

G . Longest Increasing Sequence

原案：岸本 (tokoharu)

問題文：岸本 (tokoharu)

解答：岸本 (tokoharu), 河田 (kawatea)

解説：岸本 (tokoharu)

問題概要

- ◆ 数列があって適当な場所で切り分ける
 - 同じ場所で切っちゃダメ
- ◆ 切り分けた部分ごとにすべて足し合わせる
- ◆ 出来上がった数列が増加列になっていればよい
 - 1,1とかは増加列ではないので注意
- ◆ 出来上がる数列の最大の長さ、そうなるような切り分け例を挙げるのが目的。
 - 例えば、(6,5,4,3,2,1)が与えられる
 - 1番目のところで切り分けて((6),(5,4,3,2,1)) → (6,15)で増加列(これで最大の長さ)
- ◆ 制約 : $1 \leq N \leq 4000$, $|A_i| \leq 10^8$

考察1 (想定TLE解法 p. 1)

- ◆ 累積和をとっておけば、任意区間の整数の和は $O(1)$ で求めることができる。
- ◆ 区間に対してDPを考えてみる
 - $dp[始点][終点]$:= 「始点終点にそれぞれ切れ目を入れて、始点位置から終点位置までの領域に切れ目を入れず、終点までにできる増加列の長さで最大のもの」とする。
 - 不可能なものは $-inf$ (無限大の意味) とする
- ◆ $dp[始点][終点]$ の求め方
 - $dp[i][始点]$ ($0 \leq i < 始点$) のうちで最大のものに1を加えるとよい

考察2 (想定TLE解法 p. 2)

- ◆ 計算量について
 - 求めたい場所は $O(N^2)$ 個だけある。
 - 求めたい値を得るためにそれぞれ $O(N)$ だけ時間をかけなければならない。
 - 以上の考察から時間計算量は $O(N^3)$
- ◆ $N \leq 4000$ なので代入すると $6.4 * 10^{10}$ くらいの見積もり
 - 想定TLE
 - ただし定数項はかなり軽いため、 $N=1000$ だと思ったより時間がかからない

考察3 (想定解 p. 1)

- ◆ 今まででよくなかったところ
 - 今までは3つの要素のうち、[始点][終点]を固定させ、その前の点を変動させていて考えていた。
- ◆ なのでちょっと変えてみる
 - こんどは真ん中の点(今まで始点として見ていたもの)のみを固定してみよう
- ◆ 必要なものを再確認
 - [今の点][次の点]を求めるために必要なものは、[前の点][今の点]のもので、 $\text{Sum}(\text{今の点}, \text{次の点}) > \text{Sum}(\text{前の点}, \text{今の点})$ を満たすもので最大のものである

考察4 (想定解 p. 2)

- ◆ 実はソートをするとうまくいく
 - つまり、 $\text{Sum}(\text{今の点}, \text{次の点})$ と $\text{Sum}(\text{前の点}, \text{今の点})$ をマージしてソートする。
 - そして、ソートの結果を小さい方からなめていくことを考えてみる。
- ◆ 言葉だと分かりにくいので後のページの図を見ながらだとわかりやすいかも

考察5 (想定解 p. 3)

- ◆ ある「今の点」について以下の事を行う
- ◆ 二つの変数 `cand`, `best` を持っておく
 - `dp[前の点][今の点]` で最大値を達成するような前の点を `cand` としておく。
 - `dp[前の点][今の点]` の最大値を `best` として保存しておく
 - `best` の初期値は `-inf` (かなり小さな値) としておく
- ◆ そして以下のように更新処理を行う
 - 1. 前の点に関する情報がきたとき
 - `cand`, `best` に更新する必要があるら更新する
 - 2. 次の点に関する情報がきたとき
 - `dp[今の点][次の点]` の値に `best` を入れる
 - 復元用の `prev[今の点][次の点]` に `cand` を入れる

文章は衰退しました p. 1

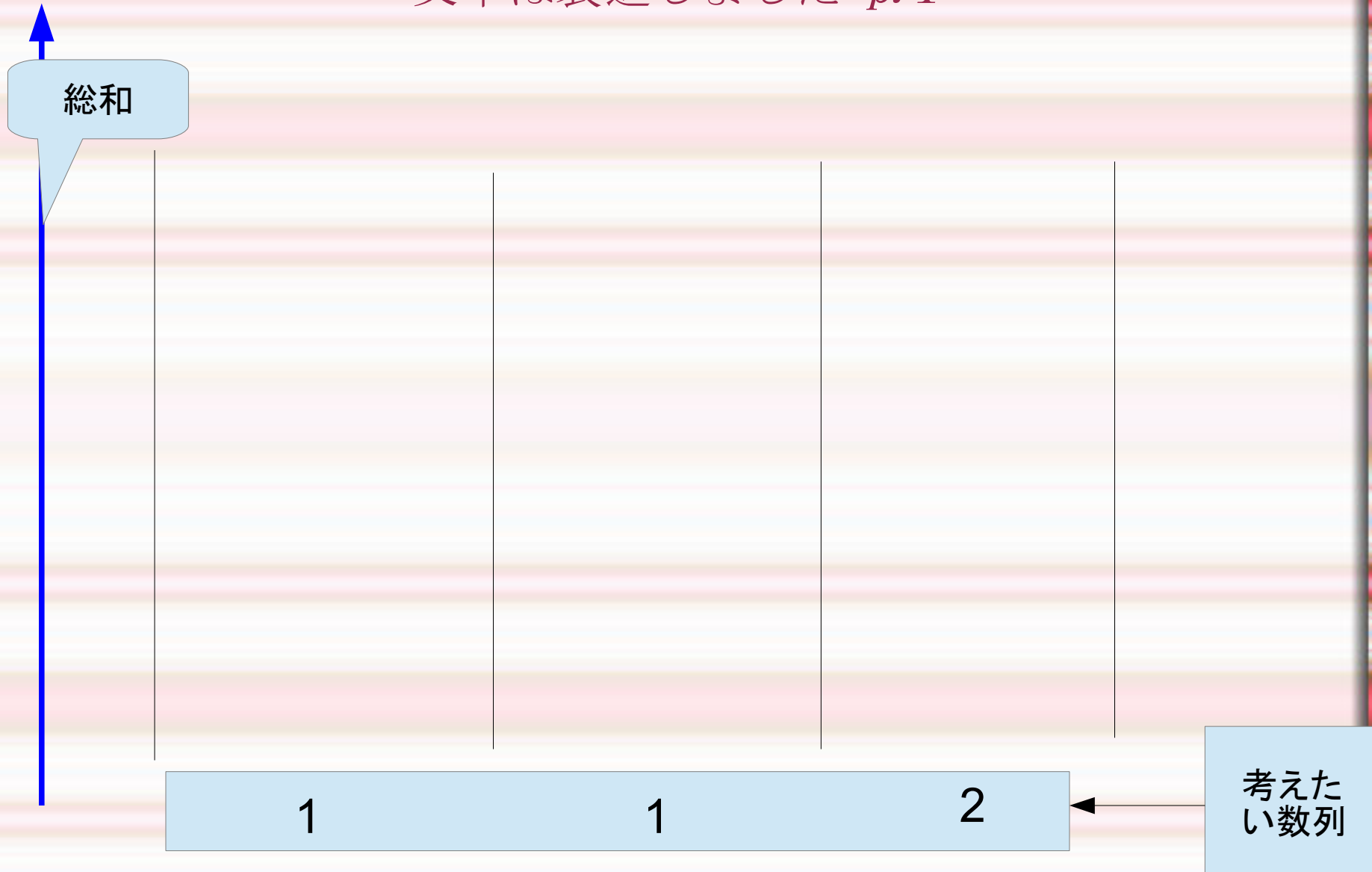
総和

1

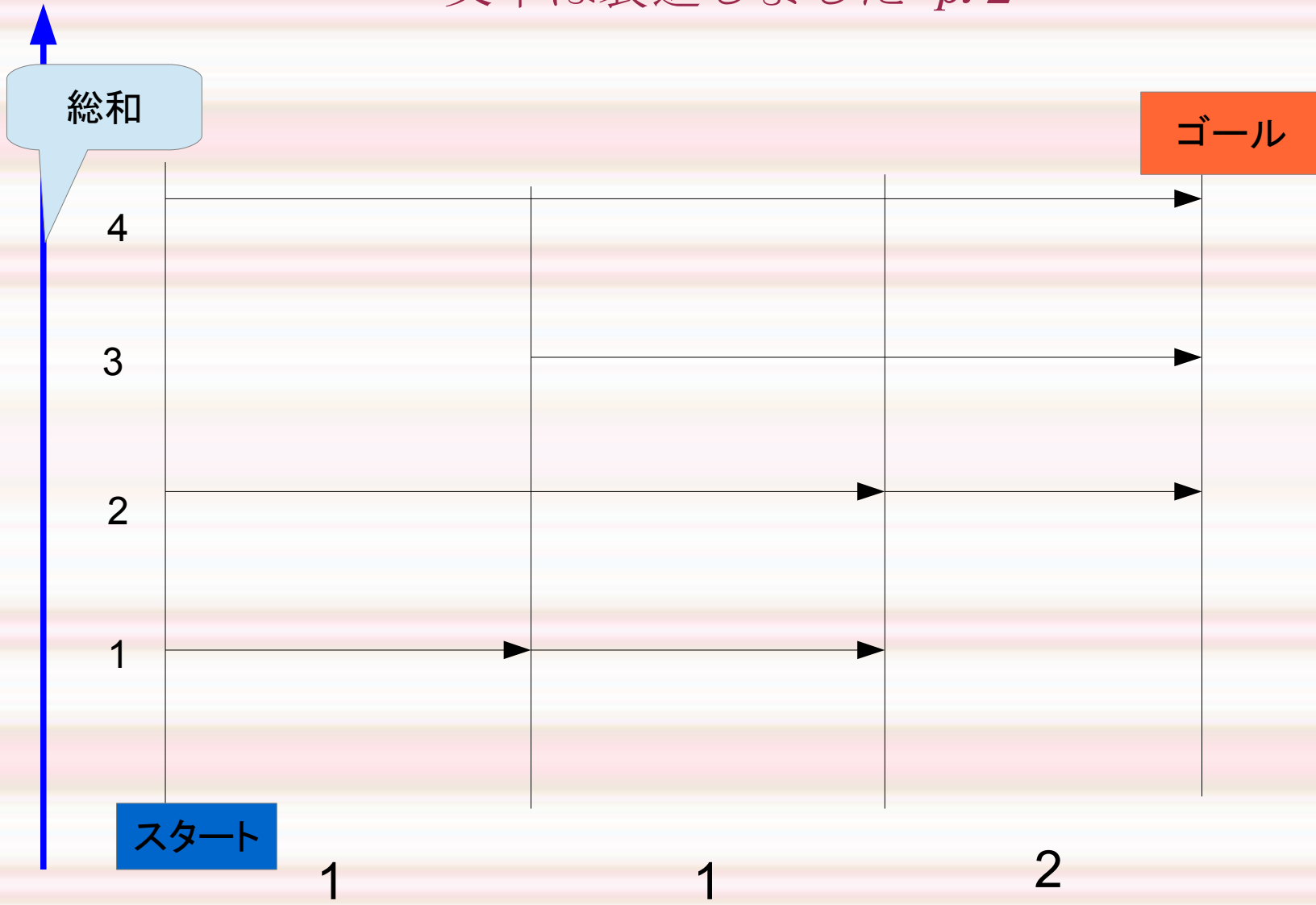
1

2

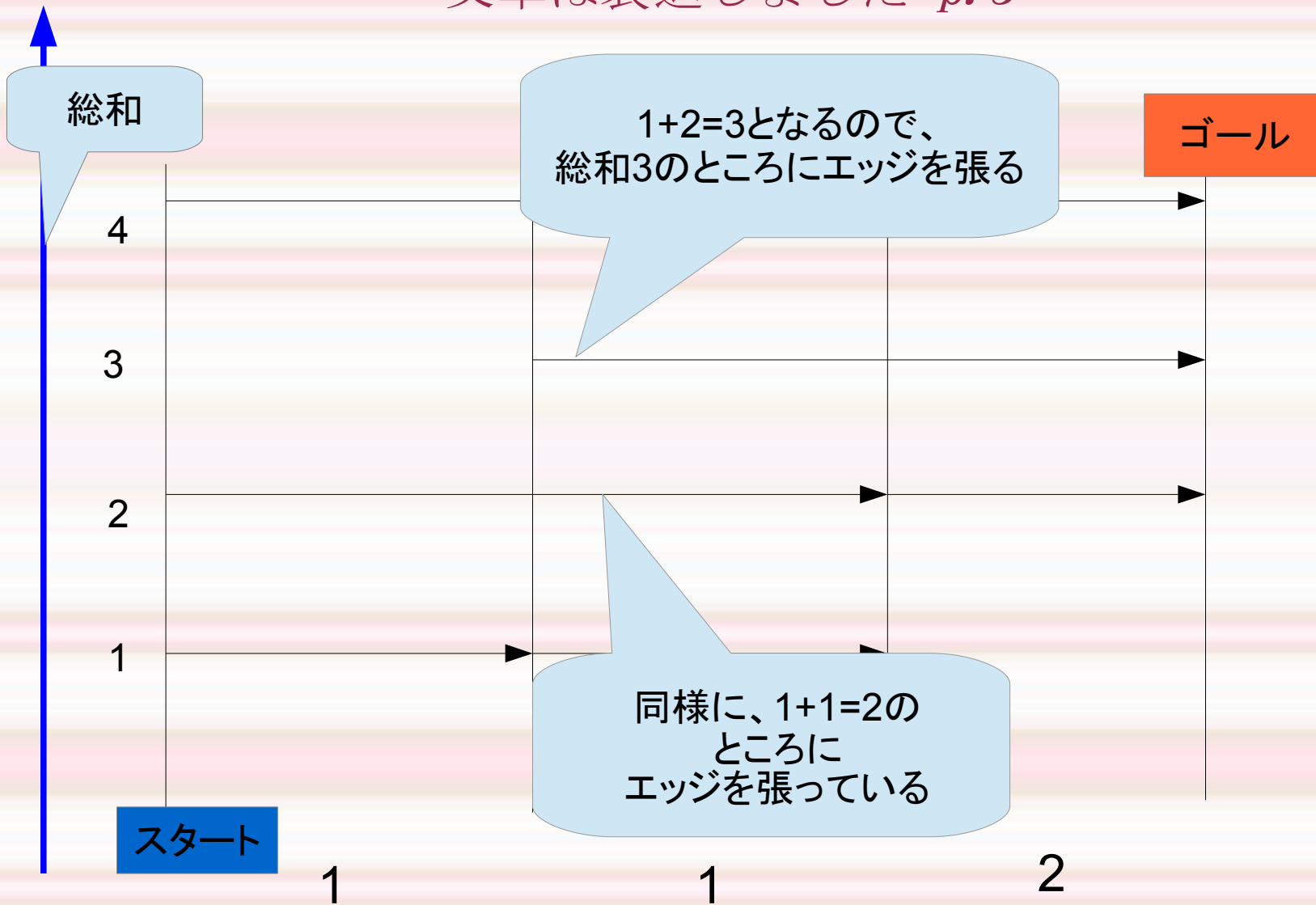
考えた
い数列



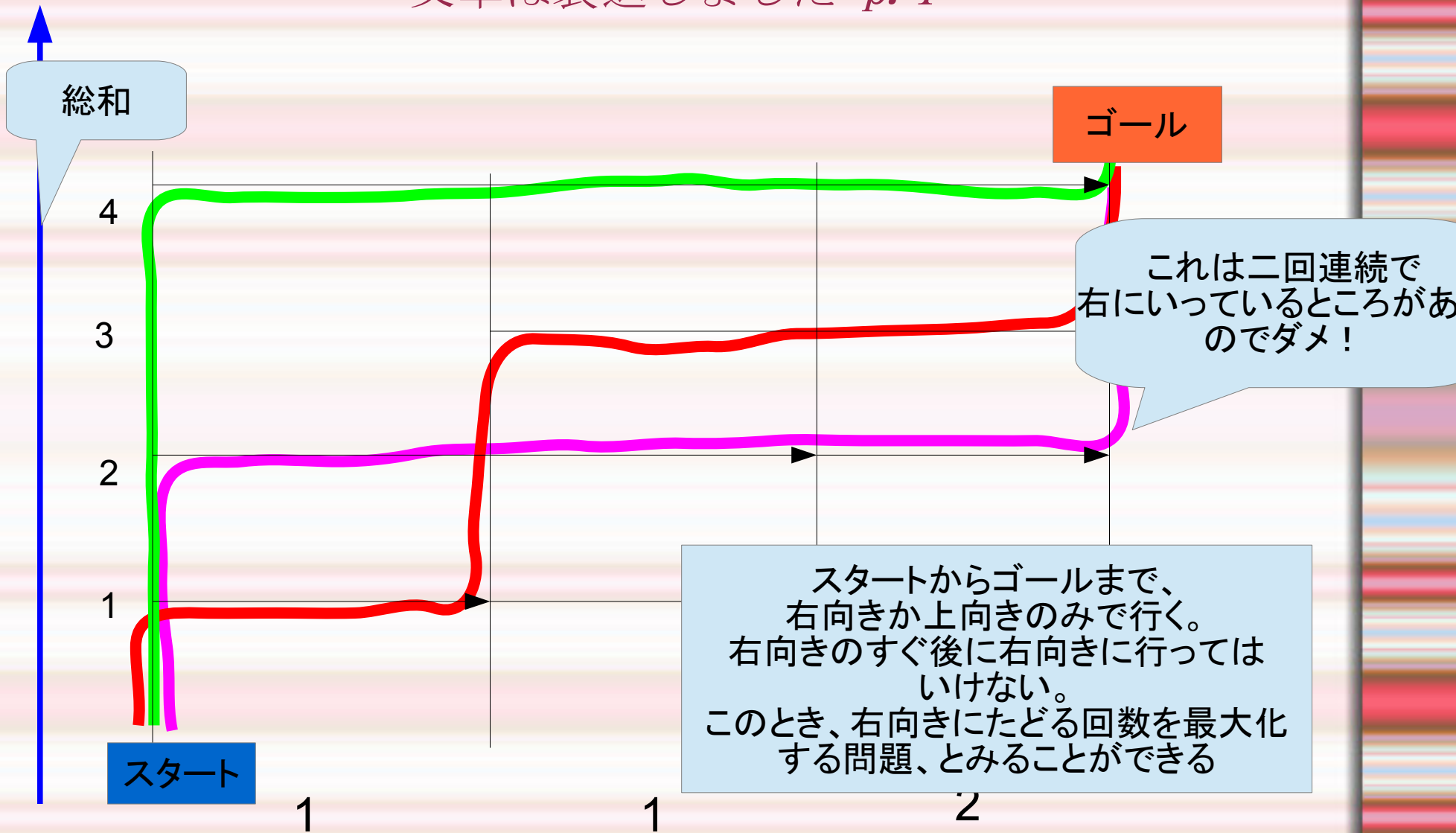
文章は衰退しました p. 2



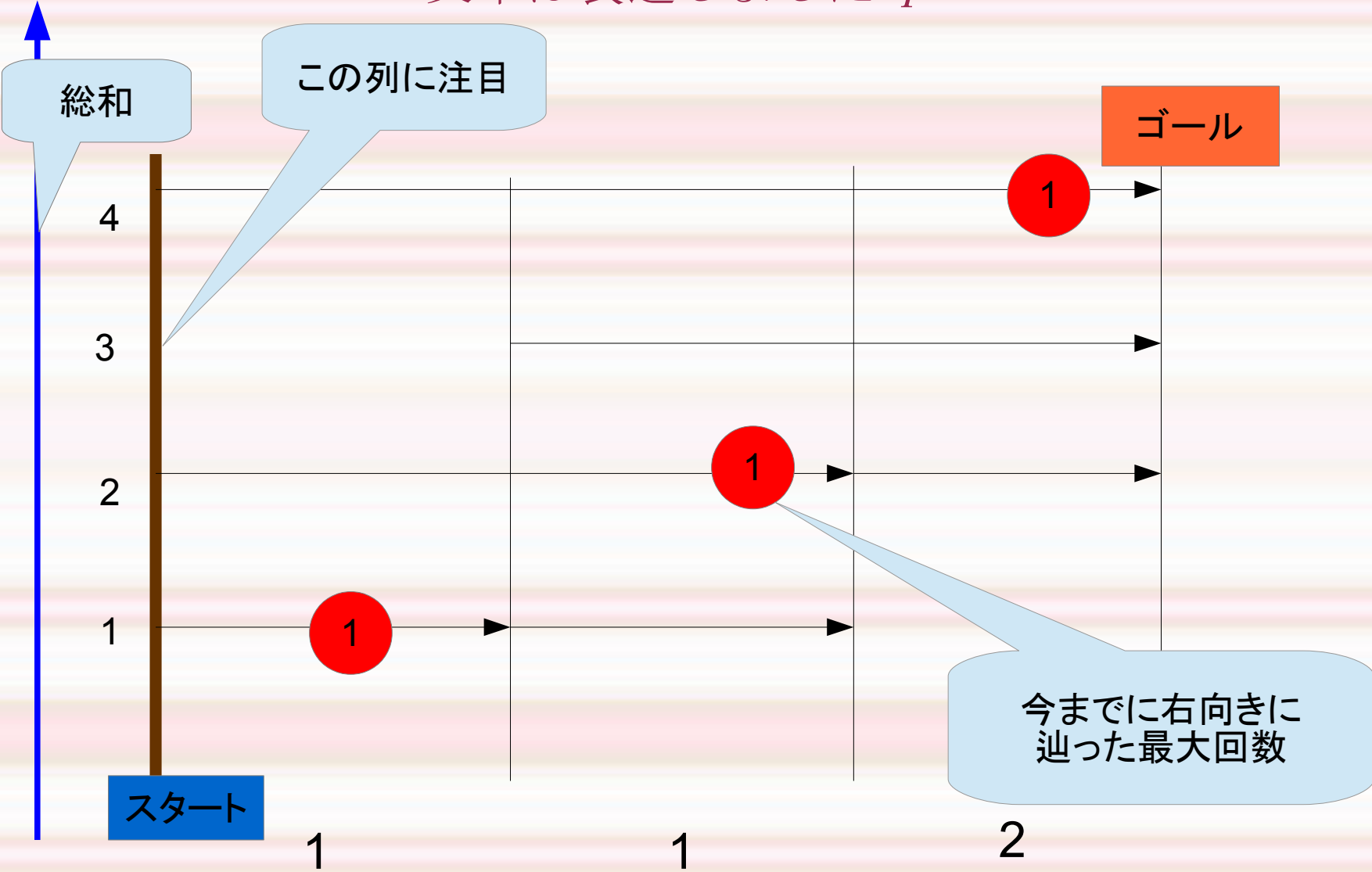
文章は衰退しました p. 3



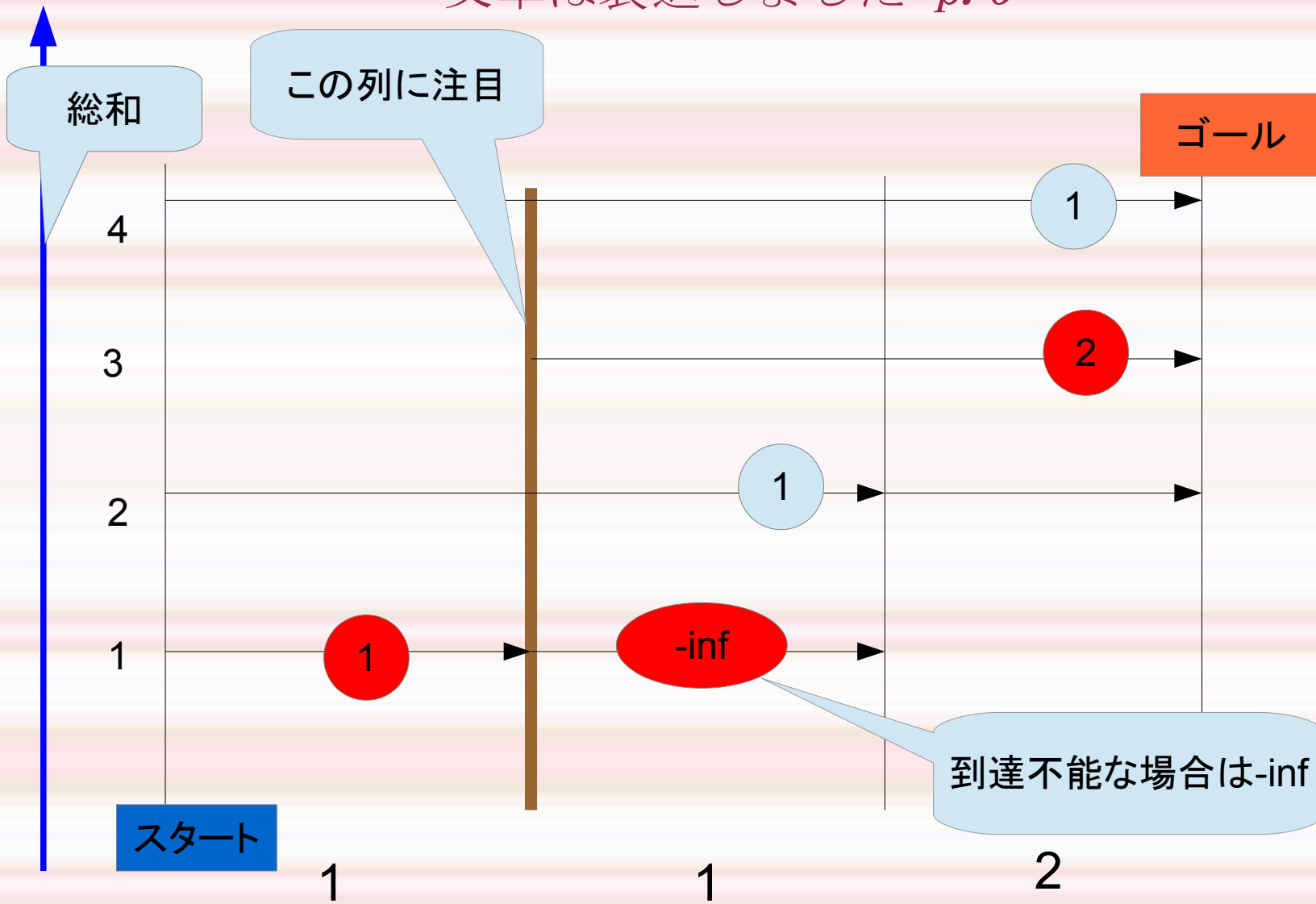
文章は衰退しました p. 4



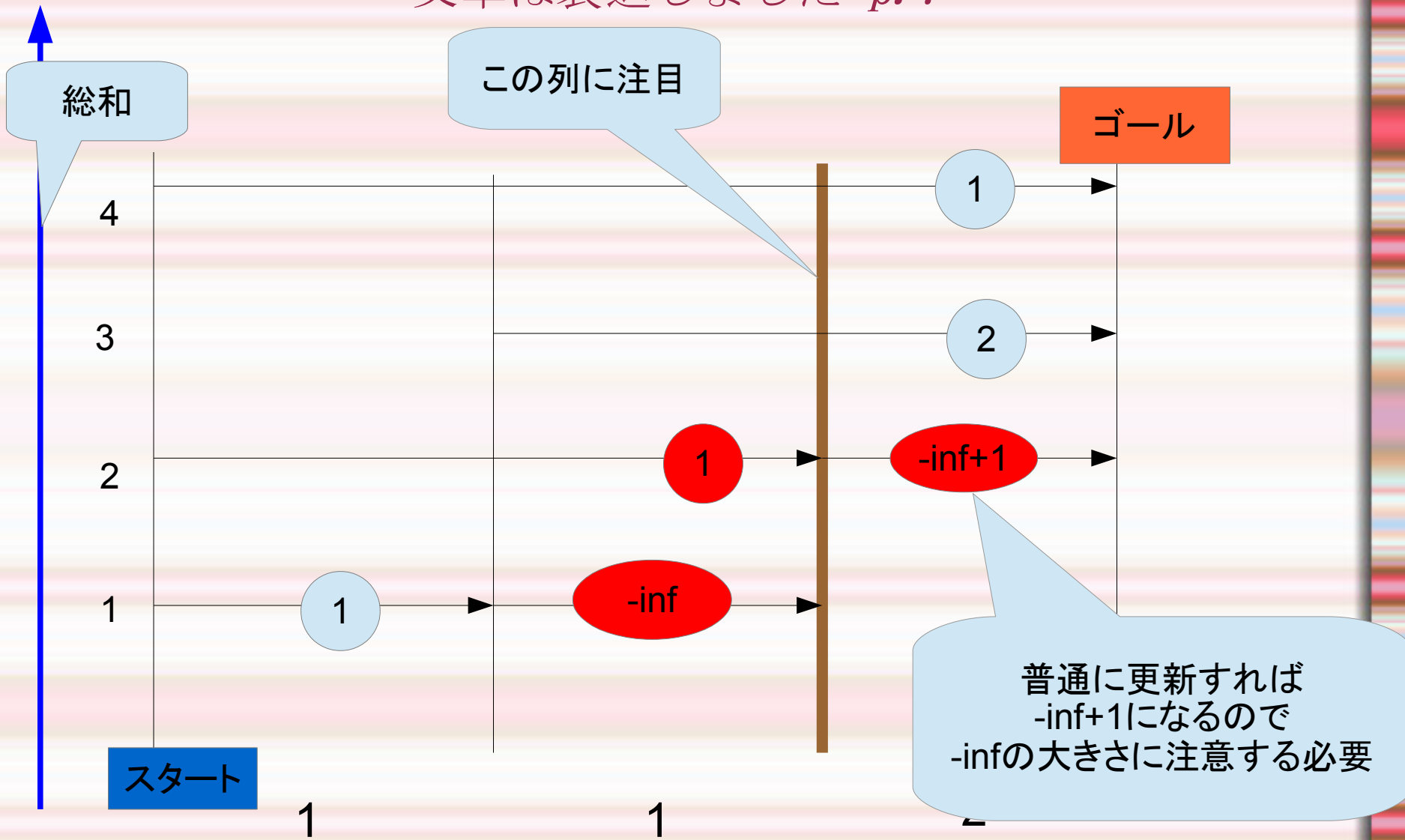
文章は衰退しました p.5



文章は衰退しました p. 6



文章は衰退しました p. 7



考察6 (想定解 p. 4)

◆ 注意事項

- ソートする際に、同じ値だった場合は、一つ後側を小さいようにとること。そうでないと、更新順がおかしくなってしまう (想定誤答)
- long longに対応してないと死にます。
- 負の数の事も忘れないで。

考察7 (想定解 p. 5)

◆ 計算量の解析

- ソート: $(n \log n)$, なめる $O(n)$
- それを n 個だけやりたい
- 最大値を求めた後の復元Phaseもちゃんと手繰れるようにしておけば $O(n)$ でおわる

- $O(n^2 \log n)$ で解ける!!!!(想定解法)

考察8 (別解)

- ◆ 作問者が見逃していた解
 - SpecialThanks : kyuridenamida
- ◆ SegmentTreeを使う
 - n個のsegment treeをもっておく
 - 範囲つきで最良のものを選ぶところをsegment treeで行う
 - ML:200MB であり、結構ギリギリになる可能性が高いのでshortintで持つとよさそう
- ◆ こちらの解法のほうが思いつきやすいのかなと思いました

*Judge*解詳細

- ◆ 岸本 :92行 (C++)
- ◆ 河田 :69行 (C++)

提出状況

- ◆ AC/SUB : 2/43
- ◆ FA : rng_58 (204min)
- ◆ rng_58さんは想定解法そのまま
- ◆ watashiさんは別解だったように思います