水ようかん 2 (Mizuyokan 2) 解説

平木 康傑

問題概要

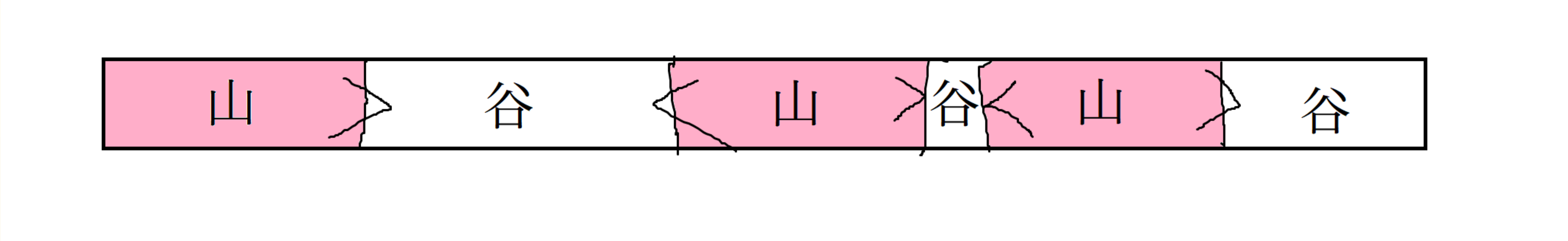
- 長さ N の正の整数列 $L_1, \ldots, L_N$ がある．
- 「L_x を y に更新」「連続部分列 $L_a, \ldots, L_b$ に対して以下の問題を解く」の 2 つのクエリを処理せよ．
  - 問題: パートごとの総和が**ジグザグ**になるように分割するとき，最大で何個に分割できる？
  - **ジグザグ**とは，(狭義の)増加・減少が交互に起こることとする．

例

- $1, 2, 3, 4, 5, 5, 6, 3$ をジグザグに分割する方法を考える.
- $1 \,/\, 2 + 3 \,/\, 4 \,/\, 5 + 5 \,/\, 6 + 3$ のように分割すると, $1 < 5 > 4 < 10 > 9$ となるので**条件を満たす**.
- $1 + 2 \,/\, 3 \,/\, 4 \,/\, 5 + 5 \,/\, 6 + 3$ のように分割すると, $3 = 3 > 4 < 10 > 9$ となる. $=$ は狭義の増加・減少ではないので, **条件を満たさない**.

言葉はあった方が便利

- 和が大きい方のパートを「**山**」，
  和が小さい方のパートを「**谷**」と呼ぶことにする．



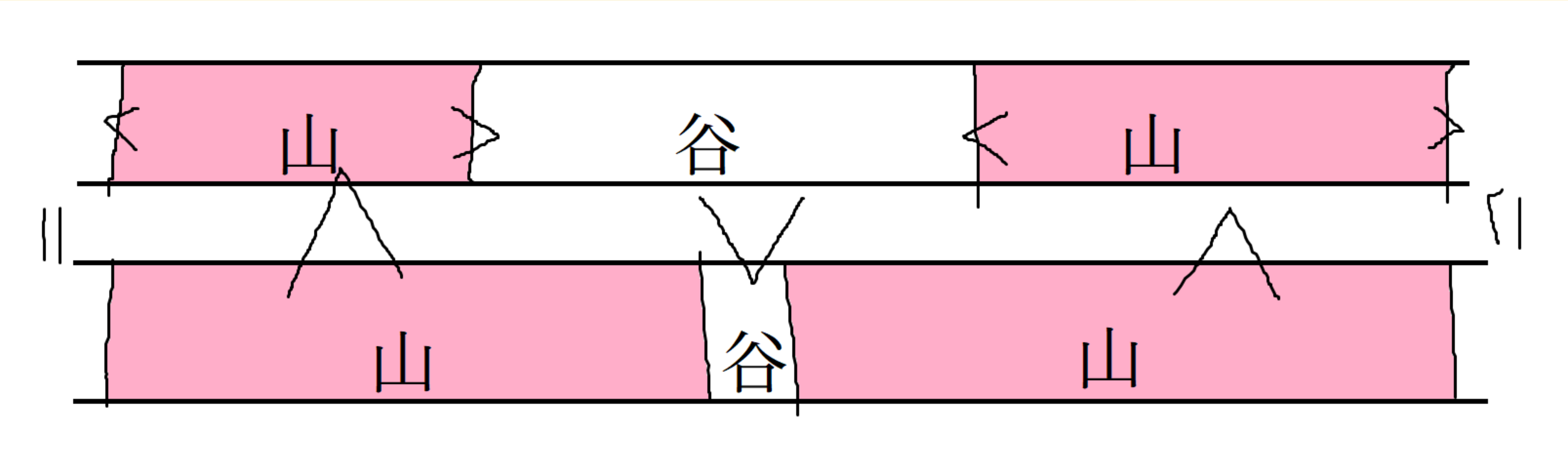
- $1 \,/\, 2 + 3 \,/\, 4 \,/\, 5 + 5 \,/\, 6 + 3$ という分割なら
  谷，山，谷，　山，　谷.

小課題

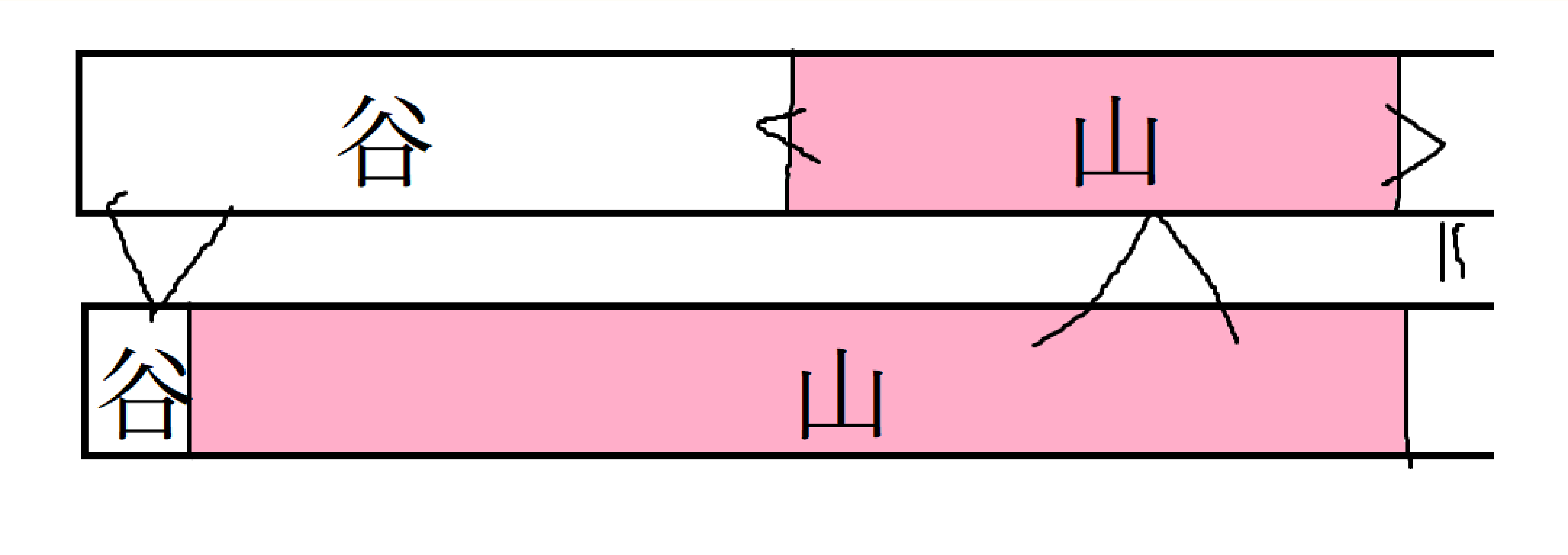
1. $N \leqq 200$，$Q \leqq 10$
2. $N \leqq 2\,000$，$Q \leqq 10$
3. $Q \leqq 10$
4. 更新なし
5. 数列の要素は $120,000$ 以下
6. 追加の制約はない

- 小課題 1 は区間 DP で，小課題 2 はその高速化で解ける．

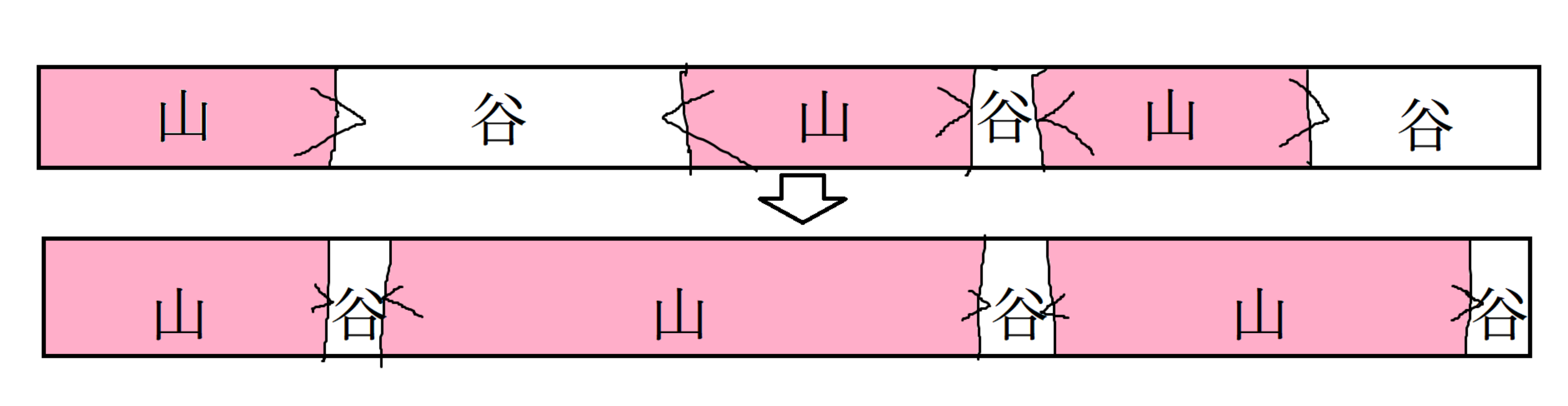
考察 1-a: 2 つの山の間にある谷について，両隣の山をのばすことで，谷の長さを 1 にできる．



考察 1-b: 端にある谷について，隣の山を適切にのばすことで，谷の長さを 1 にできる．



考察 1: 任意の valid な分割に対して，山を適切にのばせば，もとと同じ分割個数で谷の長さがすべて 1 であるような valid な分割が作れる．



- つまり，谷の長さが 1 であるような分割のみを考えれば十分．

考察 1+: 谷の長さが 1 であるような分割のみを考えれば十分.

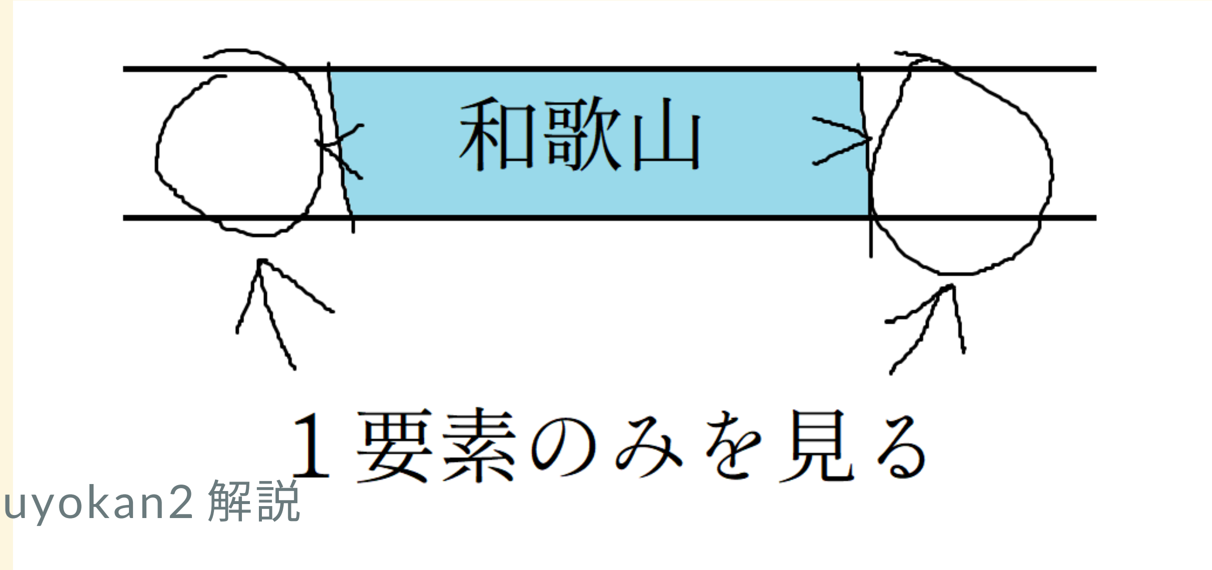
- 次のような DP をすればよい:
  - $dp[i]$:= L_i が谷であるときの，そこまでの分割個数の最大値.
- $L_i < L_{i+1} + \cdots + L_k$ を満たす k は，$k_i \leq k \leq N$ の形で表せる.
- $L_k + \cdots + L_{i-1} > L_i$ を満たす k は，$0 \leq k \leq k'_i$ の形で表せる.
- 上の性質から，セグメント木などを使った高速化が効く(詳細略).
  - 小課題 3 ($Q \leq 10$) は通る.

谷と山の両方が絡むとややこしい

- 先ほどは「谷」のみに注目する方針をとった．
- 「山」のみに注目する方針が取れないか？

山のみに注目したい

- 両隣の要素のみを見て判断できるような山の亜種を考える．
- $L_{l-1} < L_l + \cdots + L_r > L_{r+1}$ ($L_{-1} = L_{N+1} = -\infty$ とする)を満たすような区間 $[l, r]$ を**和歌山**と呼ぶことにする．
  - 山は必ず和歌山である．

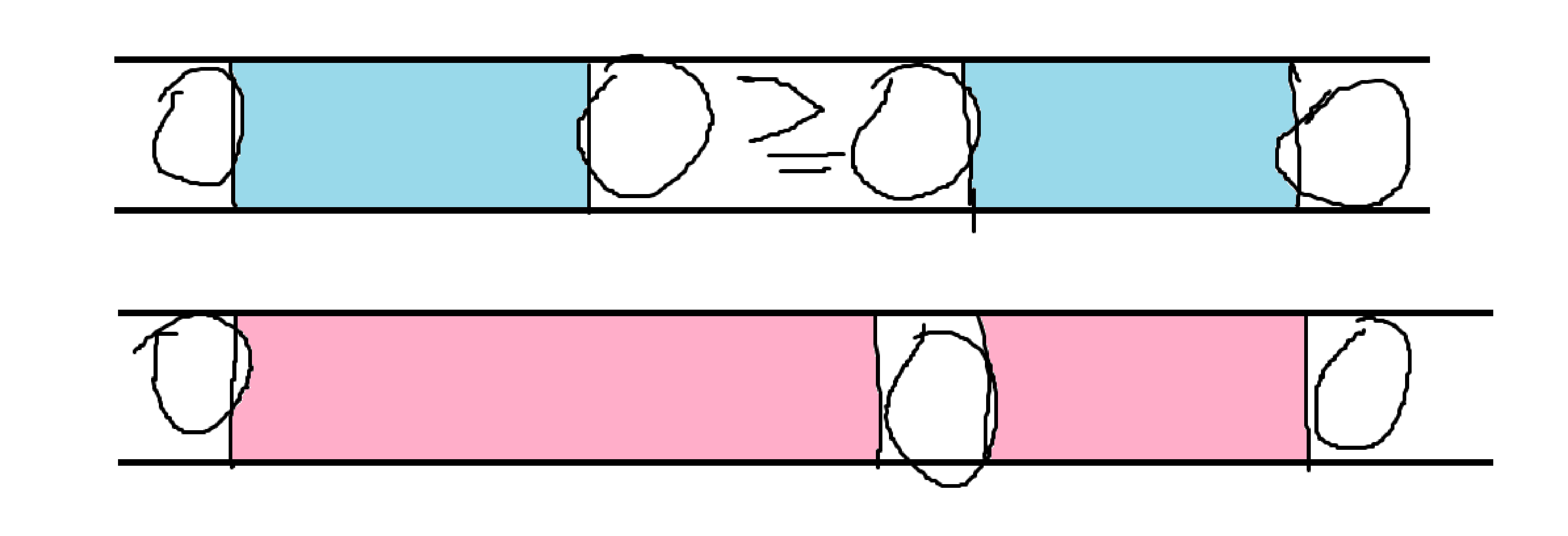


考察 1: 任意の valid な分割に対して，山を適切にのばせば，もとと同じ分割個数で谷の長さがすべて 1 であるような valid な分割が作れる．

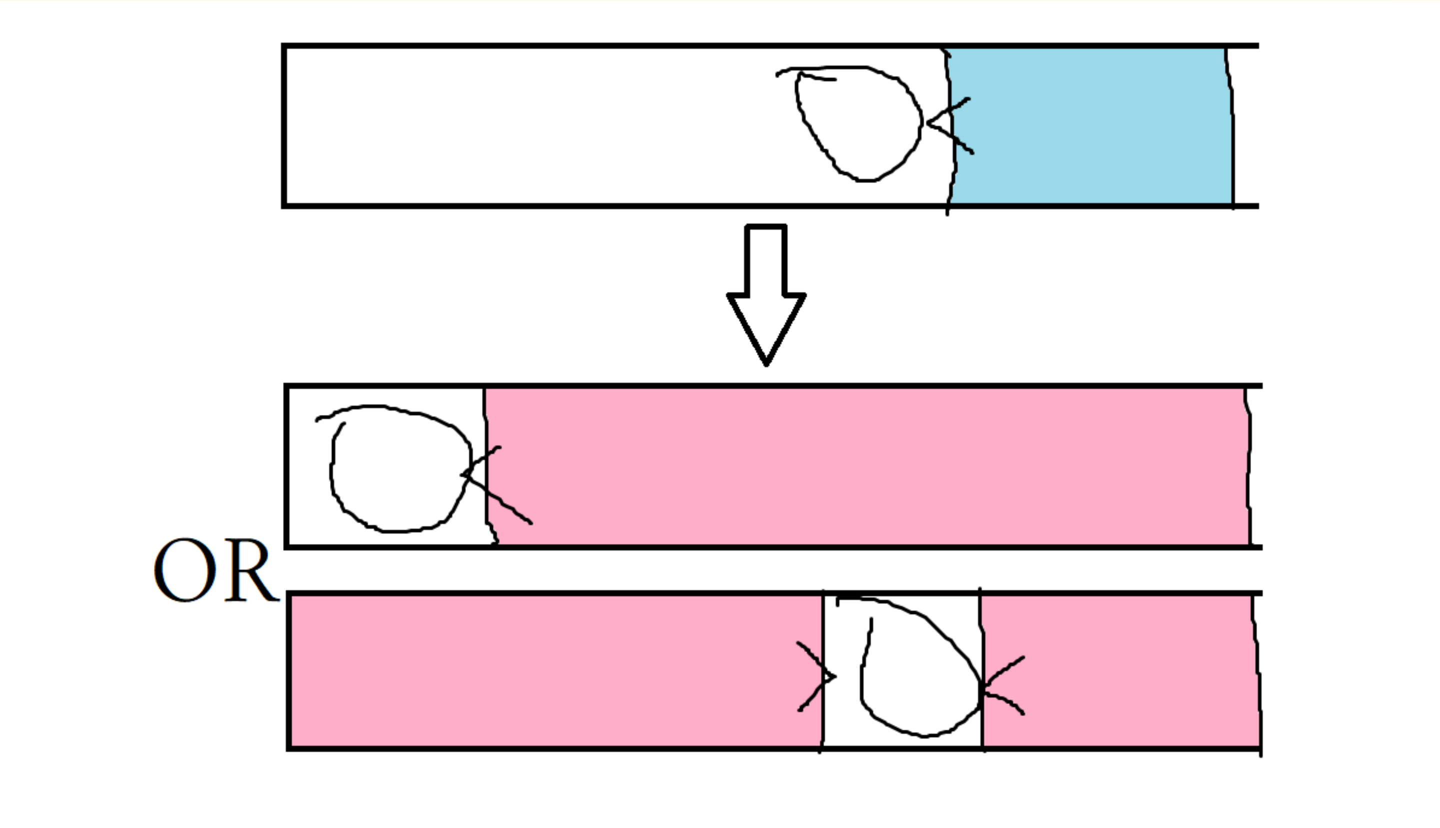
仮説 2: 任意の valid な和歌山分割に対して，和歌山を適切にのばせば，分割個数を減らさずに，谷の長さがすべて 1 であるような valid な分割が作れる．

- ここで**和歌山分割**は，和歌山 $[l_1, r_1], \ldots, [l_k, r_k]$ の取り方であって $l_{k+1} - r_k \geqq 2$ を満たすようなものとする．

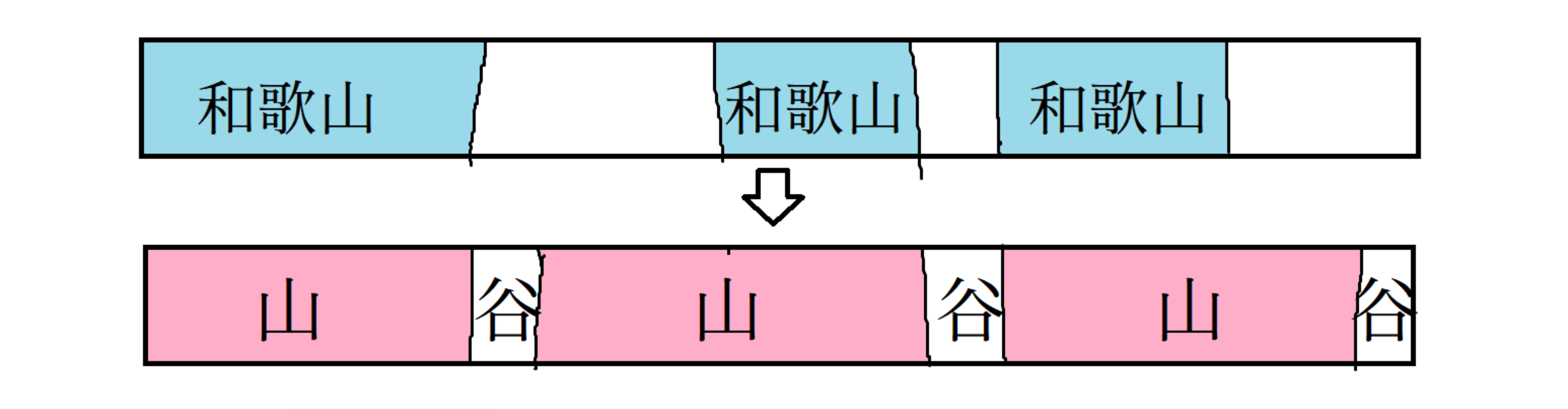
考察 2-a: 2 つの和歌山の間の谷モドキについて，どちらかの和歌山をのばすことで，長さ 1 の谷にできる．



考察 2-b: 端にある谷モドキについて，隣の和歌山をのばすことで，長さ 1 の谷にできる．



考察 2: 任意の valid な和歌山分割に対して，和歌山を適切にのばせば，分割個数を減らさずに，谷の長さがすべて 1 であるような valid な分割が作れる．



考察 $2+$: 和歌山分割の最大個数は，分割の最大個数と一致する．

考察 $1+$: 谷の長さが 1 であるような分割のみを考えれば十分．

考察 2: 任意の valid な和歌山分割に対して，和歌山を適切にのばせば，分割個数を減らさずに，谷の長さがすべて 1 であるような valid な分割が作れる．

小考察: valid な分割は，和歌山分割としても valid．

考察 2+: 和歌山分割の最大個数は，分割の最大個数と一致する．

- ある区間が山であるかは，前後の山の選び方に依存する．
- ある区間が和歌山であるかは，その区間と両隣の要素の値しか関係しない(そこだけ見れば判断できる！)
- つまり，もともと考えていた分割よりも和歌山分割のほうが扱いやすい．

和歌山分割は

- ある区間が和歌山であるかは，その区間と両隣の要素の値だけが関係する．
  - 和歌山として valid な区間 $[l, r)$ がたくさん与えられるので，$[l, r]$ を被らないように選べばよい．

和歌山分割は区間スケジューリング

- ある区間が和歌山であるかは，その区間と両隣の要素の値だけが関係する．
  - 和歌山として valid な区間 $[l,r)$ がたくさん与えられるので，$[l,r]$ を被らないように選べばよい．
- 区間スケジューリングに帰着された．
- 区間スケジューリングは，選べるもののうち r が最小のものを貪欲に選ぶのが最適解．
  - 和歌山分割に言い換える前の分割では貪欲法は誤りなので注意．

小課題 4: 値は更新されない

- 直前の r から飛べる次の r の最小値を $f(r)$ とする．
  - 次の区間を $[l_{next}, r_{next}]$ とおくと，$l_{next} \geqq r+1$ の場合は $f(r+1)$ でカバーできるので，$l_{next} = r+1$ の場合のみを求めればよい．
- $f^x(r) \geqq b$ を満たす最大の x を求めればよい．
  - 端の処理に注意．端の山・谷の組合せを 4 通り試すと確実．

小課題 4: 値は更新されない

- 直前の r から飛べる次の r の最小値を $f(r)$ とする．$f^x(r) \geqq b$ を満たす最大の x を求めればよい(端の処理に注意).
- 値が更新されないので，ダブリングによって各クエリが $\Theta(\log N)$ 時間で処理できる.
  - ダブリング: $f^{(2^k)}(r) = f^{(2^{k-1})}(f^{(2^{k-1})}(r))$ を利用して $f^{(2^k)}(r)$ を前計算できるうえ，$f^x(r) \geqq a$ を満たす最大の x が二分探索で求まる.

小課題 $5, 6$: 値は更新される

- 一点更新とはいえ，それを含むすべての区間に影響するので，$f(r)$ が変わる r は多いのではないか？

考察 3: 一度に飛ぶ距離 $f(r) - r$ は，高々 $2\lfloor\log A_{max}\rfloor + 4$.

- $c := r + \lfloor\log A_{max}\rfloor + 2$ とする.
- $L_k \geqq L_{k+1} + \cdots + L_c$ $(r \leqq k \leqq c - 1)$ が成り立つとすると，
  $L_k \geqq 2^{c-1-k} L_c \geqq 2^{c-1-k}$ (帰納法で示せる) より
  $L_r \geqq 2^{c-1-r} = 2^{\lfloor\log A_{max}\rfloor+1} > A_{max}$ となり，**矛盾**.
  したがって，$L_k < L_{k+1} + \cdots + L_c$ が少なくとも 1 ヶ所の
  $r \leqq k \leqq c - 1$ で成り立つ.
- 同様に $L_c + \cdots + L_{k-1} > L_k$ が少なくとも 1 ヶ所の
  $c + 1 \leqq k \leqq 2c - r = r + 2\lfloor\log A_{max}\rfloor + 4$ で成り立つ.

考察 3: 一度に飛ぶ距離 $f(r) - r$ は，高々 $2\lfloor \log A_{max} \rfloor + 4$.

- $c := r + \lfloor \log A_{max} \rfloor + 2$ とする.
- $L_k < L_{k+1} + \cdots + L_c$ が少なくとも 1 ヶ所の $r \leqq k \leqq c - 1$ で成り立つことが示せた.
- 同様に，$L_c + \cdots + L_{k-1} > L_k$ が少なくとも 1 ヶ所の $c + 1 \leqq k \leqq 2c - r = r + 2\lfloor \log A_{max} \rfloor + 4$ で成り立つことも示せる.
- したがって，$r \leqq s, t \leqq r + 2\lfloor \log A_{max} \rfloor + 4$ を満たす和歌山の区間 (s, t) が存在する.

これで更新クエリがラクになる

- L_x を一点更新したとき，$f(r)$ が変わりうるのは $x-(2\lfloor\log A_{max}\rfloor+4) \leqq r \leqq x$ の範囲のみ．
- 愚直にやっても $O(\log^3 A_{max})$ 時間で済むし，小課題 4 と同様に $f(r+1)$ を利用する方針でいけば $O(\log^2 A_{max})$ 時間になる．

トドメのセグメント木

- $seg[l, r)[k] :=$ k から出発して r を超えないように繰り返し飛ぶとき，最大何回でどこまで飛ぶか
    - 必要な k の範囲は $l \leqq k \leqq l + 2\lfloor \log A_{max} \rfloor + 4$
- $seg[l, r) = seg[l, m) * seg[m, r)$ の二項演算は $O(\log A_{max})$ 時間．
    - $f(\cdot)$ を利用すれば，$O(\log A_{max})$ 個ある k それぞれについて $seg[l, r)[k]$ が定数時間で求まるため．
- セグ木の更新と区間取得は，セグ木区間の合成を $O(\log N)$ 回行うのでともに $O(\log A_{max} \log N)$ 時間．

以上より，この問題は

- 谷を長さ 1 につぶす方法を思いつき，

以上より，この問題は

- 谷を長さ 1 につぶす方法を思いつき，
- 谷ではなく山に注目することで区間スケジューリングに帰着し，

以上より，この問題は

- 谷を長さ 1 につぶす方法を思いつき，
- 谷ではなく山に注目することで区間スケジューリングに帰着し，
- 必要な区間の長さが高々 $2\lfloor\log A_{max}\rfloor + 4$ であることに気付いて，

以上より，この問題は

- 谷を長さ 1 につぶす方法を思いつき，
- 谷ではなく山に注目することで区間スケジューリングに帰着し，
- 必要な区間の長さが高々 $2\lfloor \log A_{max} \rfloor + 4$ であることに気付いて，
- ダブリングの考え方をセグメント木に乗せることで，
- クエリあたり $O(\log A_{max}(\log A_{max} + \log N))$ 時間で解くことができました！

やっべスライド作るの間に合わない

スクショ載せます