# E: カリフラワーとブロッコリー

原案: 鈴木
問題文: 栗田
解答: 栗田, 杉江, 鈴木
解説:栗田

# 問題と制約

**入力:** 根付き木とそれぞれの頂点に対する2種のラベルGとW

**タスク:** 各クエリ後に，Gを持つ頂点が多いなら”broccoli”，そうでない場合”cauliflower”を出力せよ

**クエリ:** ある頂点vの子孫のラベルを反転する.

**制約:** $1 \leq n \leq 10^5$，$1 \leq q \leq 10^5$

n: 頂点数，q:クエリ数

# naive解法

毎回dfsなりbfsをして，ラベルを入れ替える.

O(nq)時間　→　**TLE**

なので，各クエリ処理を高速化する.

# 高速化のアイディア

ラベルの入れ替えとは…

2種のラベル　→　-1と1　→

-1と1の入れ替え　→　**-1の乗算**

木の子孫すべてに対するクエリ処理とは…

**オイラーツアー上の区間更新**

各クエリ処理を**区間への乗算処理**に帰着できる.

# 高速化のアイディア

ラベルの入れ替えとは…

2種のラベル　→　-1と1　→

-1と1の入れ替え　→　**-1の乗算**

木の子孫すべてに対するクエリ処理とは…

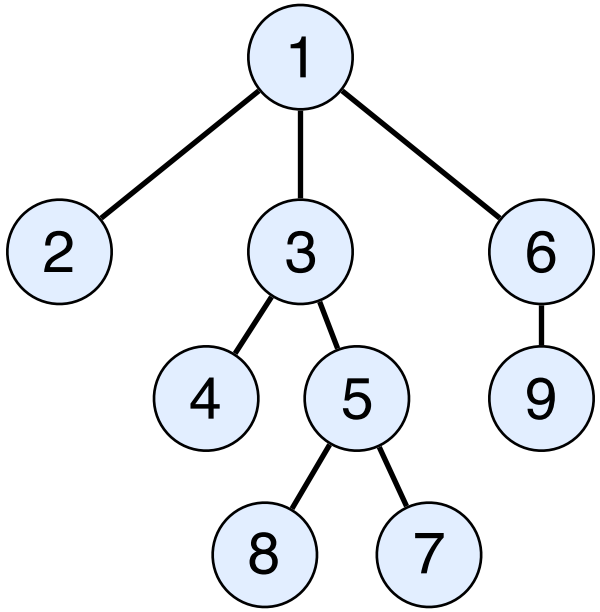
**オイラーツアー上の区間更新**

各クエリ処理を**区間への乗算処理**に帰着できる.

なんか見たことある気がする…

# 補足: オイラーツアー

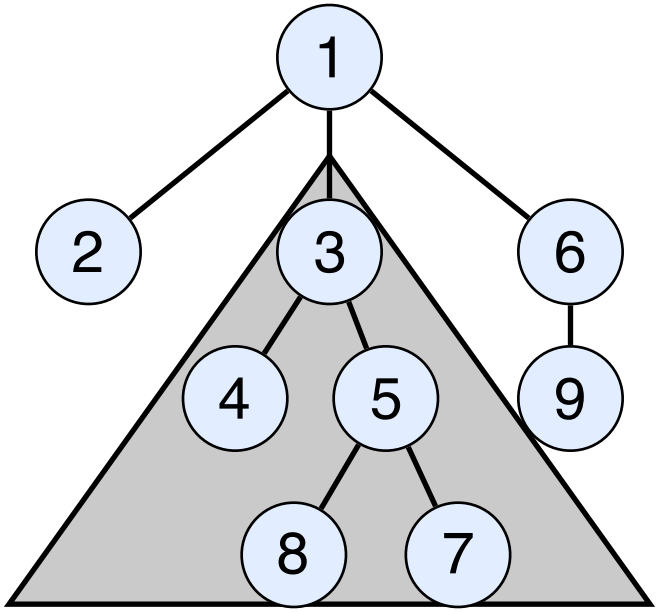
オイラーツアー: 木の頂点番号を，DFSの行きがけ順と帰りがけ順の両方で出力した頂点列




Tのオイラーツアー: 12234588775369 96

# 補足: オイラーツアー

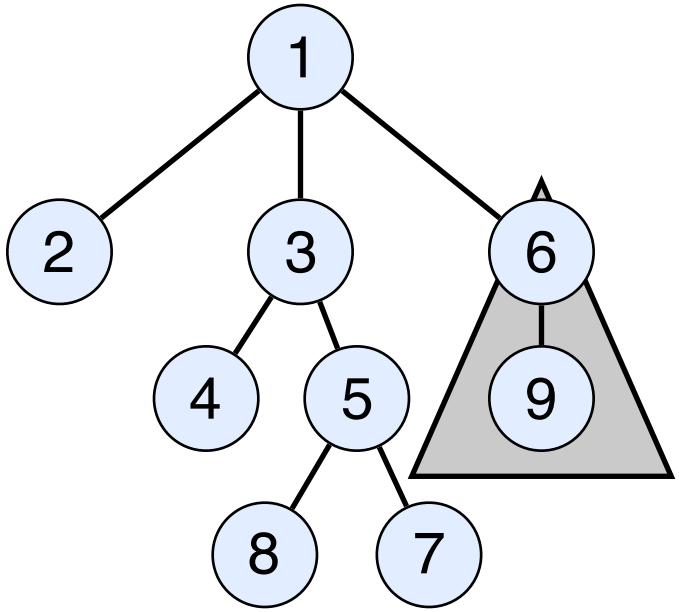
オイラーツアー: 木の頂点番号を，DFSの行きがけ順と帰りがけ順の両方で出力した頂点列




Tのオイラーツアー: 122**3445887753**6996

# 補足: オイラーツアー

オイラーツアー: 木の頂点番号を，DFSの行きがけ順と帰りがけ順の両方で出力した頂点列




Tのオイラーツアー: 1223445887753**6996**

# 区間に対する乗算の高速な処理方法

区間乗算に対する効率良いデータ構造が欲しい.

→ **遅延評価セグ木!!!**

**注)** 今回のスライドはセグ木を知っている人向けなので, セグ木を知らない人はアリ本を参照してください.

遅延評価セグ木: 区間に関する更新をO(log n)時間で処理できるセグ木の変種

# 想定解法

前処理: オイラーツアー

クエリ処理: 遅延評価セグ木

判定: セグ木の根が持つ値が…

- 正 → ブロッコリー
- 負 → カリフラワー

計算時間: O(q log n)　→　**AC!!!**

# writer解と統計情報

writer解

- 栗田: 109行 (C++)
- 杉江: 100行 (C++)
- 鈴木: 99行　(C++)

正答率

- 26/34

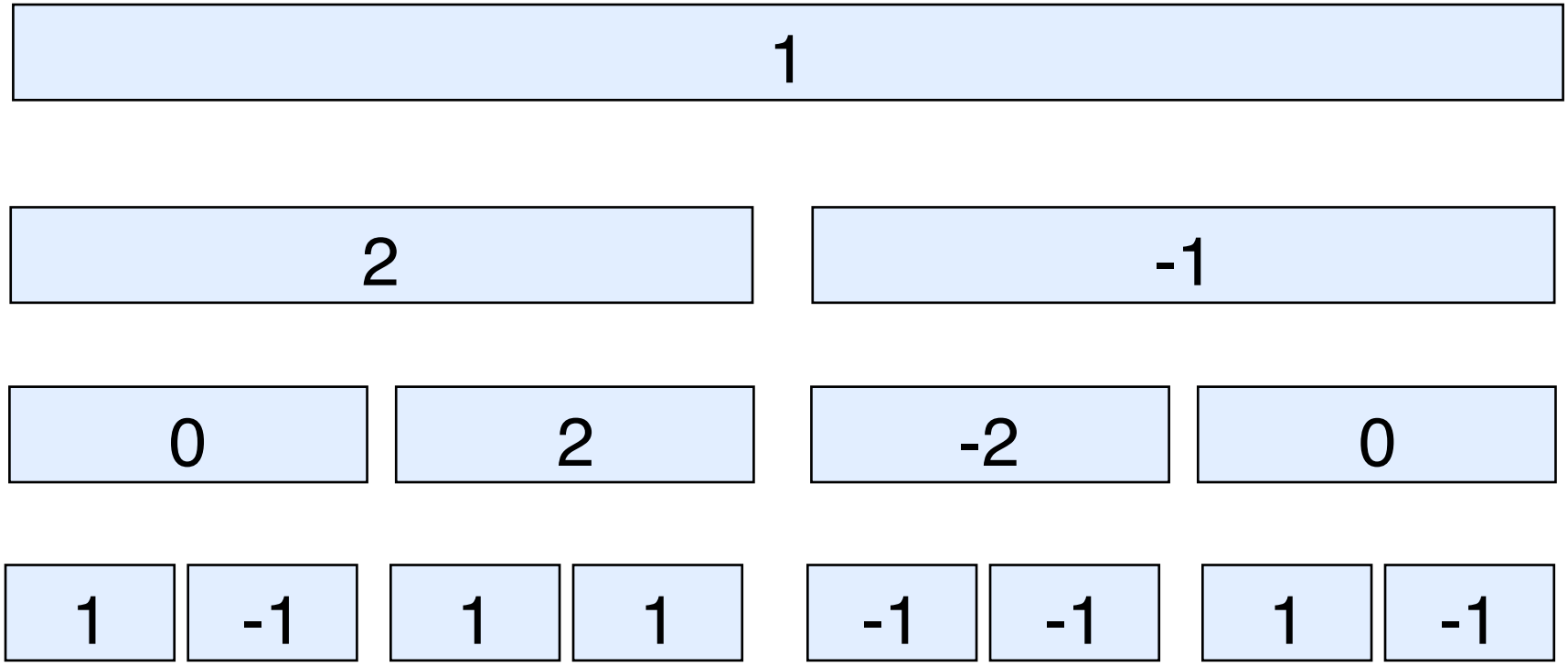
FA

- オンライン: rupc_1333parfait（34分）
- オフライン: rupc_1333parfait（34分）

# 補足: 遅延評価セグ木

区間乗算処理:

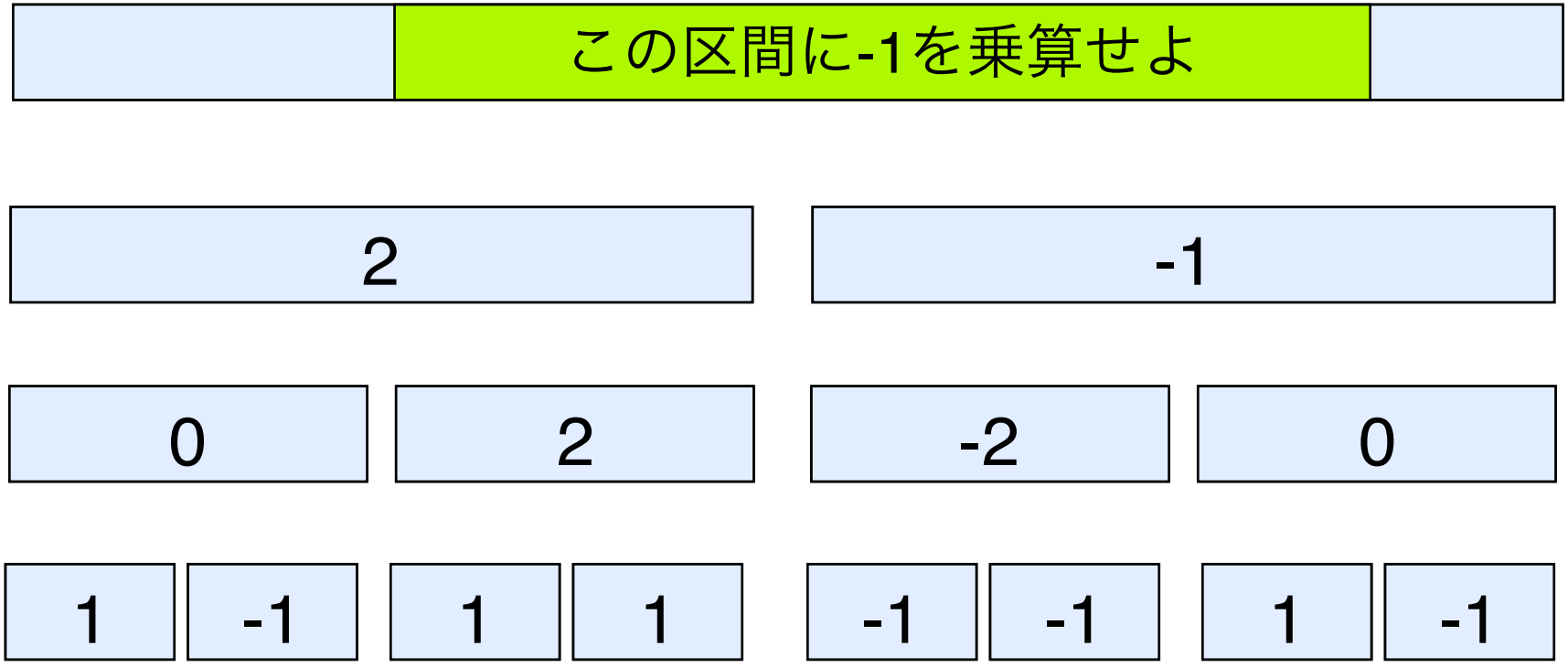
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

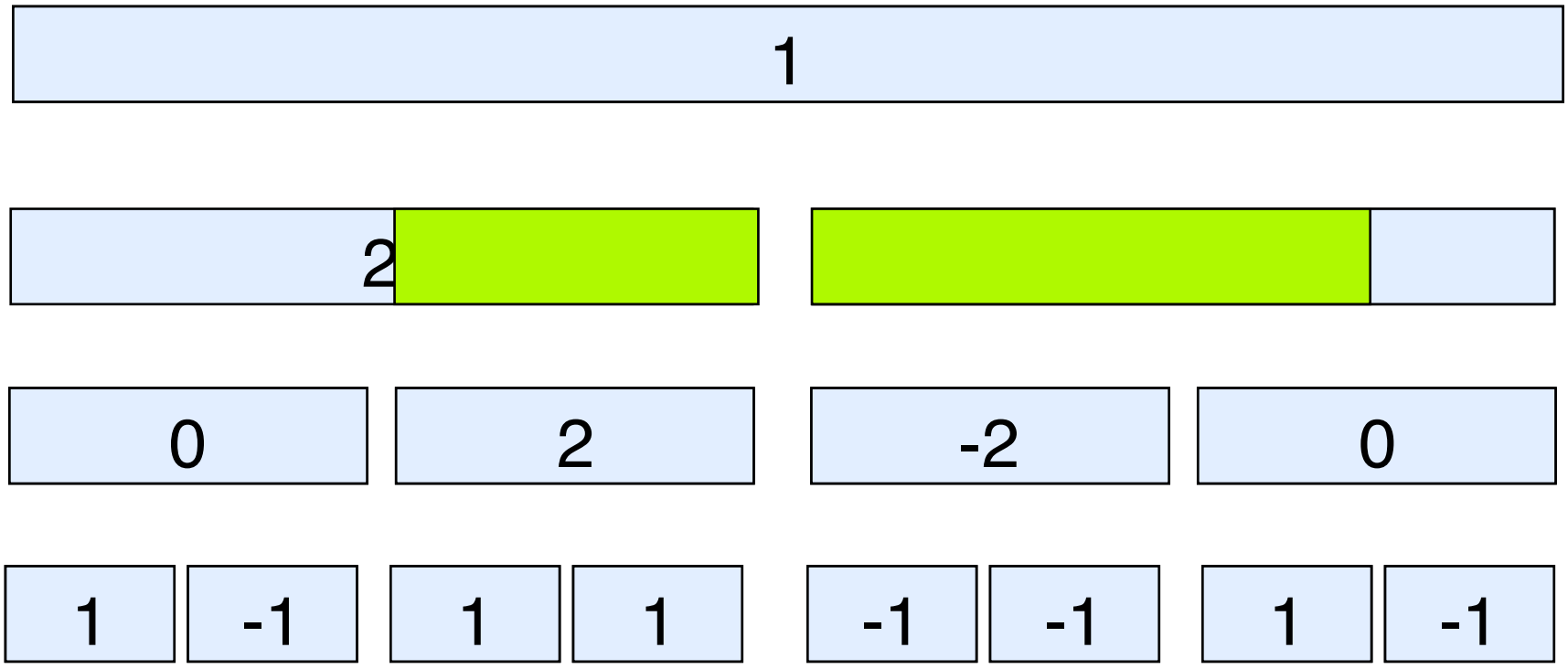
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

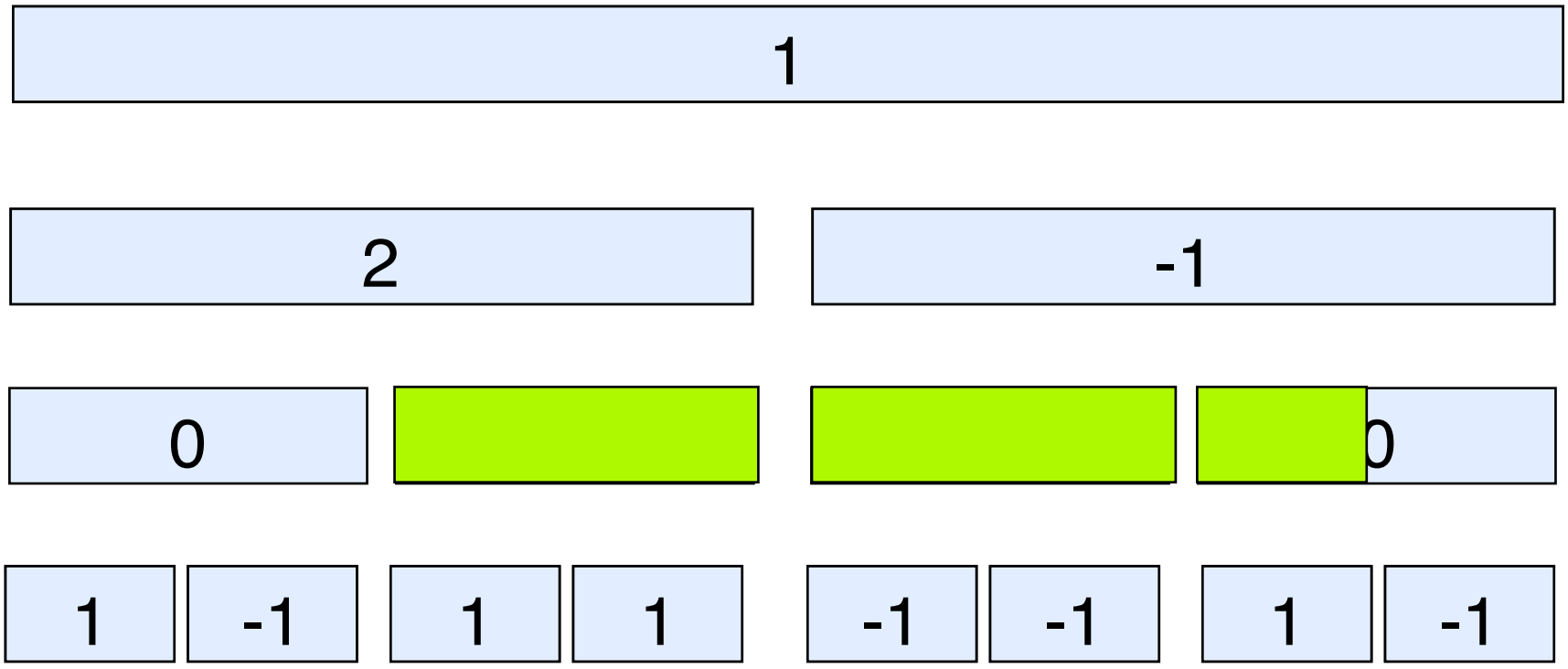
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

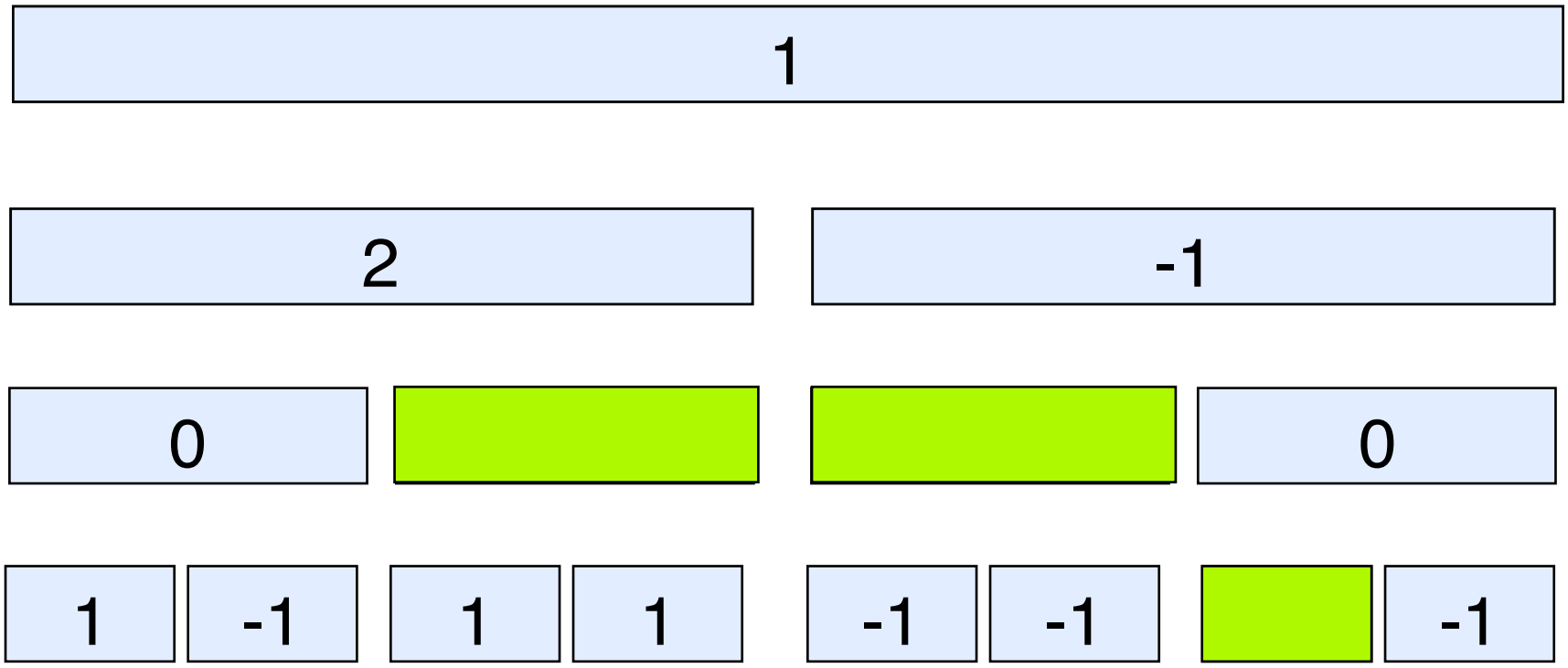
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

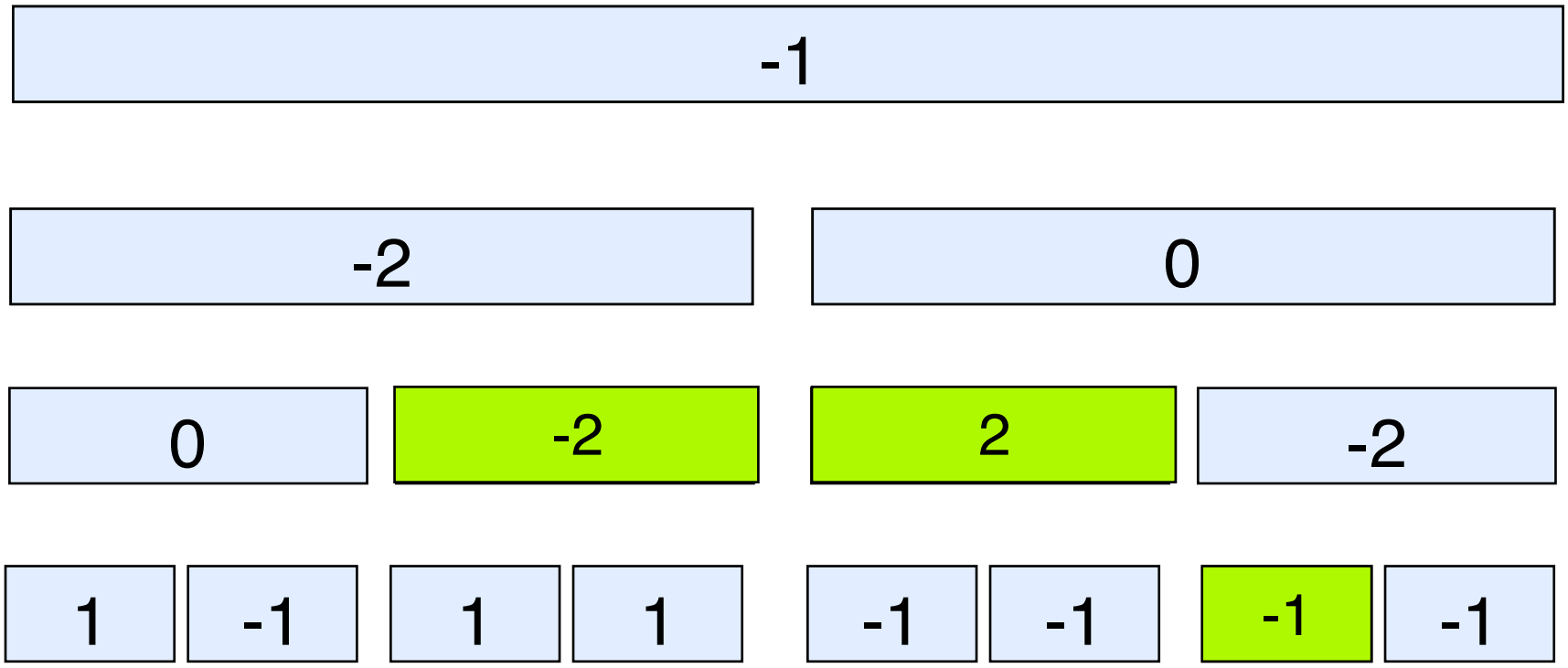
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)

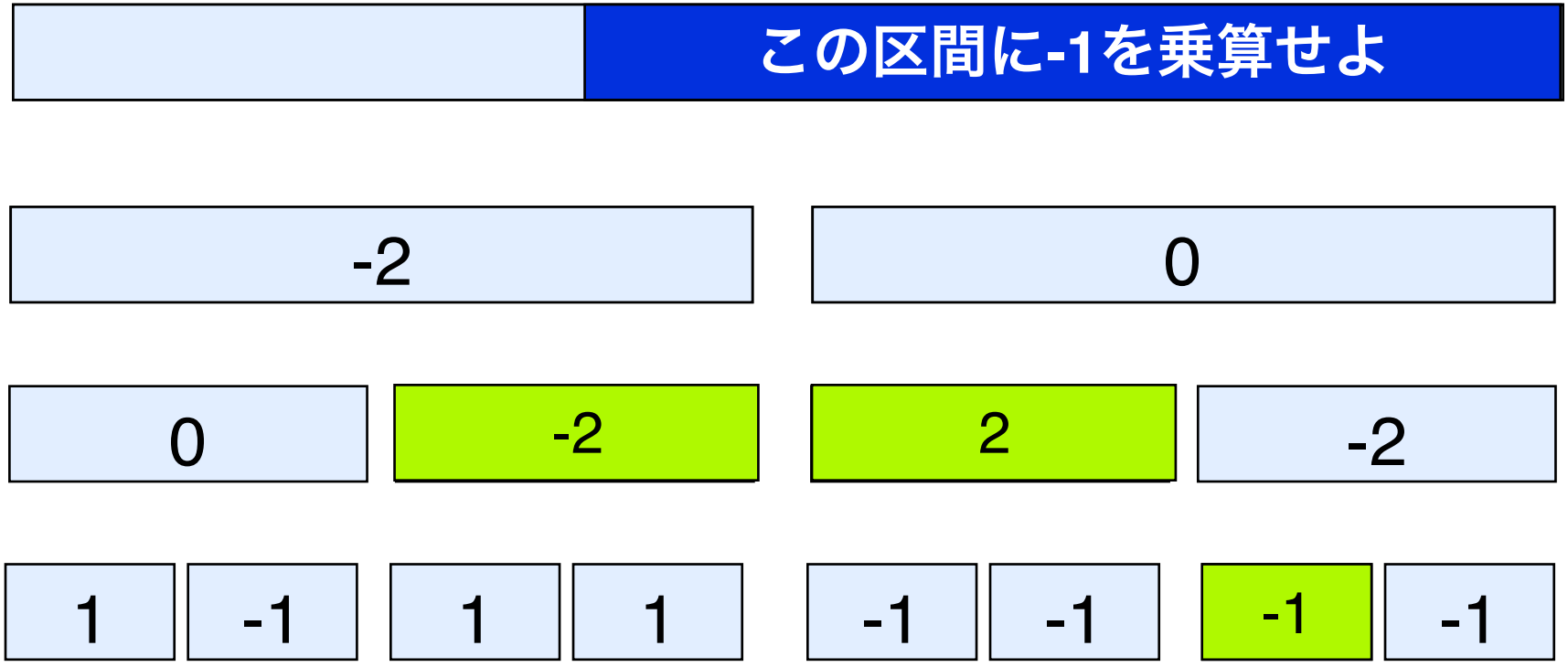


# 補足: 遅延評価セグ木

区間乗算処理:

通常のセグ木: O(k log n)

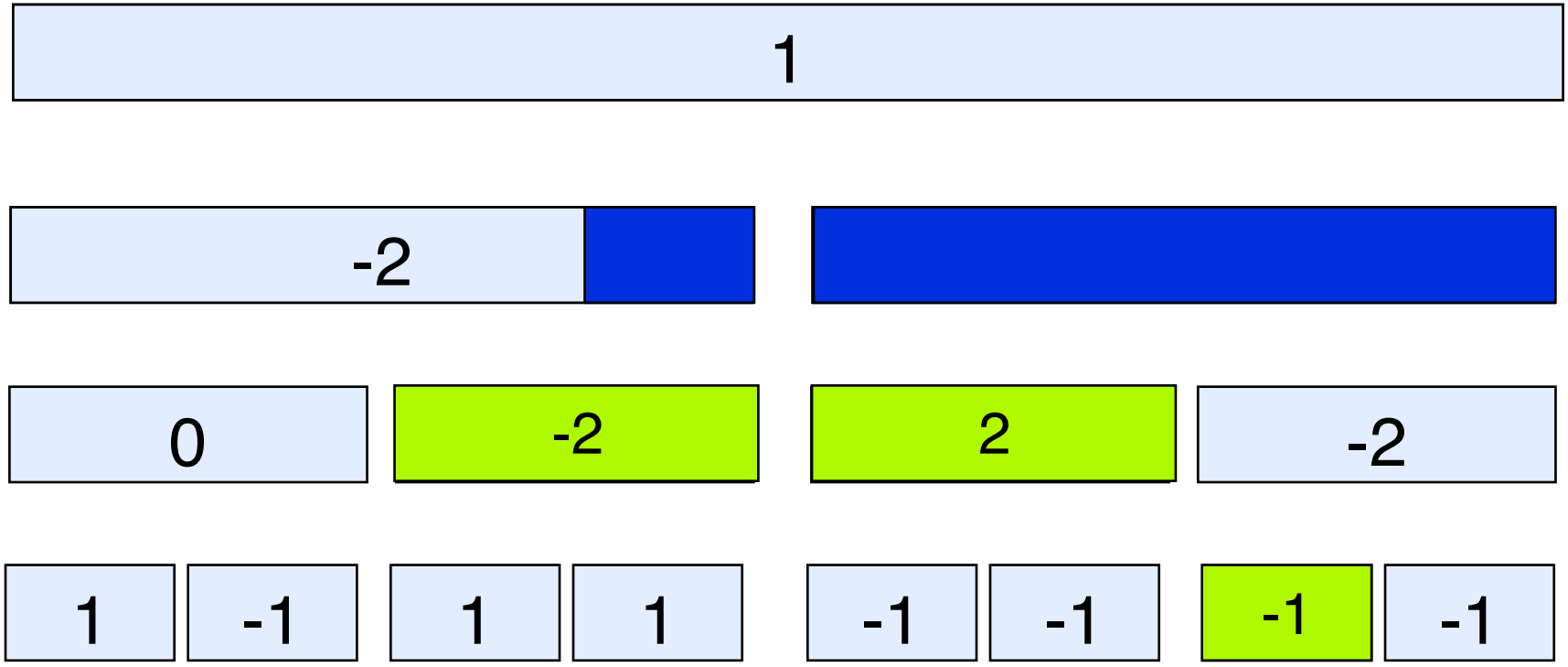
遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

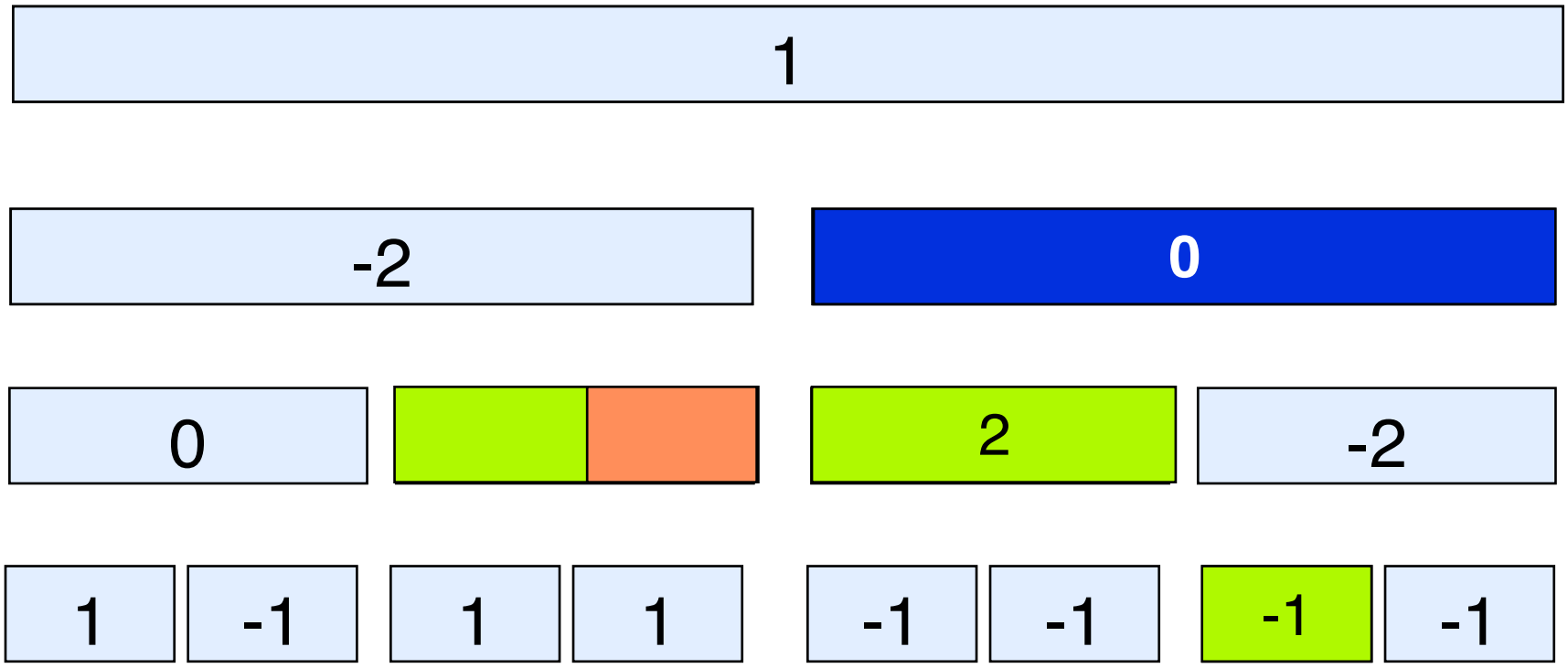
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

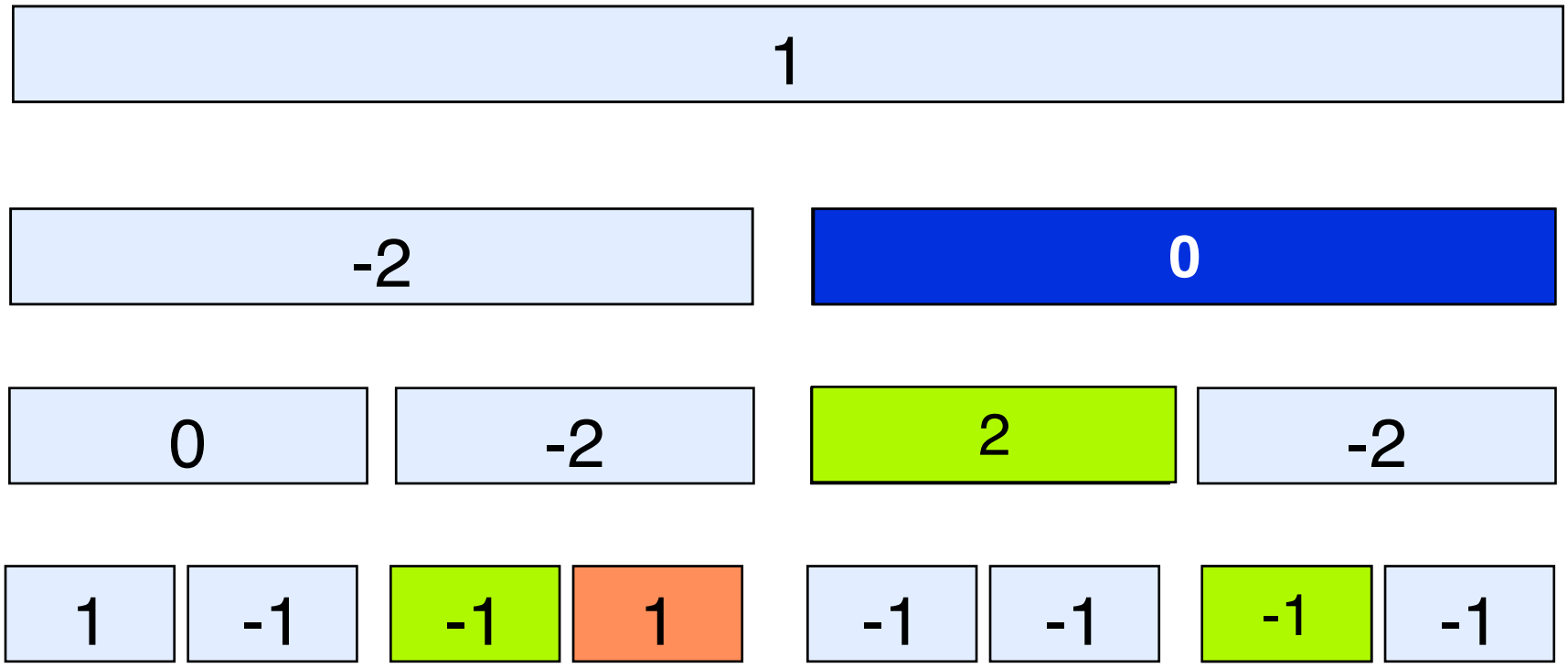
- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)



# 補足: 遅延評価セグ木

区間乗算処理:

- 通常のセグ木: O(k log n)
- 遅延セグ木: O(log n)