

Sliding Window Aggregation

**Keiya Sato
Yokoyama Lab, M1**

May 02, 2023

目次

1. Sliding Window Aggregation とは
2. しゃくとり法との関係と代数的構造
3. 2-Stack Queue
 1. push
 2. pop
 3. 時間計算量
4. SWAG
 1. 実装
 2. 計算量
5. 例題

Sliding Window Aggregation

- 一言で言うと「fold 可能な Queue」
 - Queue の min/max, gcd/lcm, sum などが $O(1)$ で計算可能
- 略して SWAG と呼ばれる
 - 実際はこれはアルゴリズムの名前
 - アカデミック的には Foldable-Queue?
- 2-stack queue と呼ばれるテクニックで実装される
 - 個人的に一番面白いポイント

SWAG で可能な操作

- 数列 a_1, a_2, \dots, a_N と演算 $+$ からなる半群を扱うデータ構造

操作	詳細	計算量
push	Queue の末尾に要素を追加	$\theta(1)$
pop	Queue の先頭の要素を取り出す	償却 $\theta(1)$
fold	$a_1 + a_2 + \dots + a_N$ を計算する	$\theta(1)$

【fold について】

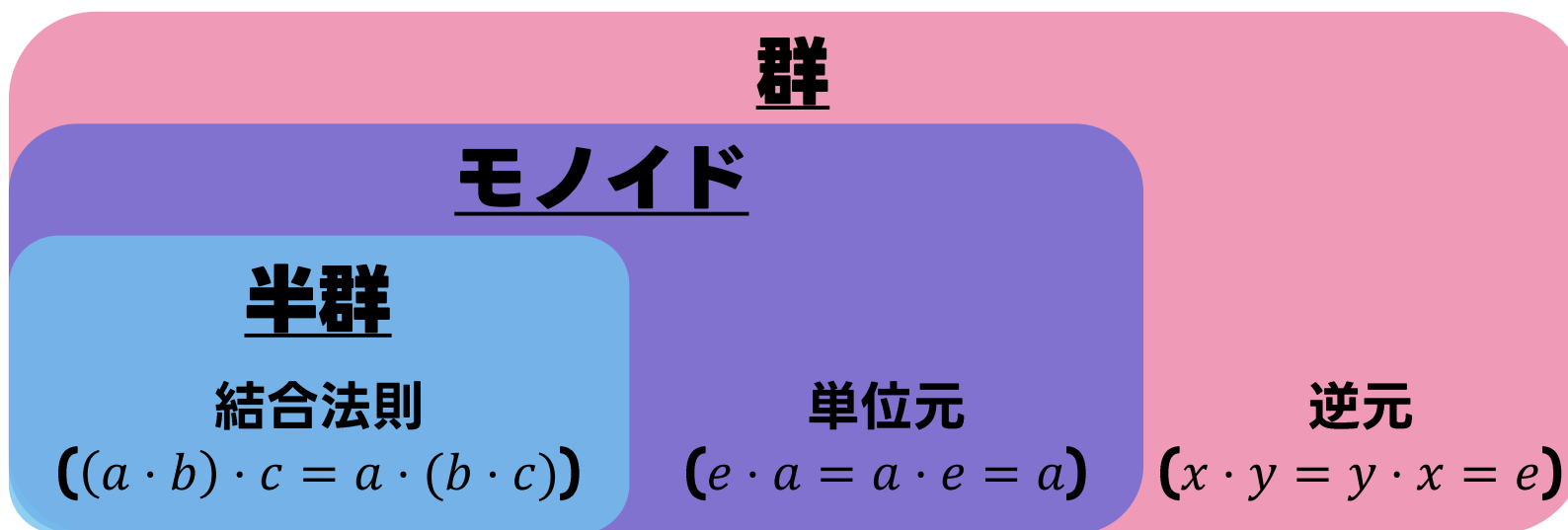
- Queue 内の要素を a_1, a_2, \dots, a_N として $a_1 + a_2 + \dots + a_N$ を計算する
- 可能な操作
 - sum, min, max, gcd, lcm, 一次関数の合成, etc...

しゃくとり法との関係

- しゃくとり法では区間の右端を加算しながら左端を減算することで区間和を計算
 - 右端を push, 左端を pop しながら fold するのと同値
- SWAG はしゃくとり法で利用できる
 - 競プロにおける主要な使い道
- 通常のしゃくとり法よりも対応できる問題が広い

群・モノイド・半群の関係

集合 G と G 上の演算 \cdot に対し, 以下が成り立つ

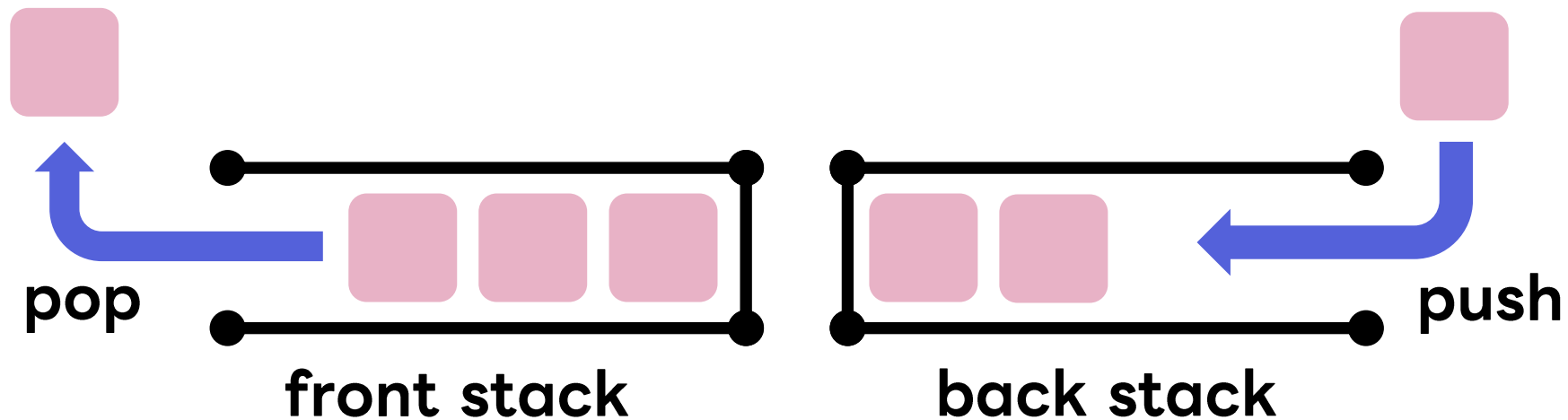


SWAG が扱う代数的構造

- SWAG は半群を扱う
 - しゃくとり法が扱うのは群
- しゃくとり法で扱える演算は全て SWAG で代替可能
- しゃくとり法が扱える演算の例
 - 区間和, 区間積, 行列積, etc...
- SWAG が扱える演算の例
 - 最大公約数, 最小公倍数, min, max, etc...

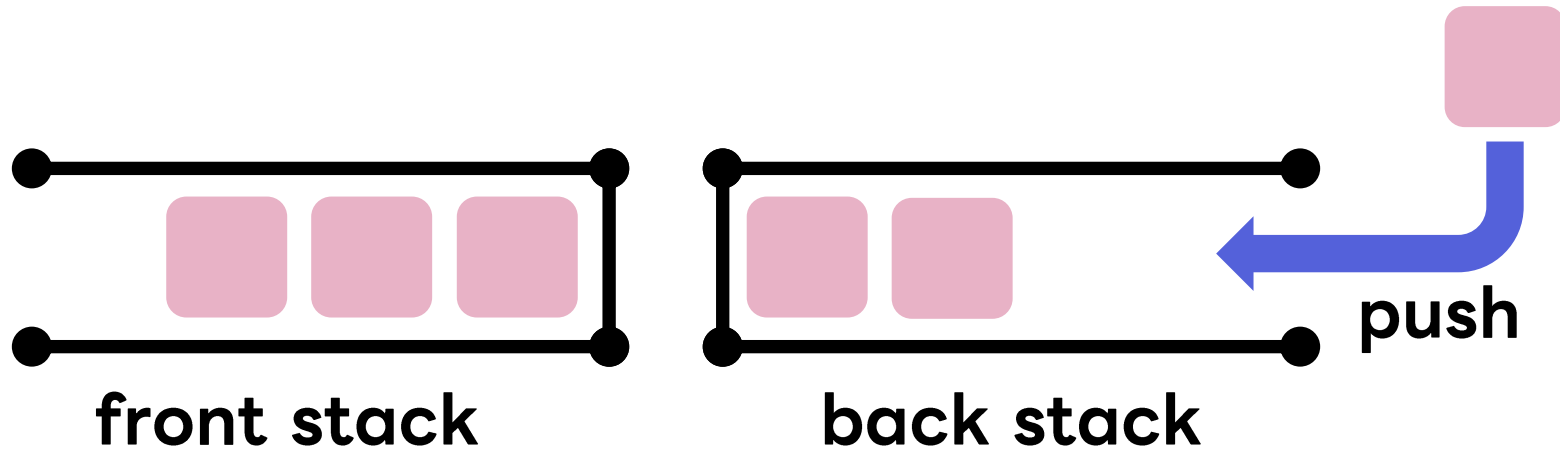
2-Stack Queue

- 2 本の Stack で Queue を再現する技法
- push を $\theta(1)$, pop を償却 $\theta(1)$ で行うことが可能



push

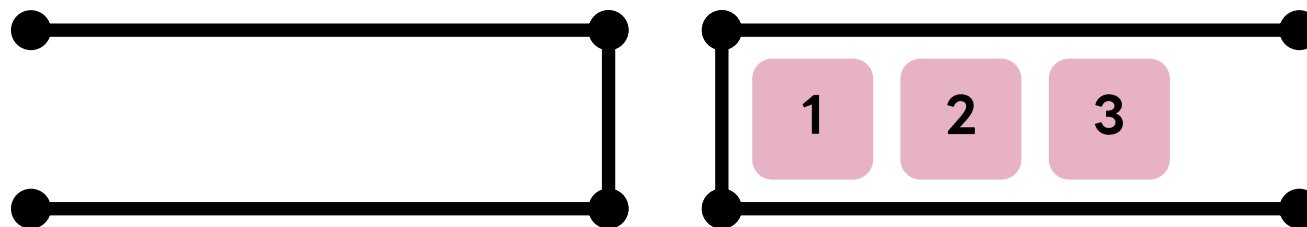
- back stack に要素を追加するだけ



pop

front stack が空の場合

- back stack の要素を全て front stack に移動させる
- front stack の先頭の要素を削除する



時間計算量

push

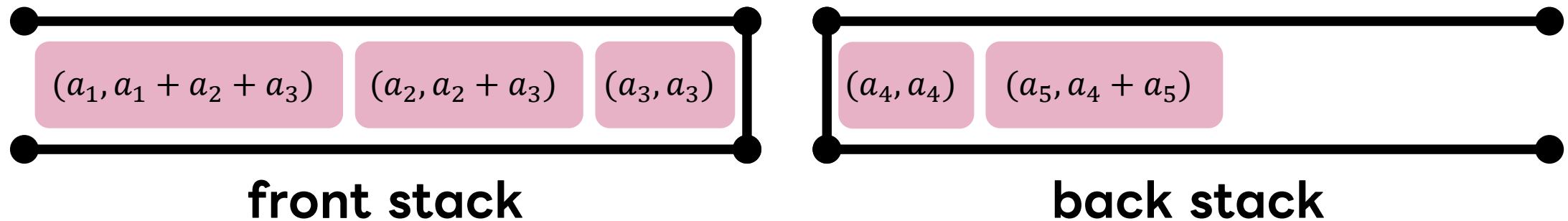
- ・ 追加するだけなので $\theta(1)$

pop

- ・ 要素の削除は $\theta(1)$
- ・ 要素の移動は $\theta(n)$ かかるが, 要素 1 つにつきこの操作は高々 1 回しか行われない

SWAG の実装

- 要素の他に累積和も一緒に保持する
 - （要素, 累積和）のペアを管理
- 各 stack の先頭の累積和を結合することで全体の総和を計算
 - 下図なら $(a_1 + a_2 + a_3) + (a_4 + a_5)$



正確な計算量

- push, pop, fold を $\theta(1)$ と紹介したが, これは半群の演算にかかる計算量を考慮していない
- 各演算にかかる計算量を $\theta(X)$ とすると, 各計算量は以下のようになる

操作	詳細	計算量
push	Queue の末尾に要素を追加	$\theta(X)$
pop	Queue の先頭の要素を取り出す	償却 $\theta(X)$
fold	$a_1 + a_2 + \cdots + a_N$ を計算する	$\theta(X)$

余談

- 同様に fold 可能な stack や deque を作成することもできる
 - stack は back stack のみを利用するだけ
 - deque も 2 本の stack を利用して再現可能
- min 演算に関しては「スライド最小値」という別のアルゴリズムが存在する
 - 競プロ的には SWAG より有名かも

例題1: Queue Operate All Composite

- Library Checker の問題

- 自作ライブラリの verify ができる
- 開発者は yosupo さん

- 一次関数の合成は半群を成す

- 競プロで有名な知識
- $f(x) = ax + b, g(x) = cx + d$ として $f(g(x)) = acx + ad + b$

例題2: ABC298 D – Writing a Numeral

- 文字列 S に対し以下の操作を行う
 1. 末尾に数字を追加する
 2. 先頭の数字を削除する
 3. 十進数の整数と見なして 998244353 で割った余りを計算する
- 上手いこと演算を定義すると SWAG に載る
 - 想定解の方が楽かも

例題3: JAG Summer Camp 2018 Day2

D - Knapsack And Queries

- 日本の競プロ界隈で SWAG が流行り始めた原因となった問題
- 難しいので解説は割愛
 - （そもそも解いてないです. すみません…）