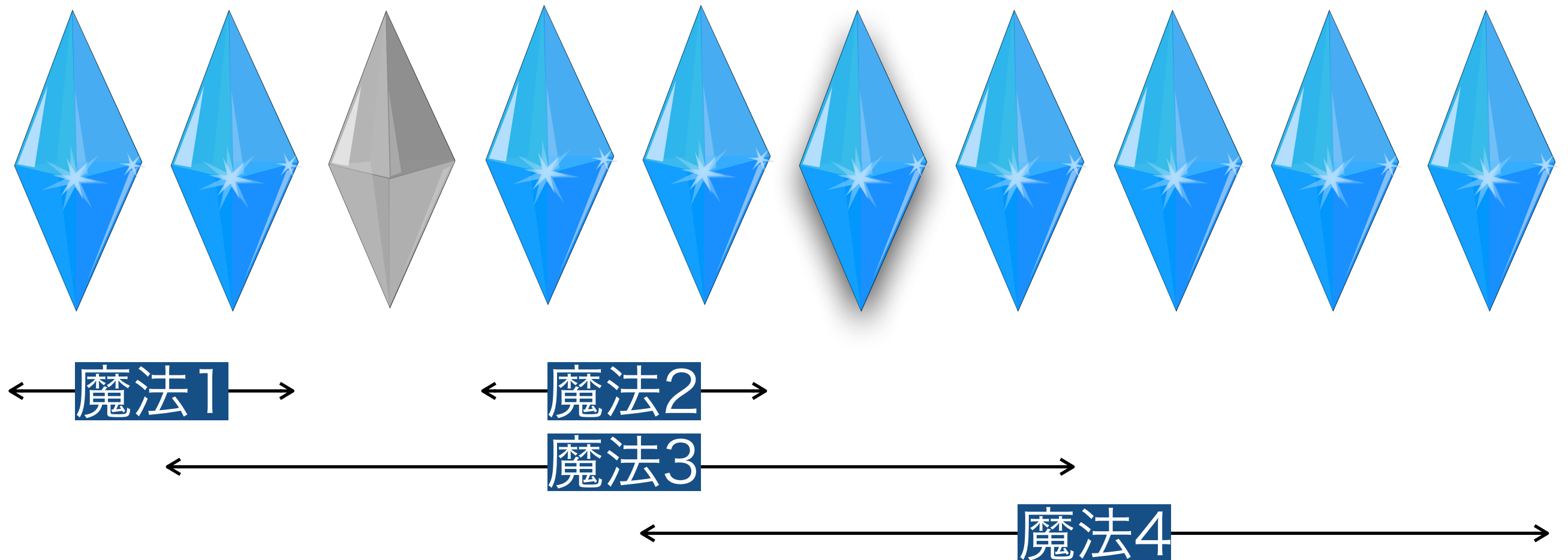
← 魔法2 →

← 魔法3 →

← 魔法4 →

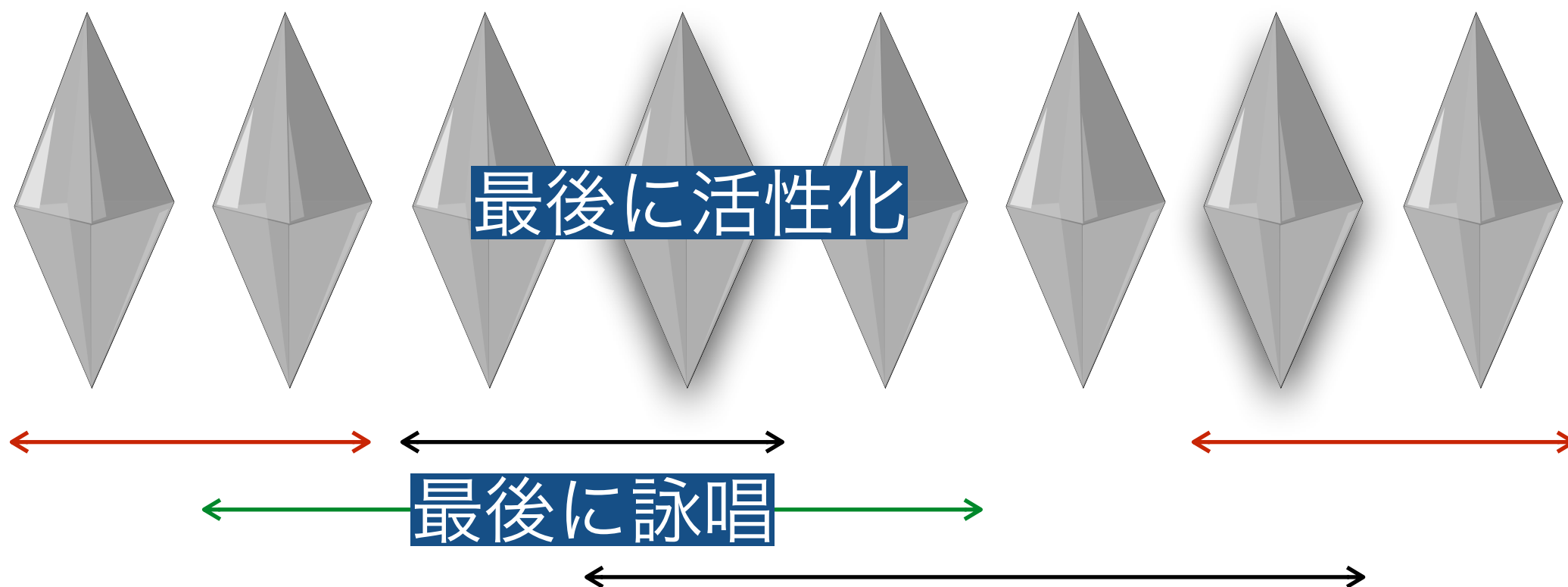
例

9個の魔法石を活性化できた



考察

- 最後に詠唱する魔法と
最後に活性化する特別な魔法石の組に対して
活性化できる魔法石の個数の最大値は簡単に求まる
 - 魔法石 $[1, r)$ を活性化できる魔法を魔法 $[1, r)$ と呼ぶ
 - 最後に活性化する特別な魔法石の位置を x とすると
 $r \leq x$ または $x < 1$ なる
 魔法 $[1, r)$ を先に全て詠唱し、
 最後に最後に詠唱する魔法を詠唱するのが最適



TLE解1

- ・ 最後に詠唱する魔法(m通り)と
最後に活性化する特別な魔法石(k通り)の組を
全探索した場合
 - imos法を使えば $O(km(n+m))$
 - これだと間に合わない

TLE解2

- ・ 最後に活性化する特別な魔法石 x_i を左から順に試せば、先に詠唱する魔法は x_i の左側で増加し、右側で減少する
- ・ 各魔法は高々1回しか増加・減少しないので、増加・減少する魔法の数は $O(m)$
- ・ x_i から x_{i+1} に移るときに増加・減少する魔法は高速に求められる(ならし $O(m)$)
- ・ 次のデータ構造を用いれば $O(km \log n + n \log n)$ で解が求まる

- ・ $[0, n)$ 上の区間の集合 S を管理する
- ・ この問題において以下のクエリを高速に処理できる
 - 追加クエリ: 区間 $[l, r)$ を S に追加する
 - 削除クエリ: 区間 $[l, r)$ を S から削除する
 - 検索クエリ: 区間 $[l, r)$ との共通部分の長さを求める

TLE解2

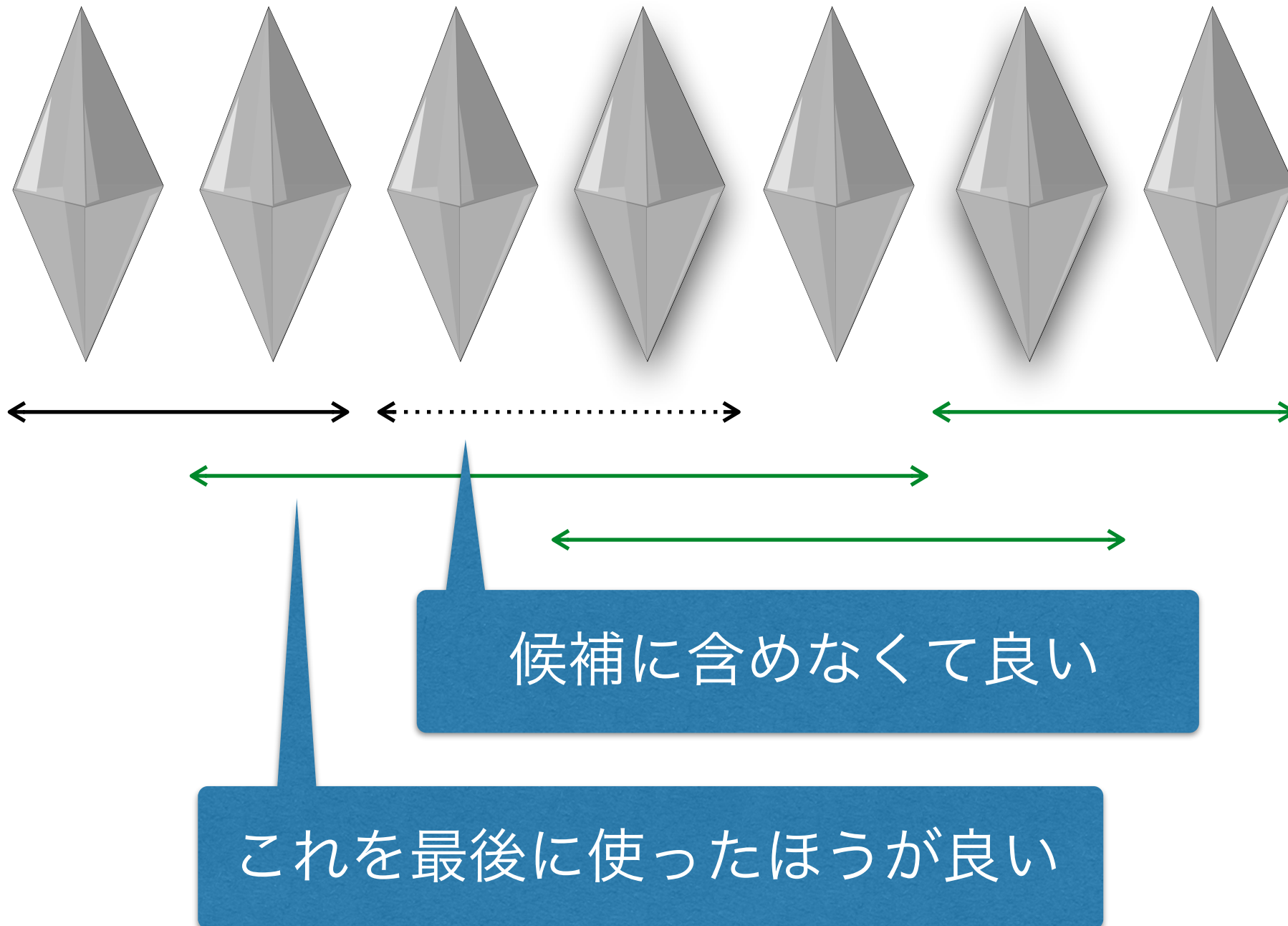
- ・ 次のデータ構造を用いれば、 $O(km \log n)$ で解が求まる
- ・ $[0, n)$ 上の区間の集合 S を管理する
- ・ この問題において以下のクエリを高速に処理できる
 - 追加クエリ: 区間 $[l, r)$ を S に追加する
 - 削除クエリ: 区間 $[l, r)$ を S から削除する
 - 検索クエリ: 区間 $[l, r)$ との共通部分の長さを求める
- ・ これは以下の2種類のセグメント木で実現できる
 - (StarrySkyTree)
区間add, 区間minが各 $O(\log n)$ で処理できる
以下sstと称する
 - 1点add, 区間sumが各 $O(\log n)$ で処理できる
以下segと称する

TLE解2

- ・ 追加クエリ
 - $sst[l, r)$ のminが0である場所の集合を X とする
 - $\forall x \in X$ について $sst[x, x+1)$, $seg[x]$ に+1
- ・ 削除クエリ
 - $sst[l, r)$ のminが1である場所の集合を X とする
 - $\forall x \in X$ について $sst[x, x+1)$, $seg[x]$ に-1
- ・ 検索クエリ
 - $seg[l, r)$ のsum
- ・ 今回の場合, sst の各点において
0から1になる, 1から0になる回数は高々1回ずつなので
m回の追加・削除を $O(m \log n + n \log n)$ で処理できる
- ・ よって全体として $O(km \log n + n \log n)$ で処理できる
 - これでも間に合わない

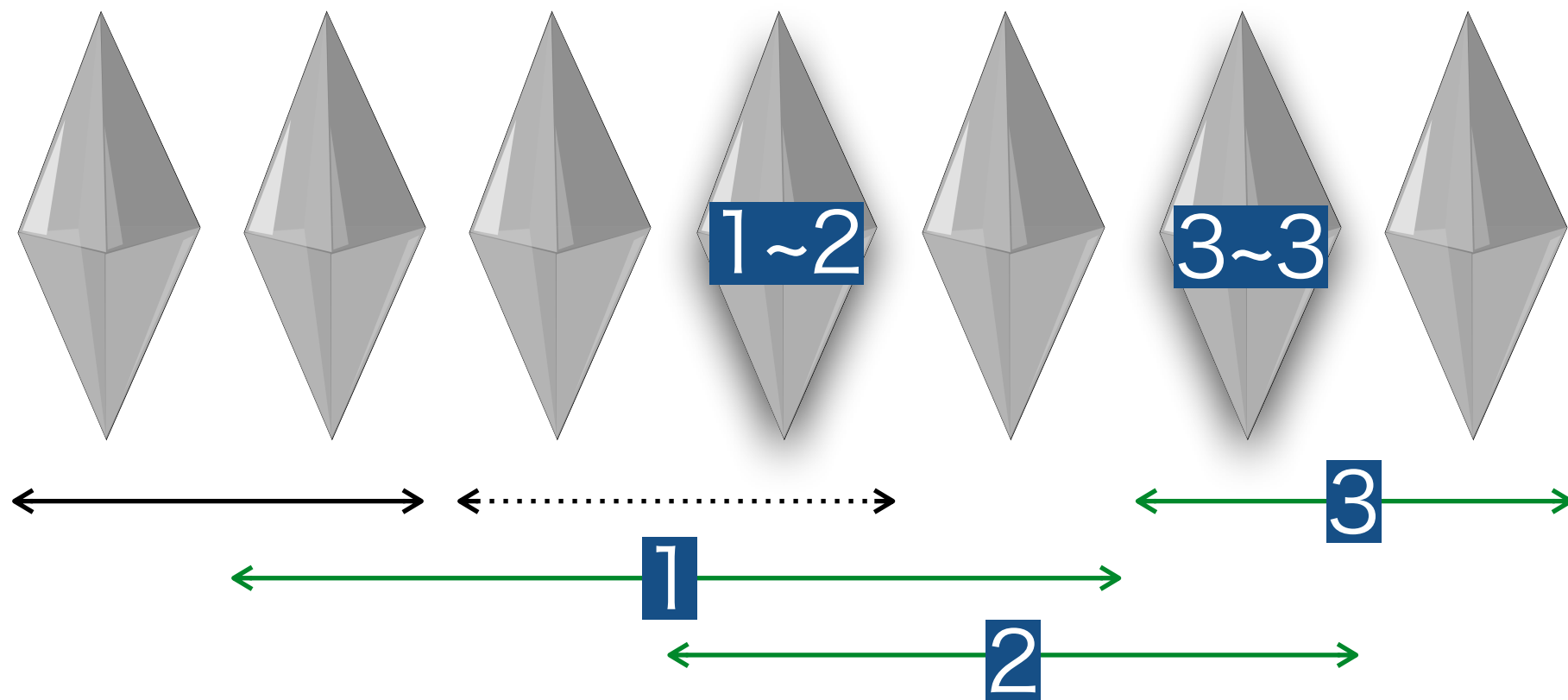
考察

- 最後に詠唱する魔法の候補として
他の魔法に包含されている魔法は考えなくて良い



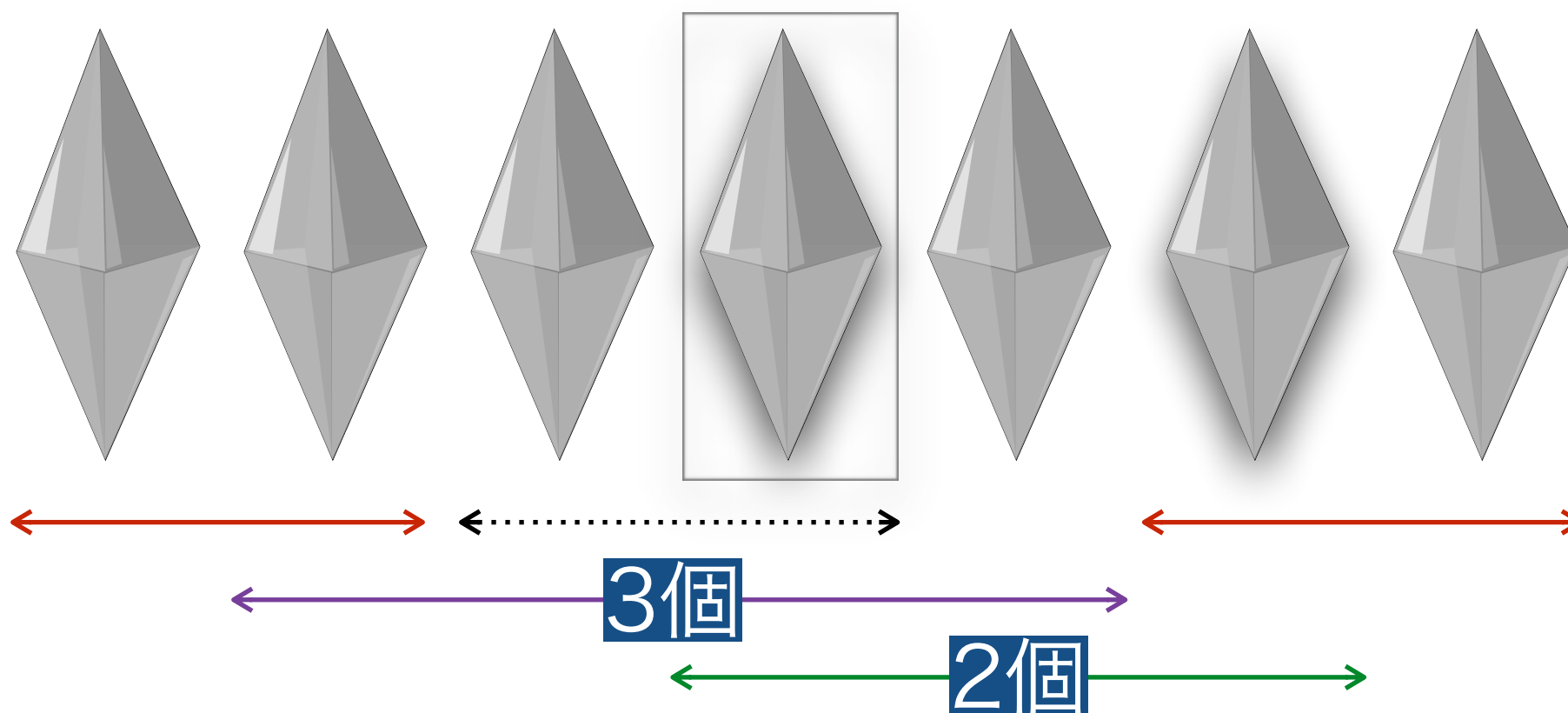
考察

- 候補魔法を左から順にindex付けすれば
最後に活性化する特別な魔法石に対する
最後に詠唱する魔法の候補のindexは区間になる



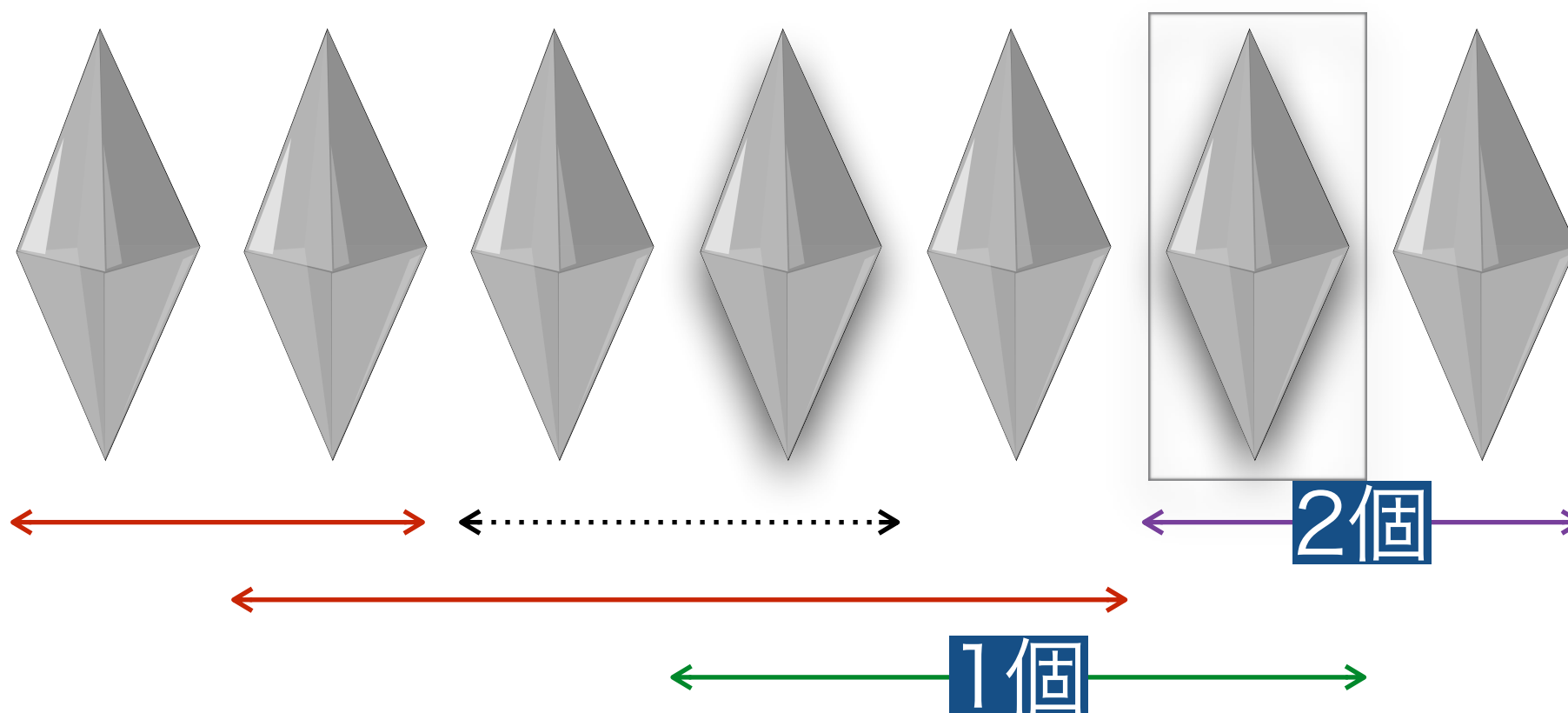
考察

- ・ 先に詠唱できる魔法は先に詠唱しているものとする
- ・ 最後に詠唱する魔法の候補の中から
新たに活性化できる魔法石の数が最も多い魔法を
最後に詠唱する魔法として選べば良い



考察

- ・ 先に詠唱できる魔法は先に詠唱しているものとする
- ・ 最後に詠唱する魔法の候補の中から
新たに活性化できる魔法石の数が最も多い魔法を
最後に詠唱する魔法として選べば良い



想定解

- ・ 初期値
 - 各魔法 $[1, r)$ の良さを $r - 1$ とする
- ・ i 番目の魔法石が新たに事前に活性化されるようになったとき
 - i を含む候補魔法の良さを -1
 - 活性化済魔法石数を $+1$
- ・ i 番目の魔法石が事前に活性化されなくなったとき
 - i を含む候補魔法の良さを $+1$
 - 活性化済魔法石数を -1
- ・ 魔法石 x を最後に活性化させるときの最適解
 - (事前に活性化した魔法石数)
 - + (x を含む候補魔法中の良さの最大値)

想定解

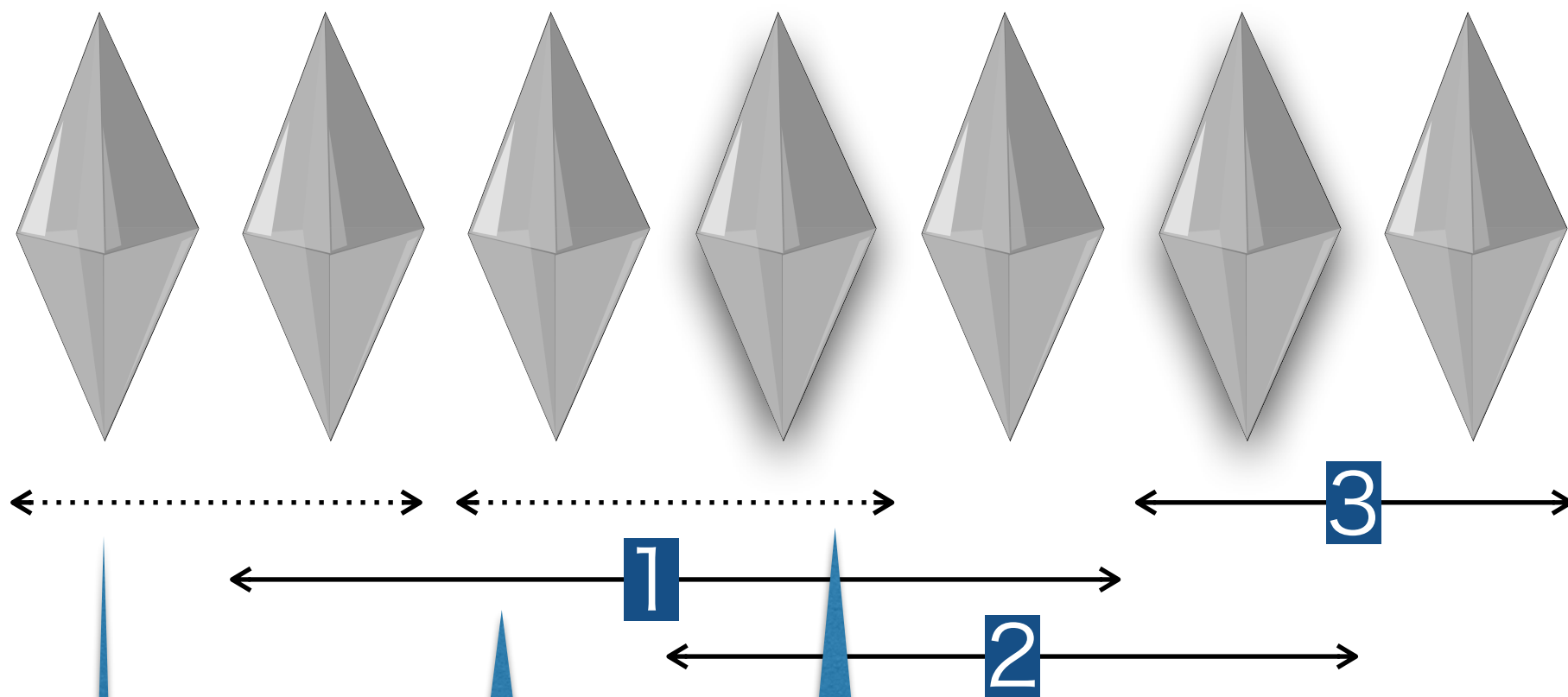
- ・ TLE解2のsegを、候補魔法の良さの列を管理するsstに変えればよい
- ・ i を含む候補魔法は、候補魔法を事前にソートしておけば、二分探索で求まる
- ・ 候補魔法数は $O(\min(m, n))$ なので、全体として $O(k \log(\min(m, n)) + m \log n + n \log(\min(m, n)))$

sst₂から良さを加減する処理

sst₁に魔法を追加・削除する処理

最後に活性化させる魔法石ごとの
sst₂に対する検索クエリ処理

想定解の動作



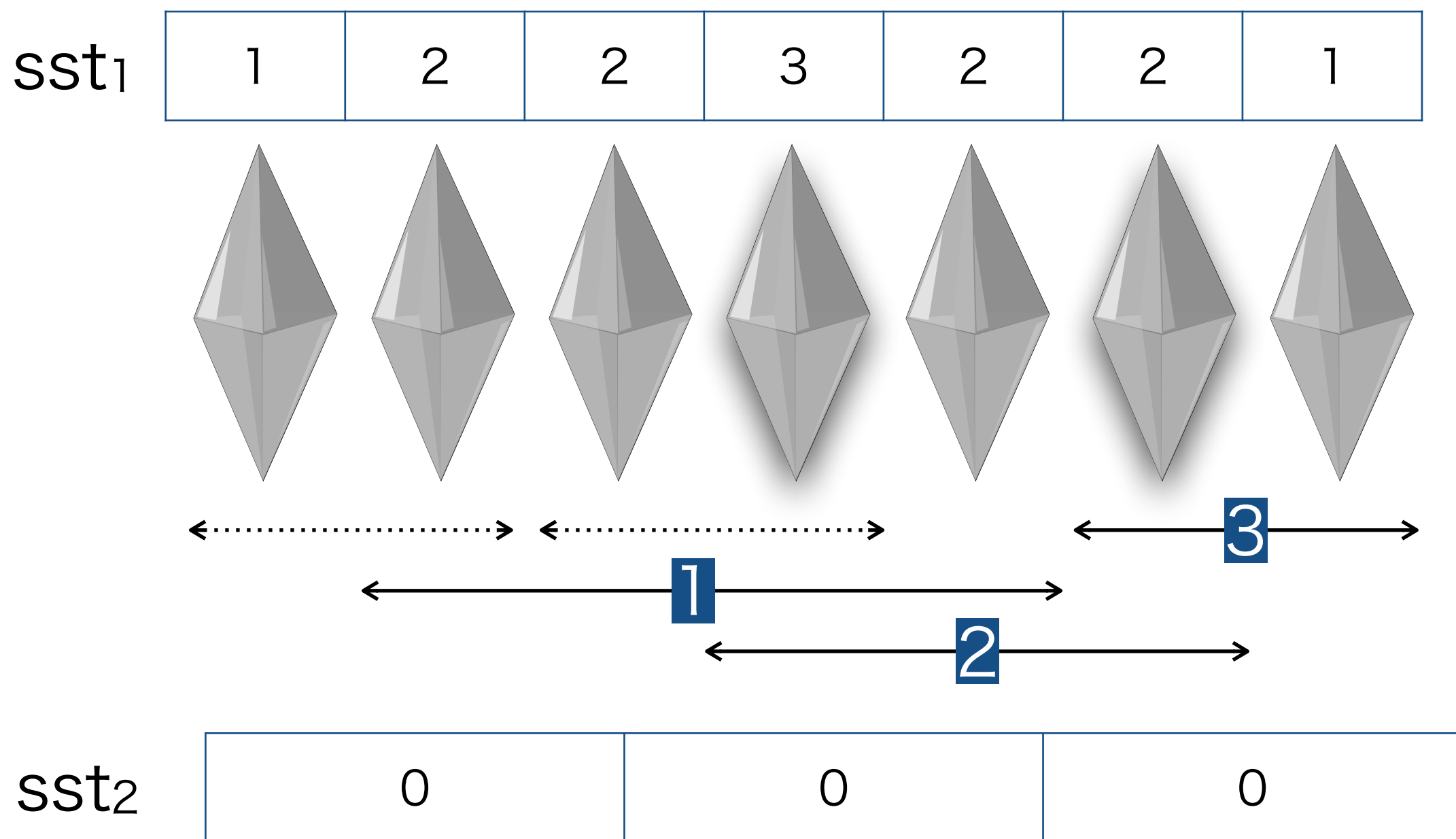
候補区間

これは1に包含されているので
候補区間でない

これも候補区間に追加しても良い

想定解の動作

全区間を追加した状態が初期状態



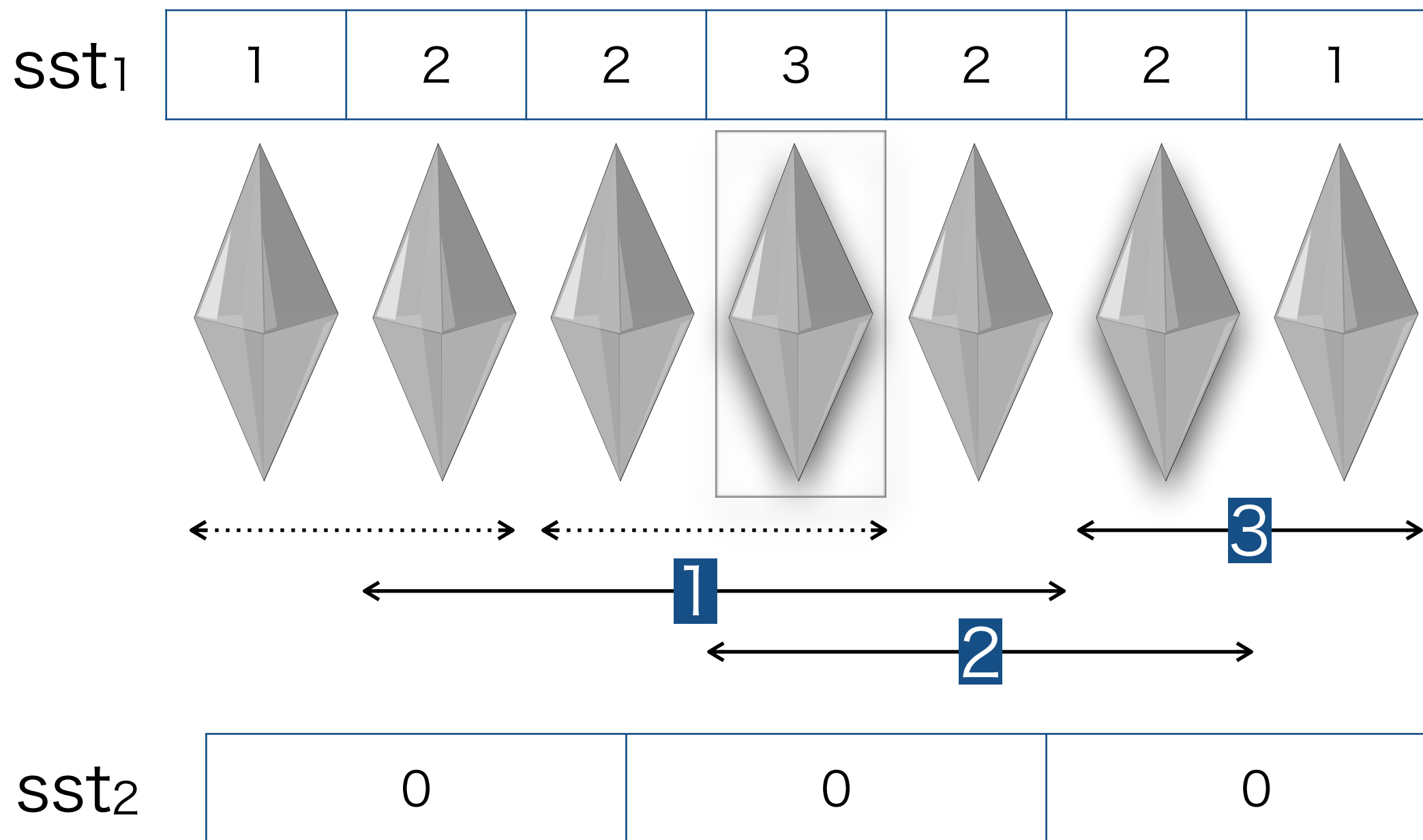
活性化済魔法石数: 7

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の削除: [1, 2], [2, 5], [3, 4], [4, 6]

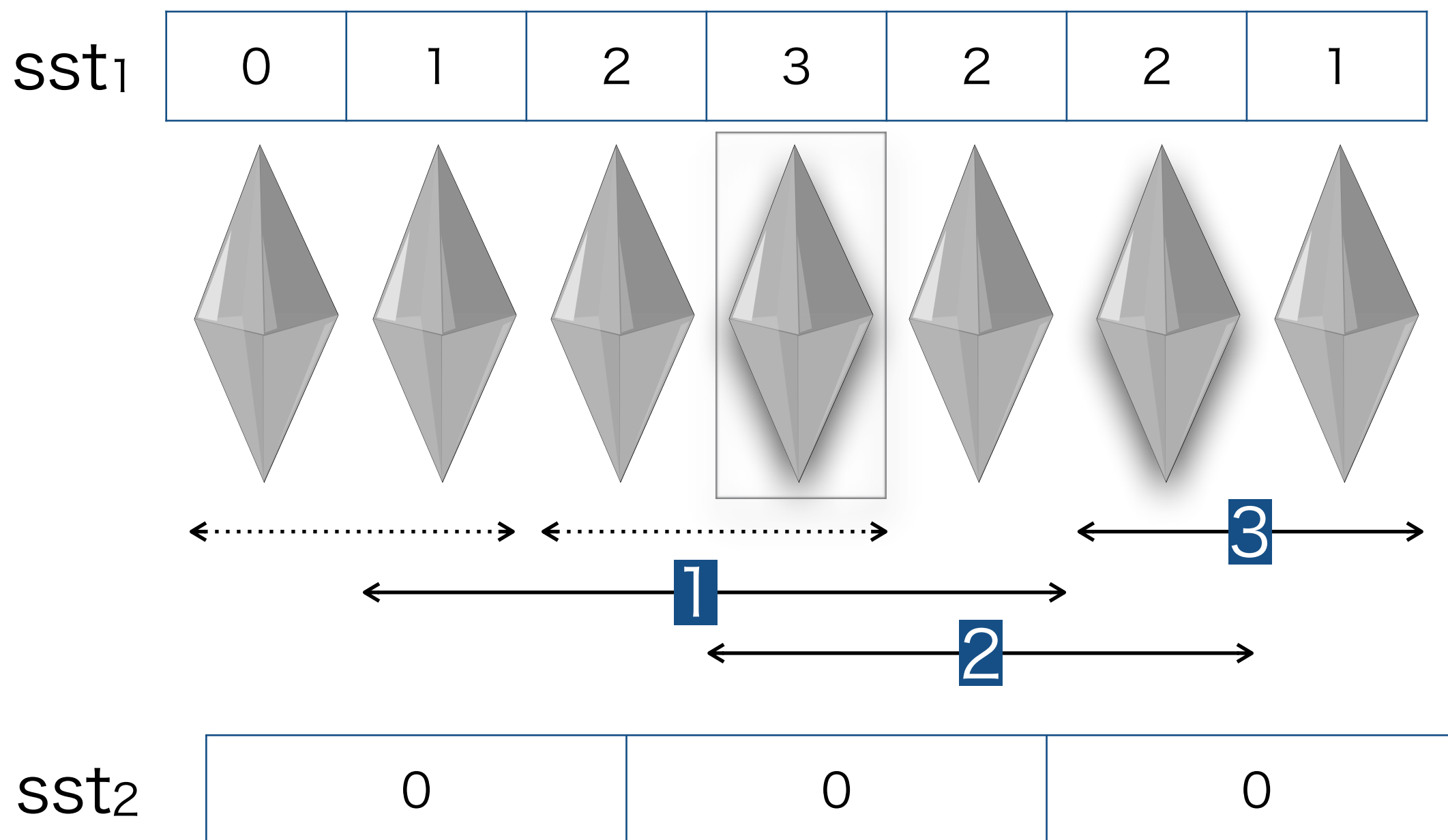
先に詠唱する魔法の追加: [1, 2]



活性化済魔法石数: 7

想定解の動作

魔法石4を最後に活性化させようとする
先に詠唱する魔法の削除: [1, 2]



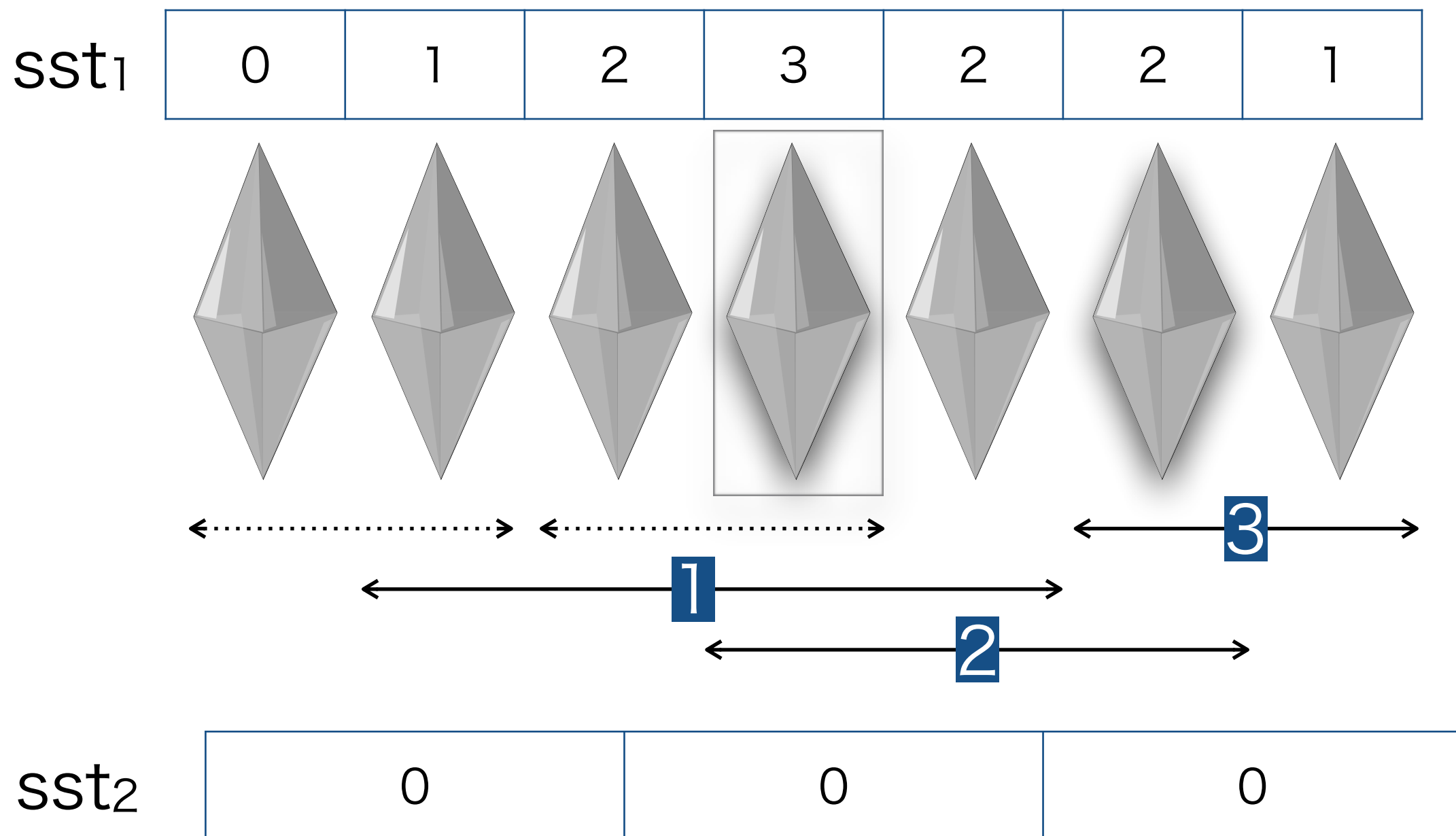
活性化済魔法石数: 7

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の削除: [1, 2]

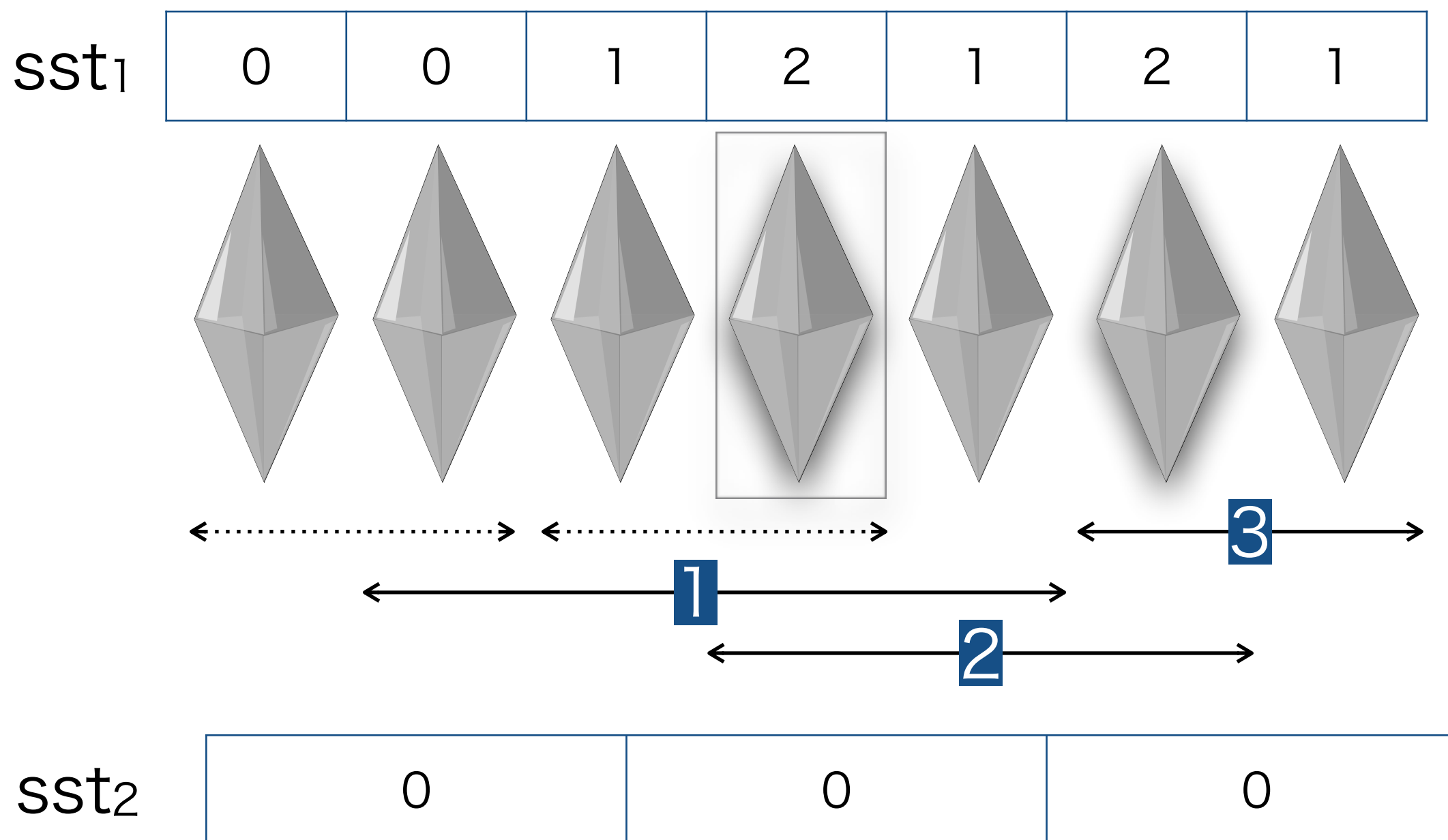
活性化済魔法石数を-1



活性化済魔法石数: 6

想定解の動作

魔法石4を最後に活性化させようとする
先に詠唱する魔法の削除: [2, 5]



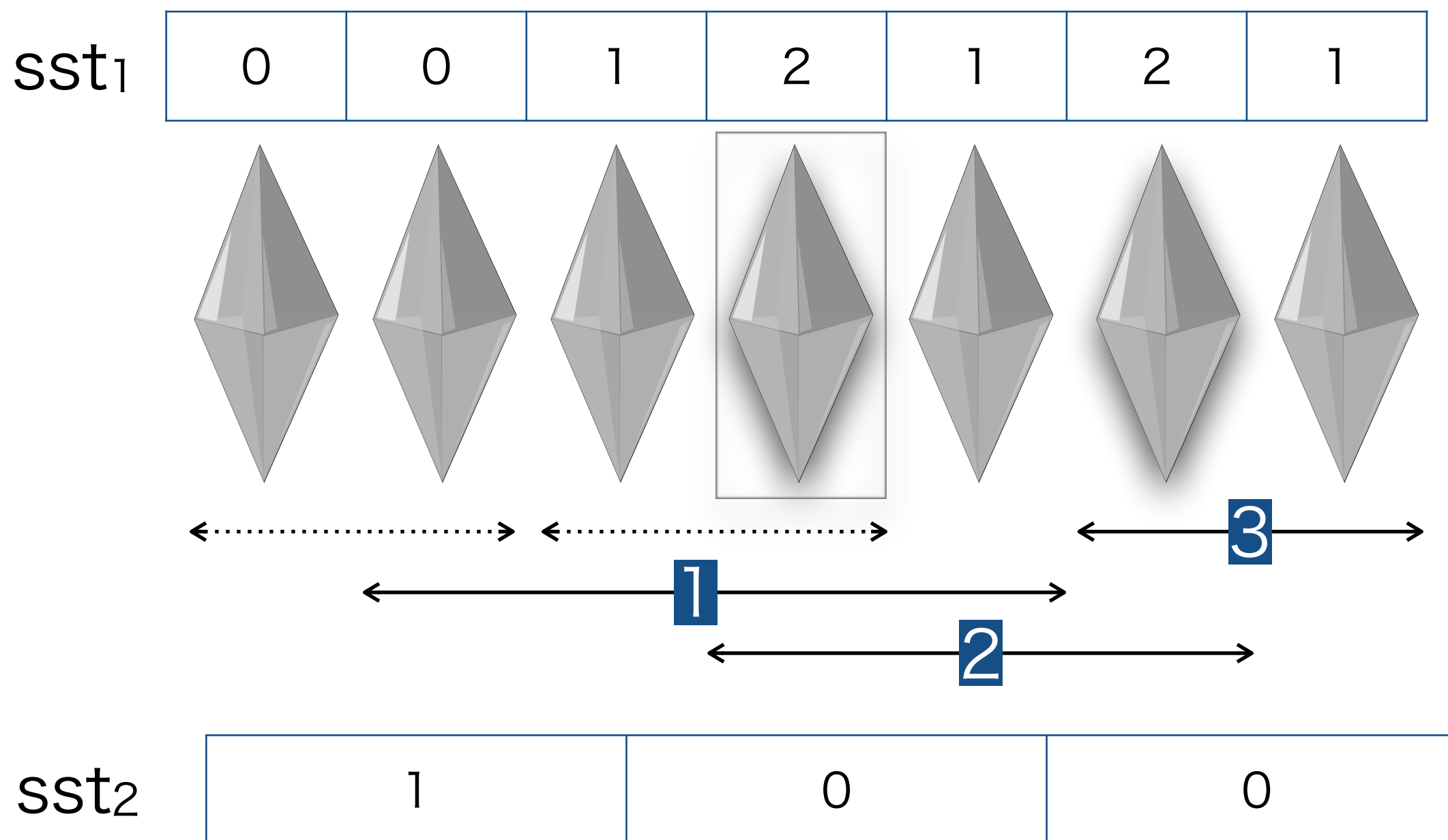
活性化済魔法石数: 6

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の削除: [2, 5]

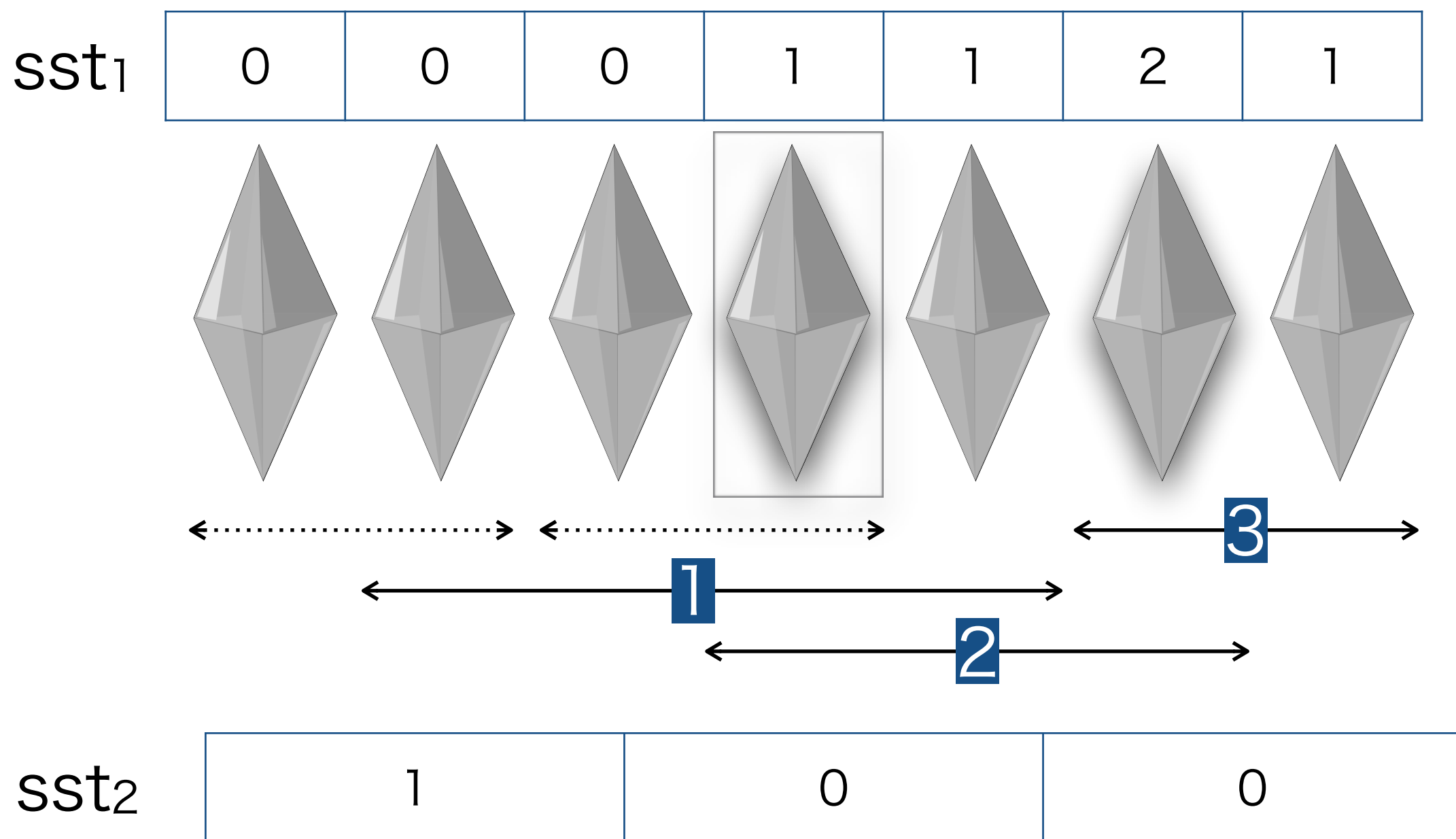
魔法石2を含む候補魔法の良さ+1, 活性化済...を-1



活性化済魔法石数: 5

想定解の動作

魔法石4を最後に活性化させようとする
先に詠唱する魔法の削除: [3, 4]



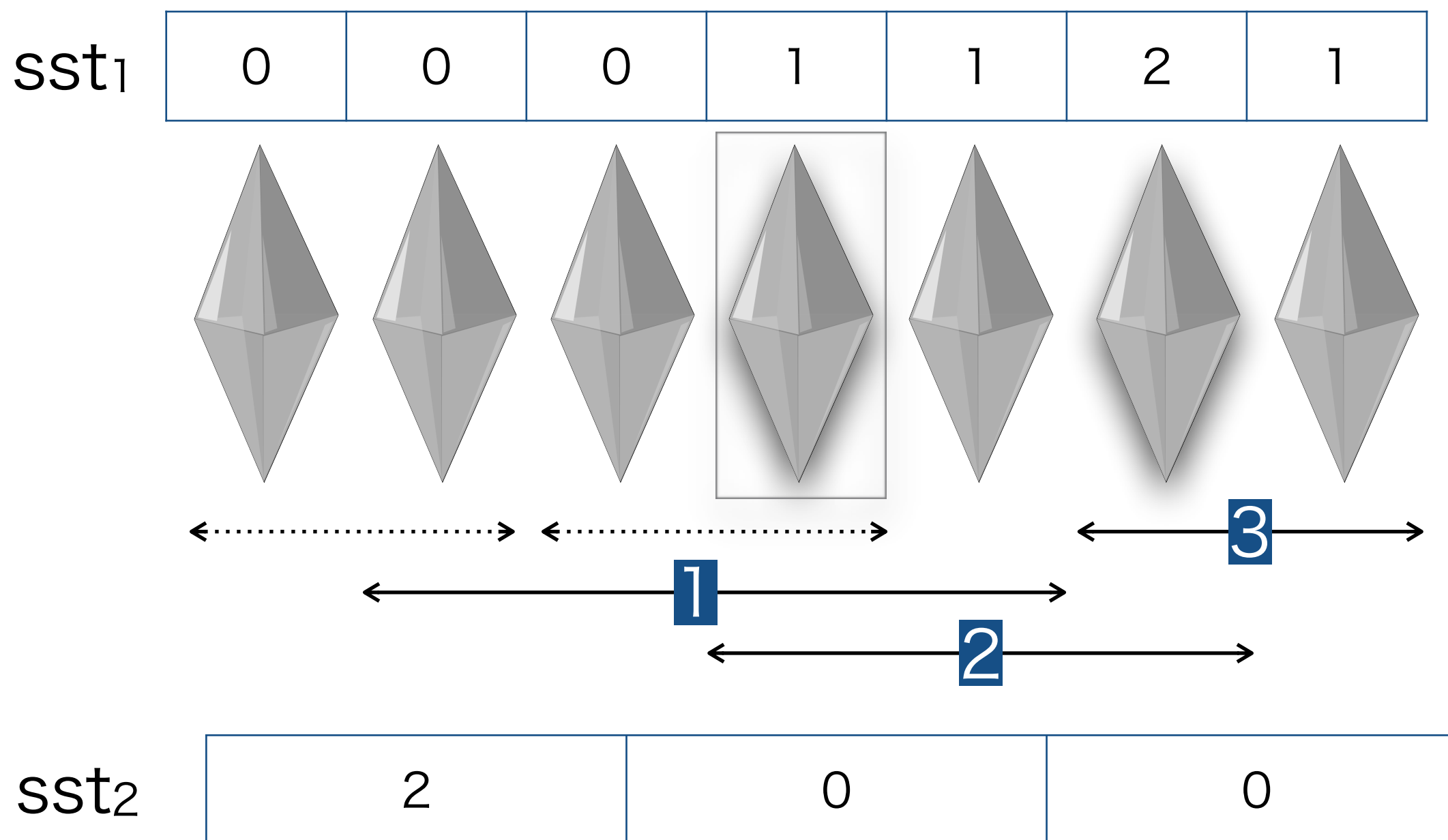
活性化済魔法石数: 5

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の削除: [3, 4]

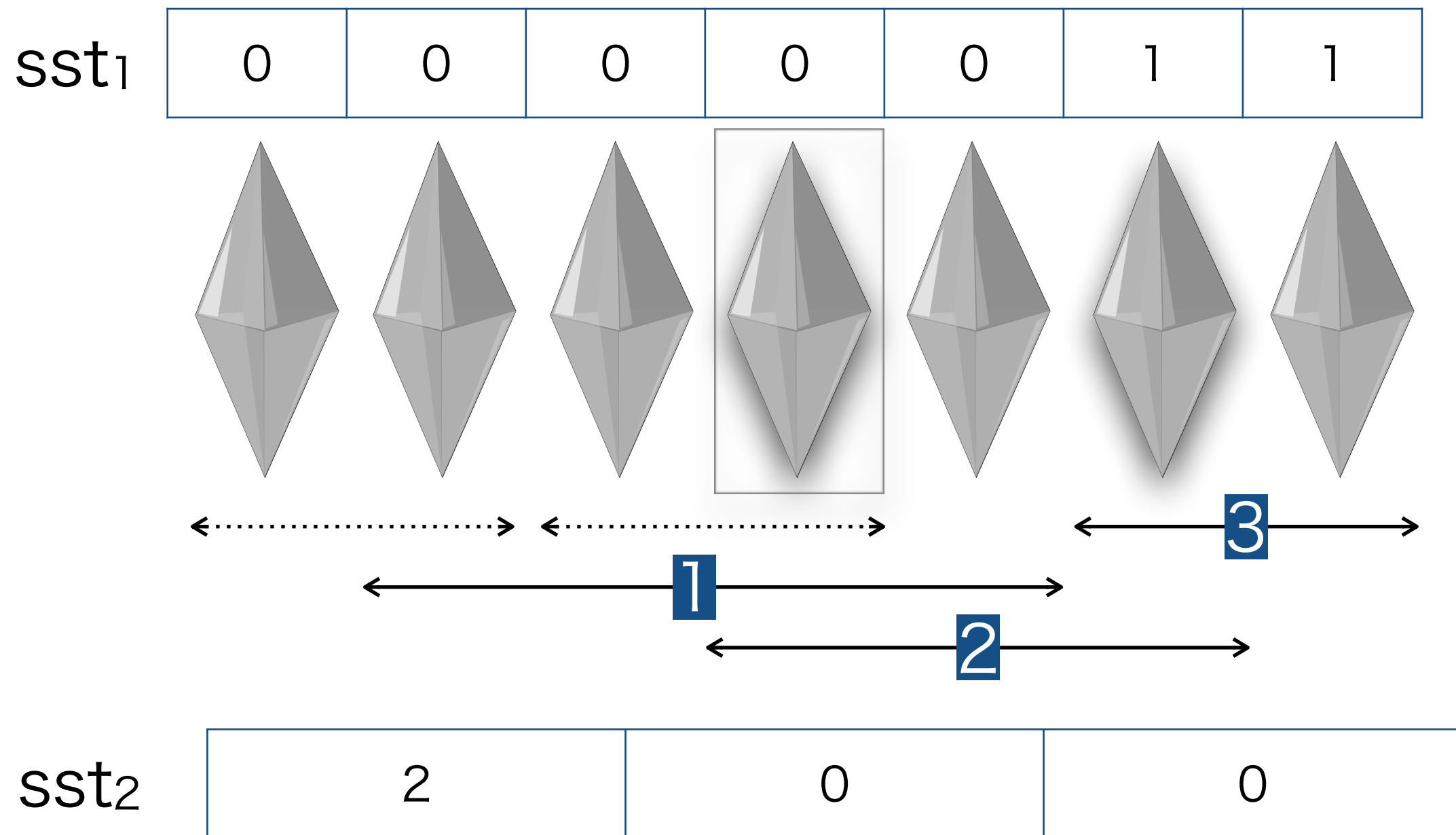
魔法石3を含む候補魔法の良さ+1, 活性化済...を-1



活性化済魔法石数: 4

想定解の動作

魔法石4を最後に活性化させようとする
先に詠唱する魔法の削除: [4, 6]



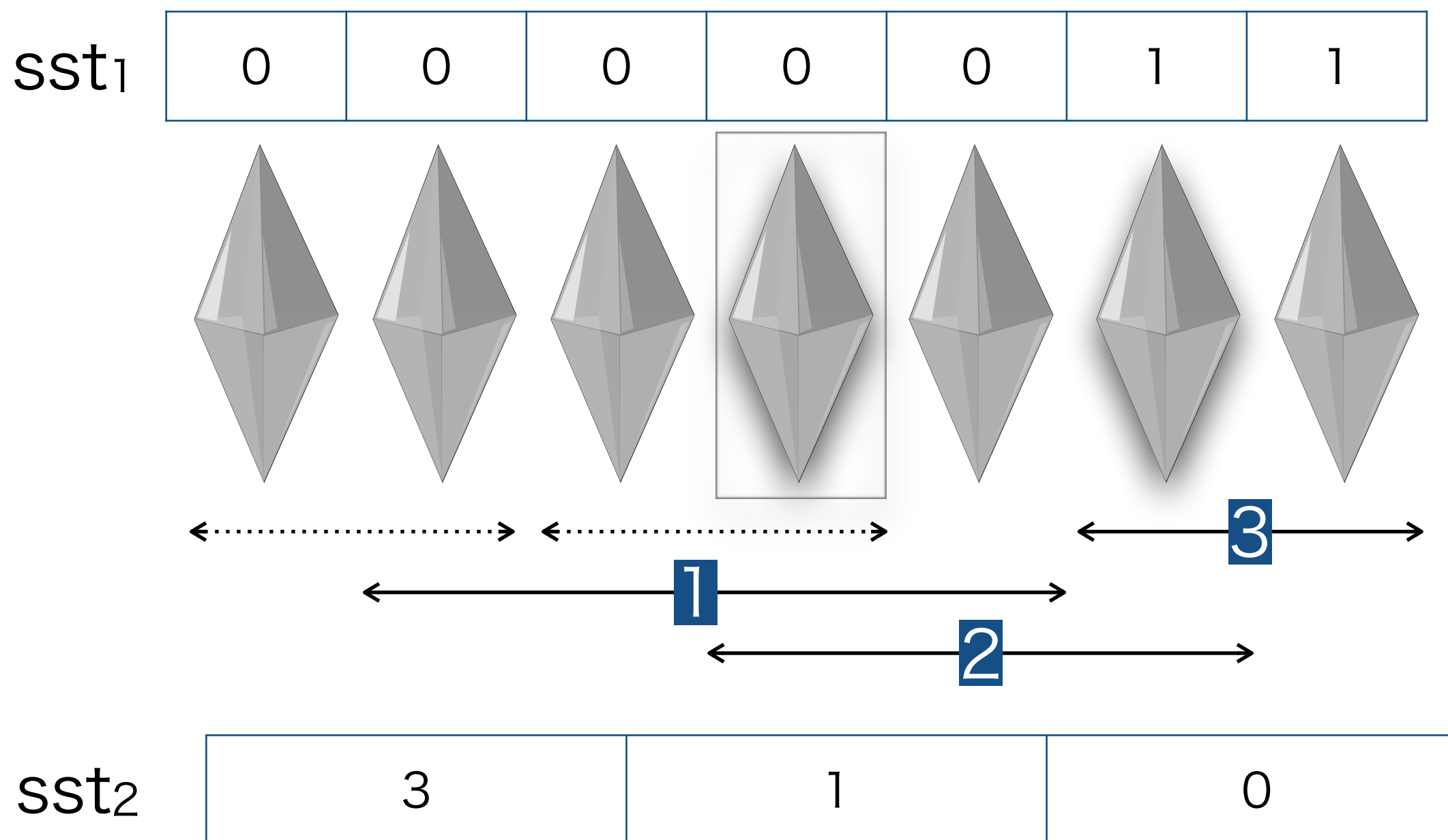
活性化済魔法石数: 4

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の削除: [4, 6]

魔法石4を含む候補魔法の良さ+1, 活性化済…を-1



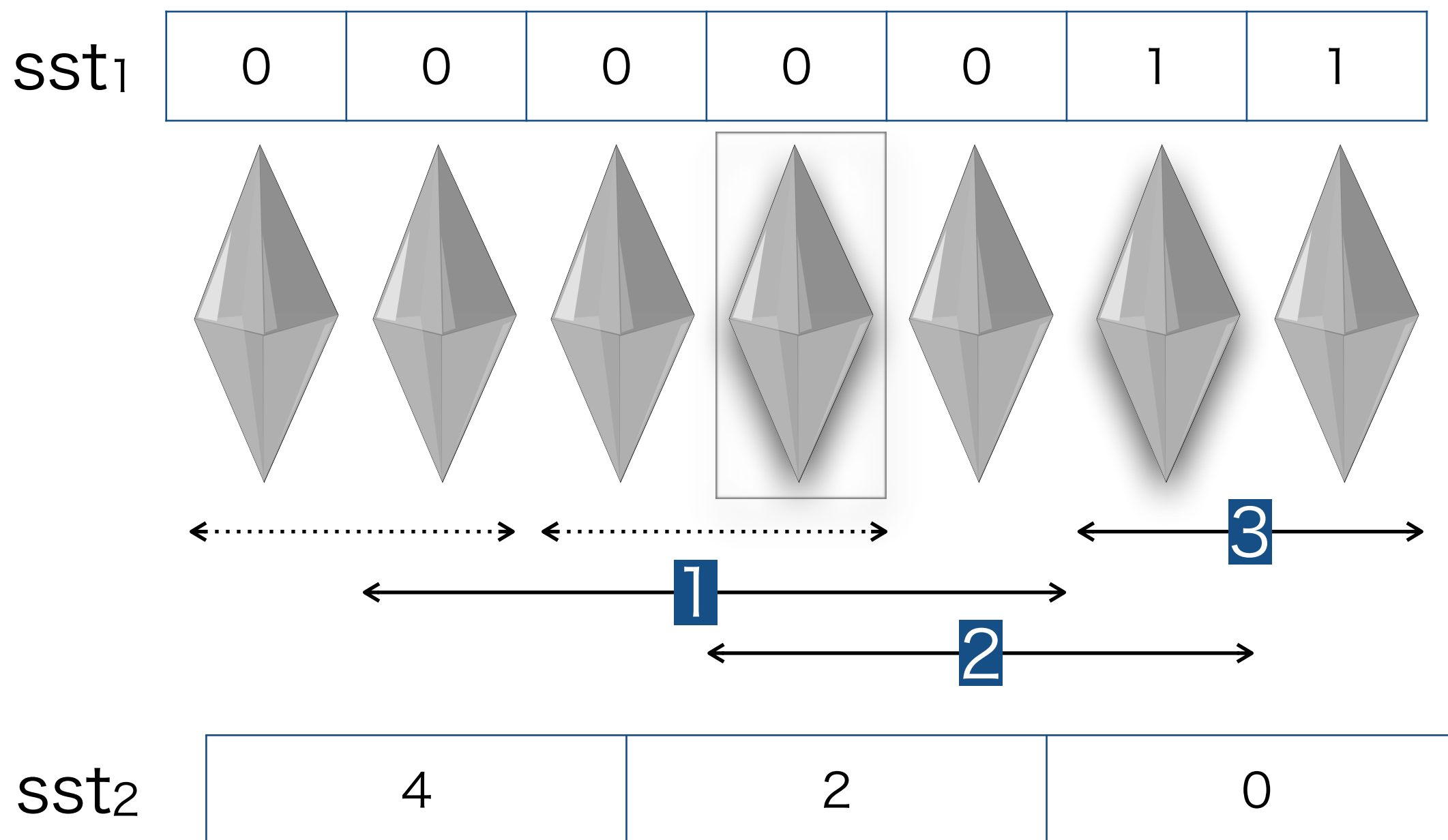
活性化済魔法石数: 3

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の削除: [4, 6]

魔法石5を含む候補魔法の良さ+1, 活性化済...を-1

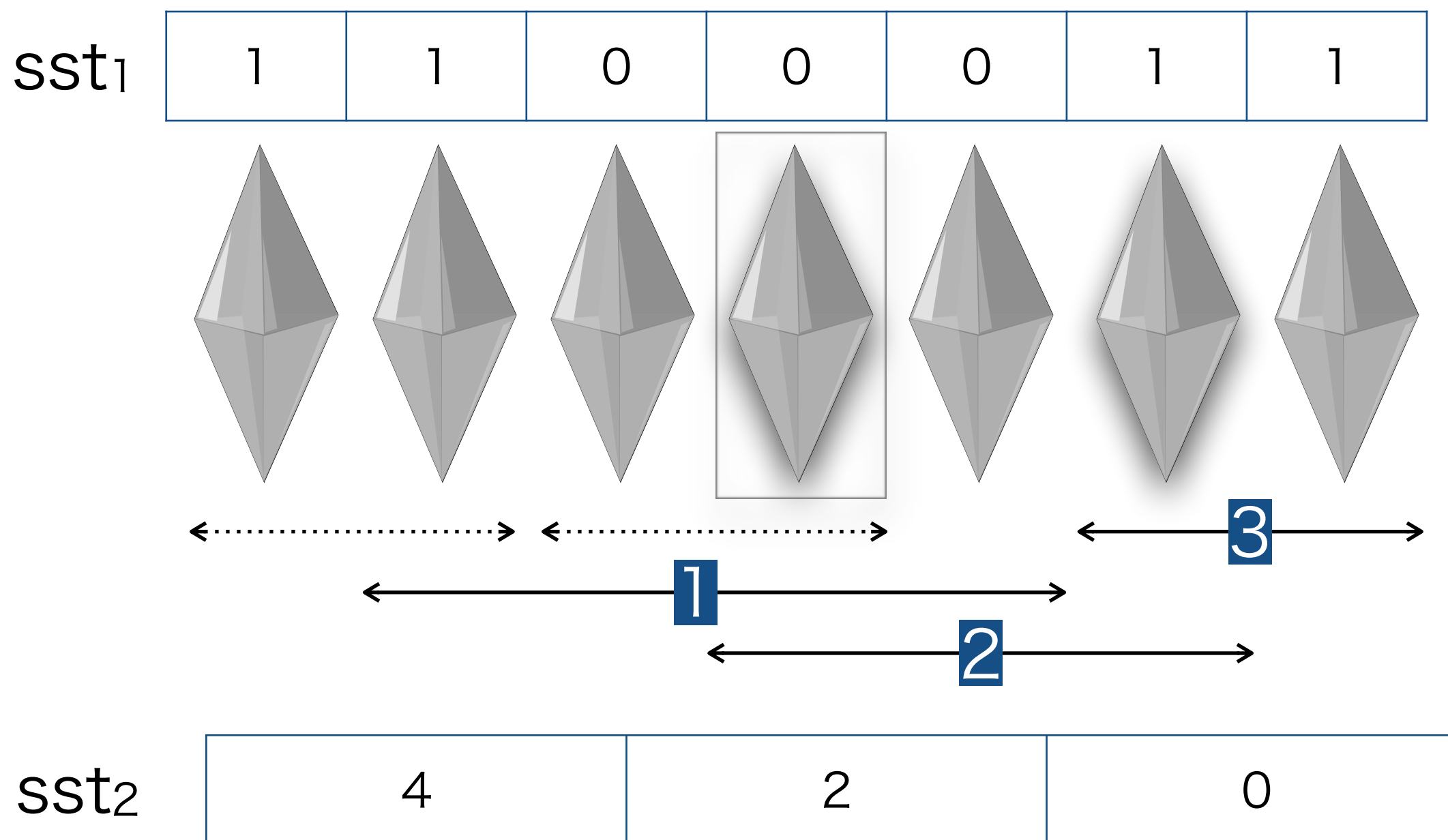


活性化済魔法石数: 2

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の追加: [1, 2]



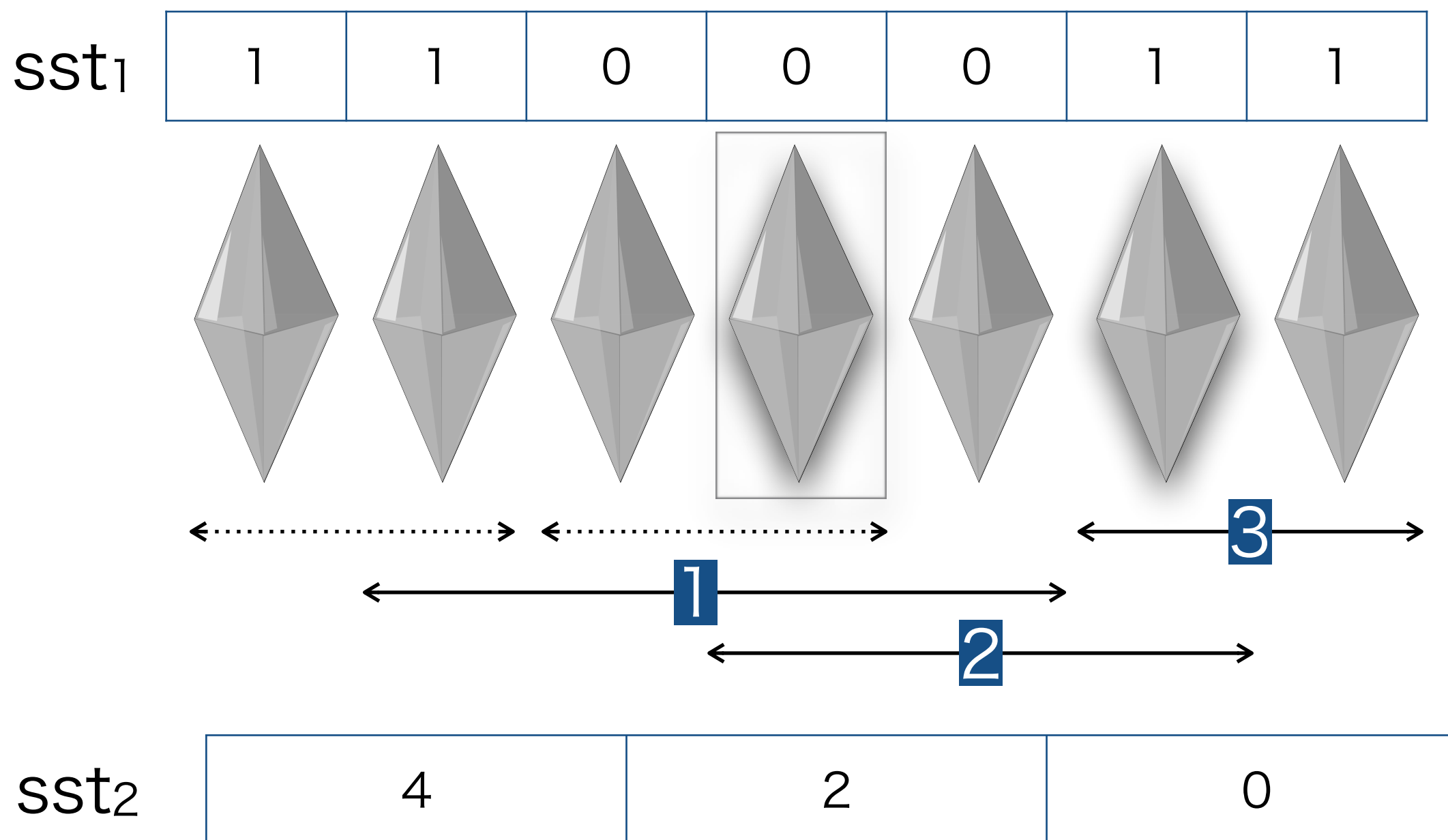
活性化済魔法石数: 2

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の追加: [1, 2]

魔法石1を含む候補魔法の良さ-1, 活性化済…を+1



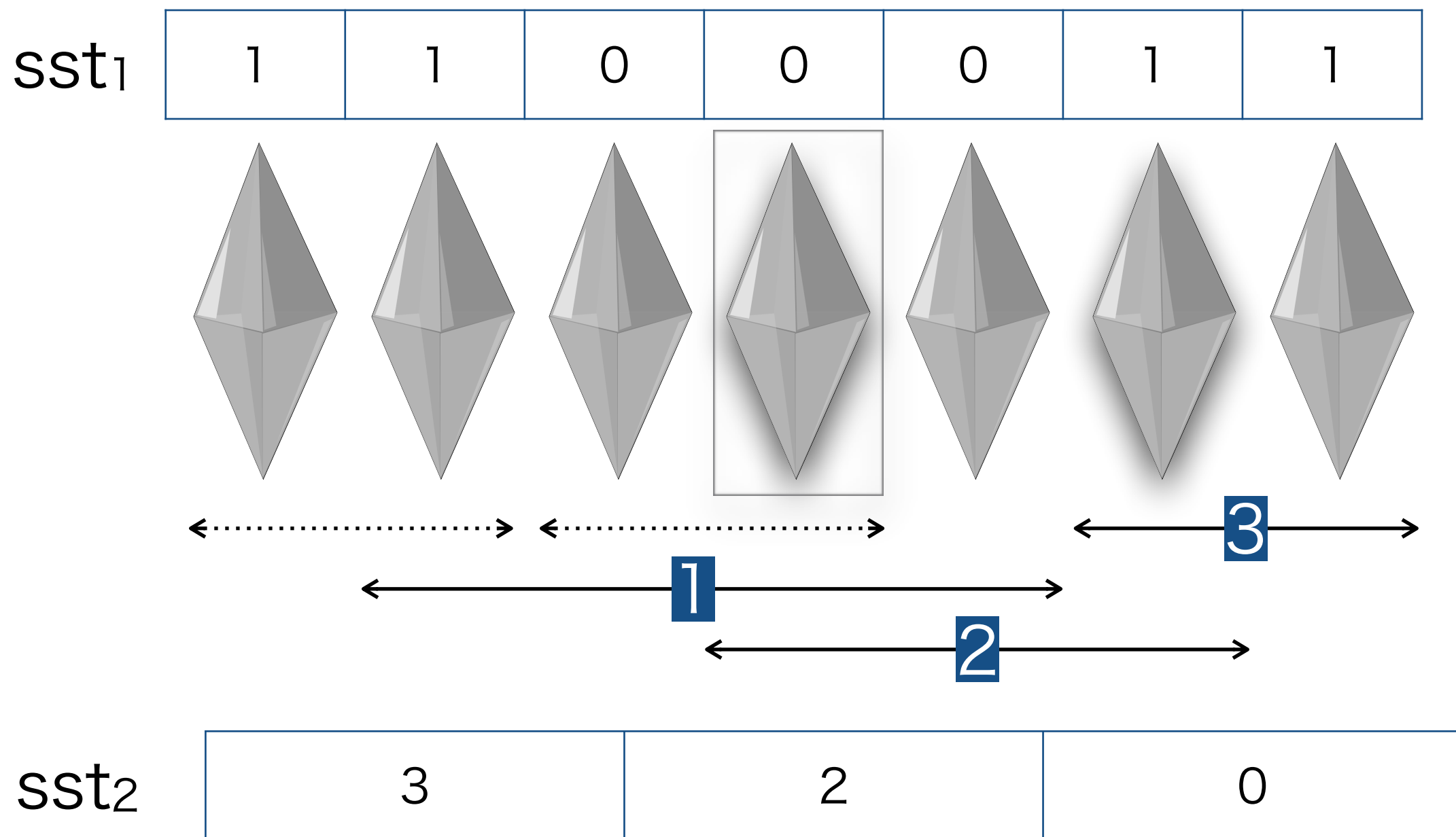
活性化済魔法石数: 3

想定解の動作

魔法石4を最後に活性化させようとする

先に詠唱する魔法の追加: [1, 2]

魔法石2を含む候補魔法の良さ-1, 活性化済…を+1



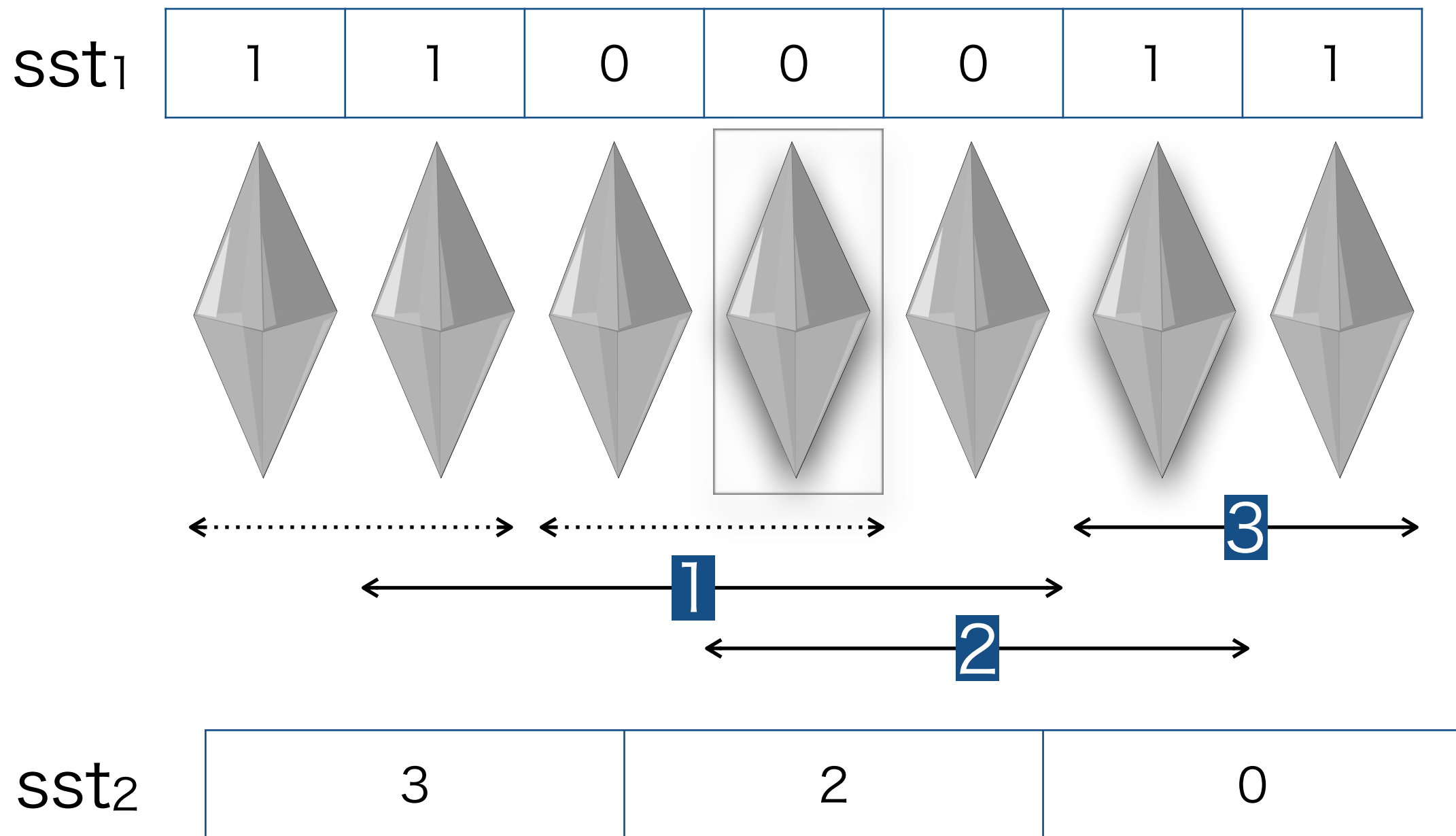
活性化済魔法石数: 4

想定解の動作

魔法石4を最後に活性化させようとする

組となり最後に詠唱する魔法の候補は1または2 ([1, 2])

答え = $\max(\text{答え}, \text{sst}_2[1, 2] \text{の} \max(3) + \text{活性化済魔法石数}(4))$



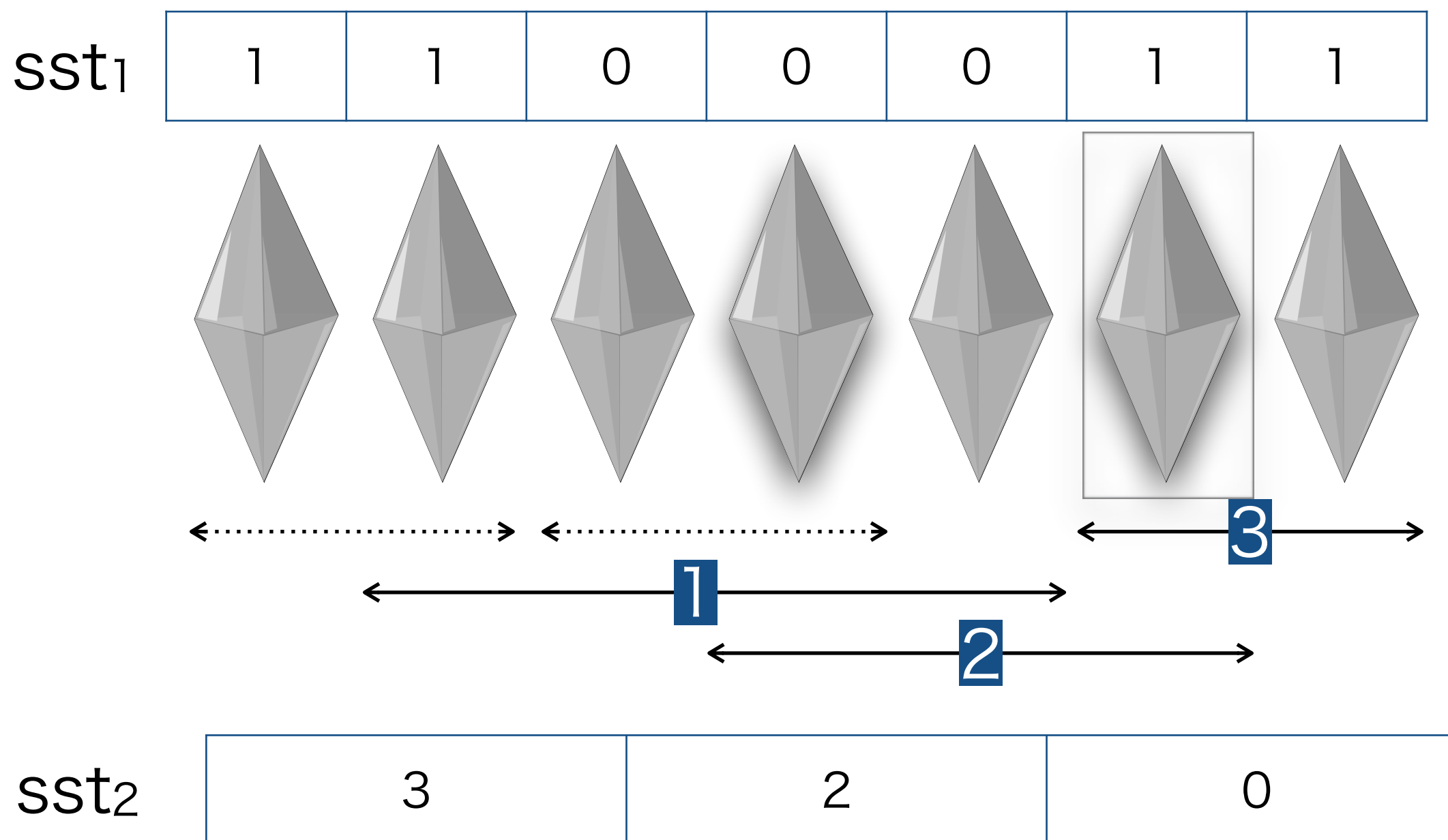
活性化済魔法石数: 4

想定解の動作

魔法石6を最後に活性化させようとする

先に詠唱する魔法の削除: [6, 7]

先に詠唱する魔法の追加: [3, 4], [2, 5]

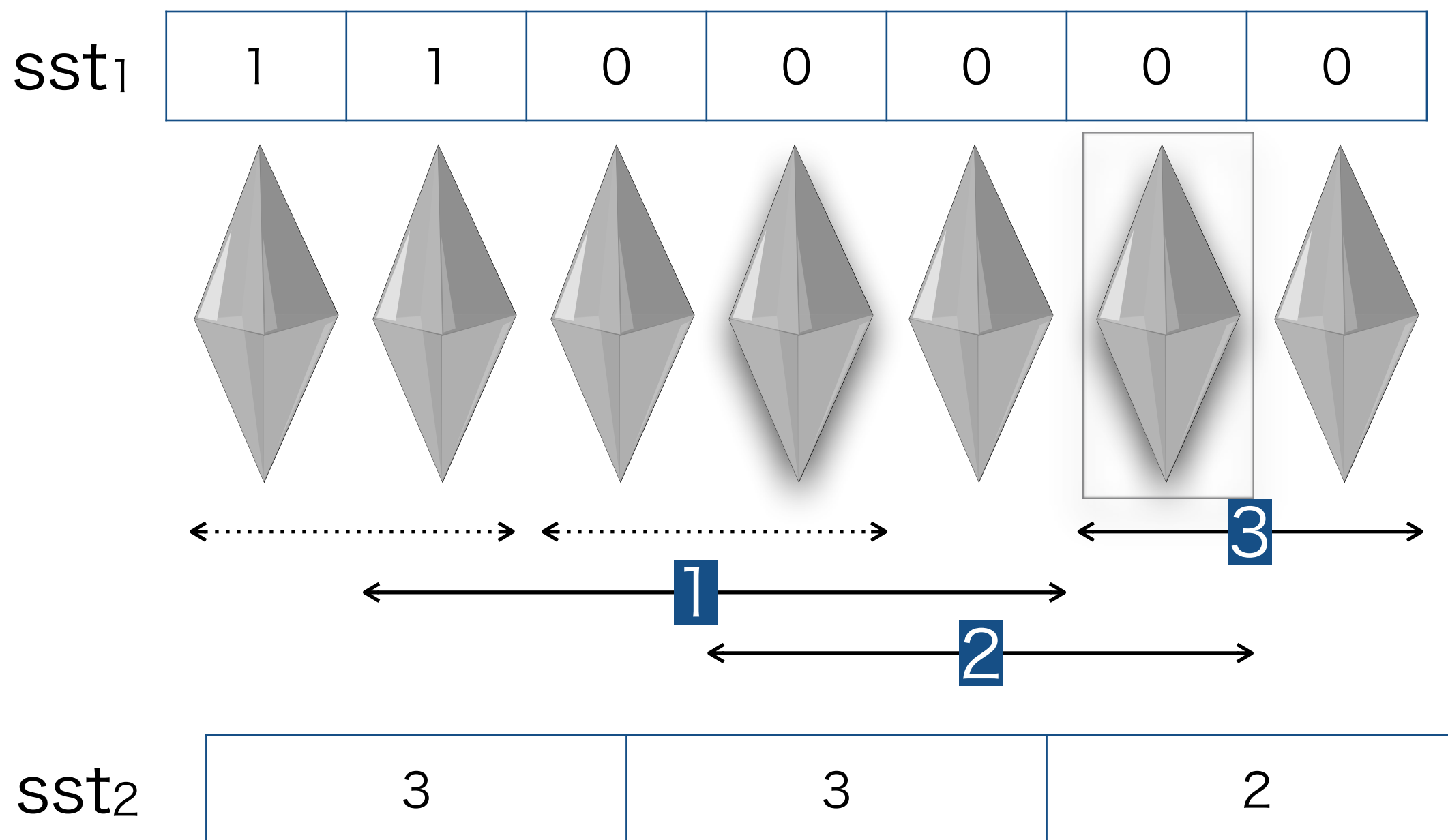


活性化済魔法石数: 4

想定解の動作

魔法石6を最後に活性化させようとする
同様の処理を行う

[6, 7] 削除

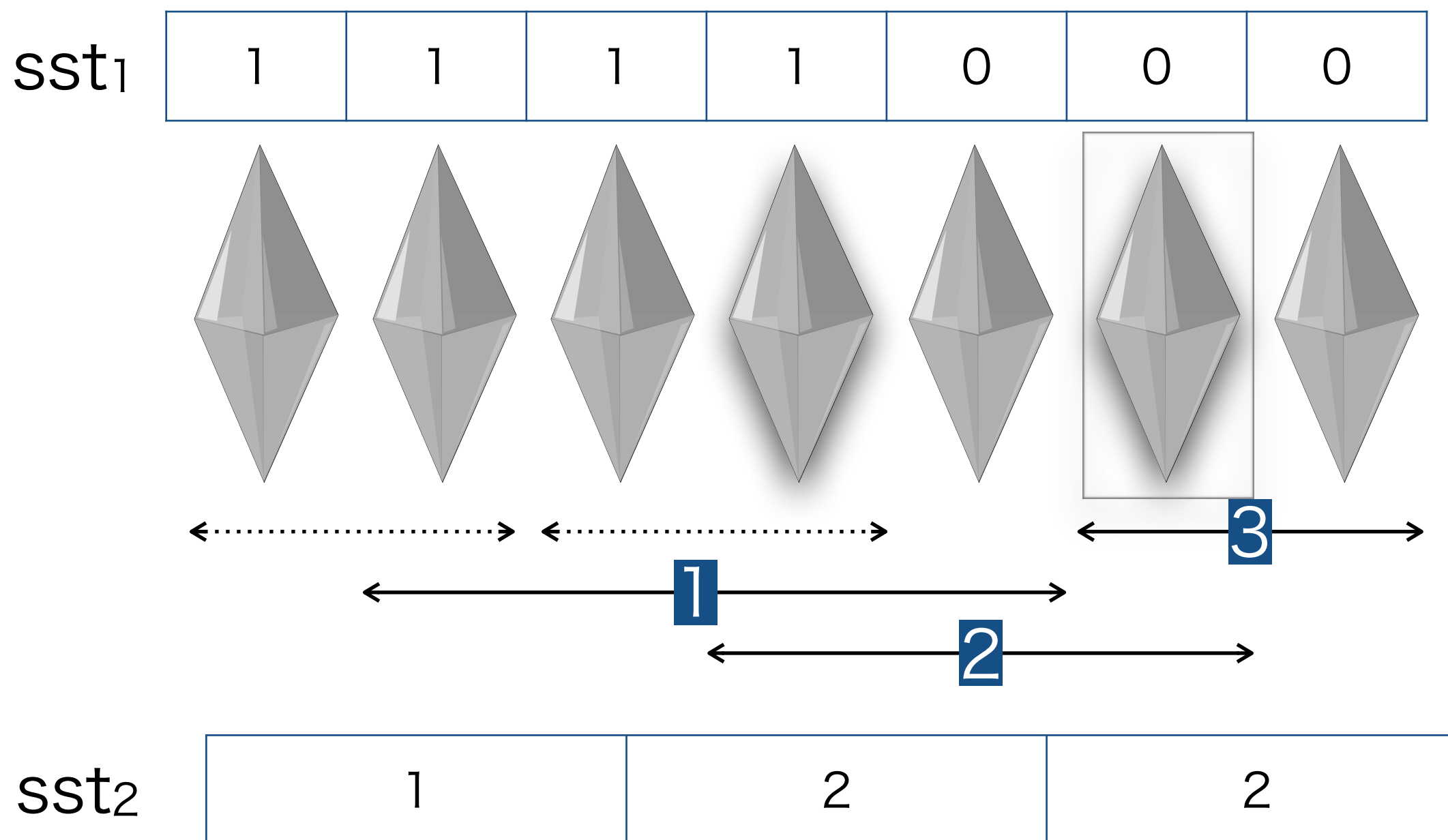


活性化済魔法石数: 2

想定解の動作

魔法石6を最後に活性化させようとする
同様の処理を行う

[3, 4] 追加

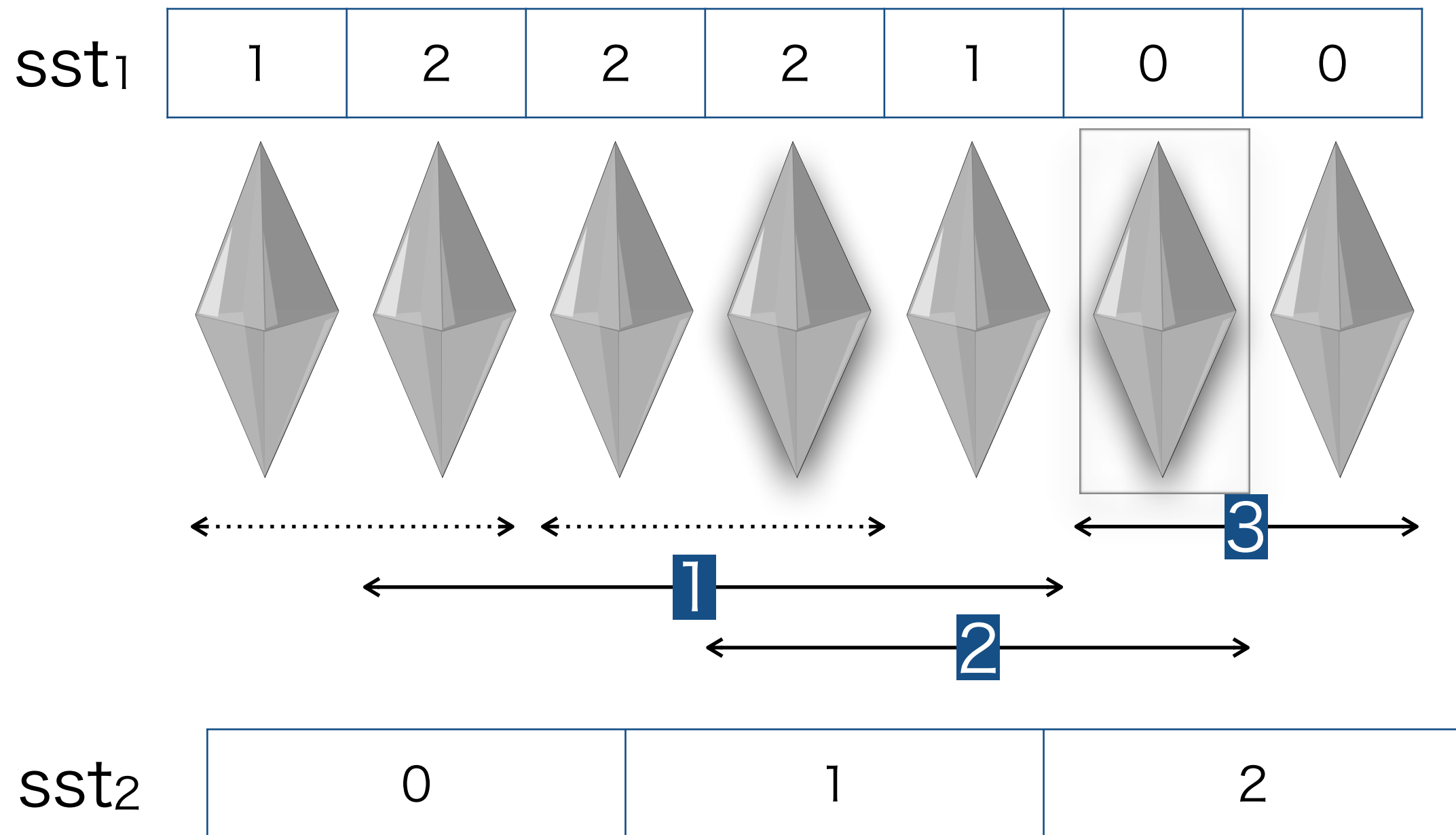


活性化済魔法石数: 4

想定解の動作

魔法石6を最後に活性化させようとする
同様の処理を行う

[2, 5] 追加



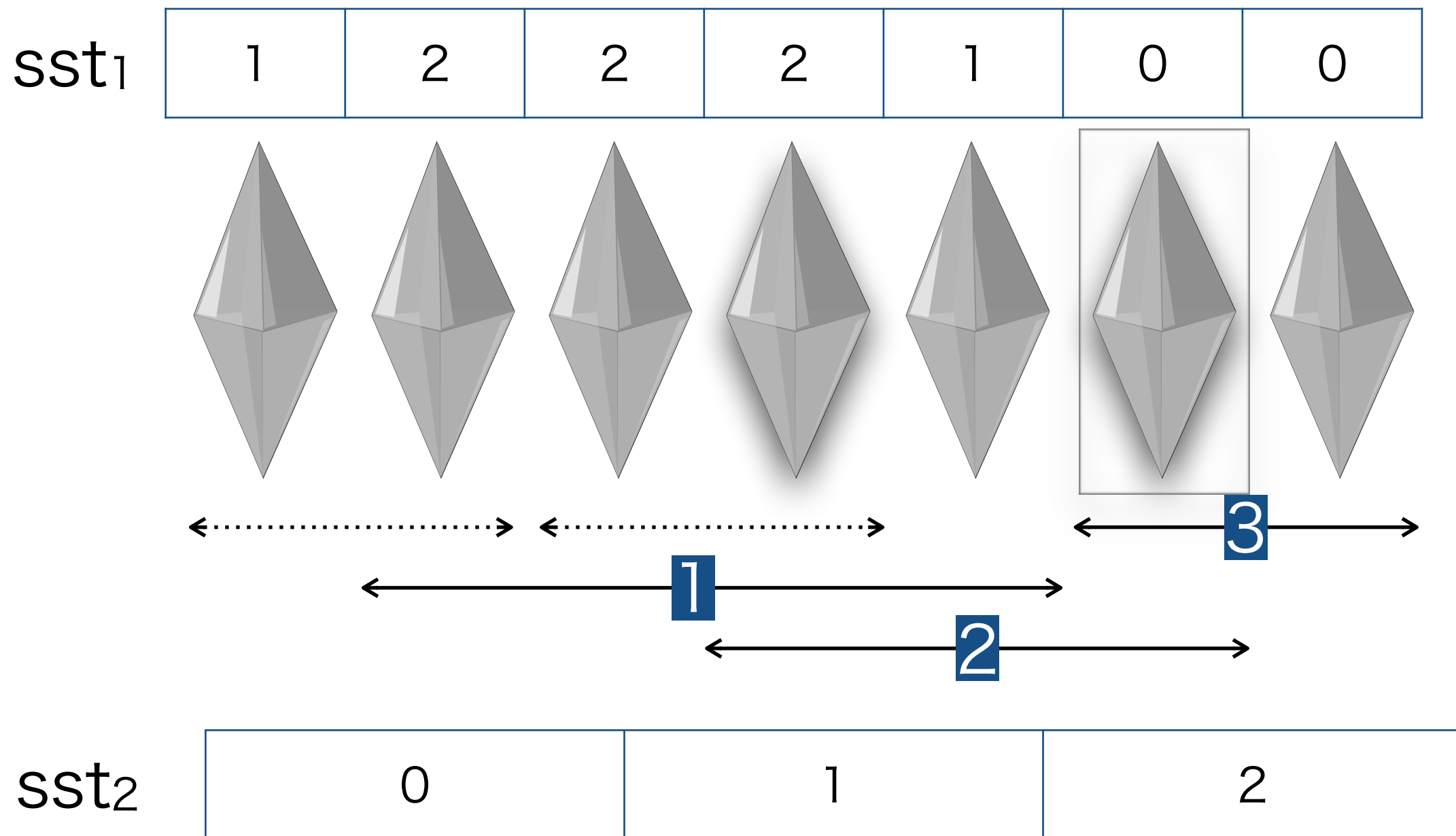
活性化済魔法石数: 5

想定解の動作

魔法石6を最後に活性化させようとする

組となり最後に詠唱する魔法の候補は2または3([2, 3])

答え = $\max(\text{答え}, \text{sst}_2[2, 3] \text{の} \max(2) + \text{活性化済魔法石数}(5))$



活性化済魔法石数: 5