

# Problem G: Chairs

作問: aoba

解説: がっちょ君

# 問題概要

- 番号 $1 \sim N$ の $N$ 個の椅子がある
- ID $1 \sim N$ の $N$ 人がいて、人 $i$ は椅子 $p_i$ に座りたい
- 人々はIDが小さい順に1列に並び、列が前の人から以下の操作を行う
  1. 椅子 $p_i$ が空席の場合はその席に座る
  2. 空席ではない場合は $p_i$ に1加算して、列の最後尾に並び直す( $N$ を超えた場合は1となる)
- 全ての人が入るまでこの操作を繰り返す
- 最終的にそれぞれの椅子に座っている人のIDを出力する
- 制約:  $1 \leq N \leq 10^5, 1 \leq p_i \leq N$

# ダメな解法

- 問題文通りにシミュレーション
  - 例えば、全ての初期位置が同じ場合は  $N \times (N + 1)/2$  回の操作をすることになり  $N = 10^5$  なので間に合わない

# ポイント

- 問題を「各椅子に対して人が並んでいる」と置き換える

# 解法1

- 累積和 + スタック

# 解法1

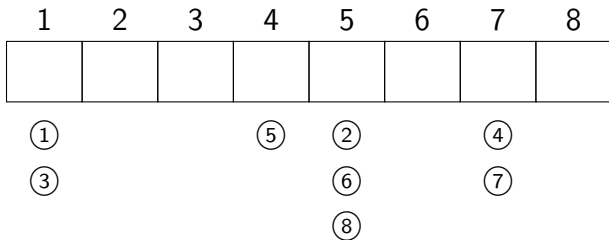
- 問題を置き換えた場合でもどこからどのような順序で処理を行うかを決める必要がある(円環するため)
- 適切な位置を選択することで円環を直線と考えることができる
- 適切な位置は次のように決められる
  1. 各椅子に対して初期値を-1として、人が並んでいる場合はその人数だけ加算して累積和を取る
  2. その中で最も値が小さい場所を $x$ とすると、 $x + 1$ が適切な位置となる( $x = N$ なら1)

# 解法1

- 適切な位置を選んだらその位置からスタックを使い、それぞれの椅子に誰が座るかを決定していく
  - $O(N)$ で解くことができる
- ※ 工夫すると累積和を使用しなくても解くことができる

# 解法1

$N = 8, p = \{1, 5, 1, 7, 4, 5, 7, 5\}$ の場合



1	-1	-1	0	2	-1	1	-1
---	----	----	---	---	----	---	----



# 解法1

累積和を取る

1	2	3	4	5	6	7	8

①

⑤

②

④

③

⑥

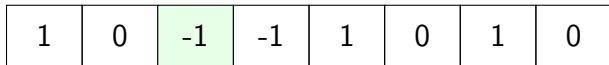
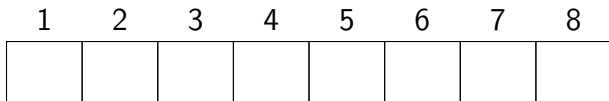
⑦

⑧

1	0	-1	-1	1	0	1	0
---	---	----	----	---	---	---	---

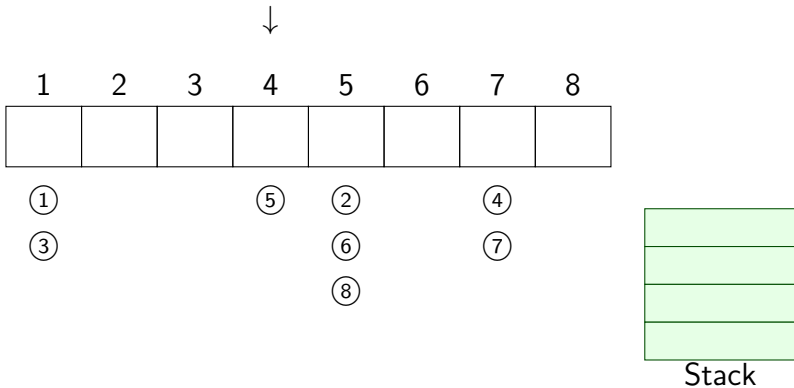
# 解法1

椅子3が最小 → 椅子4から操作を開始する

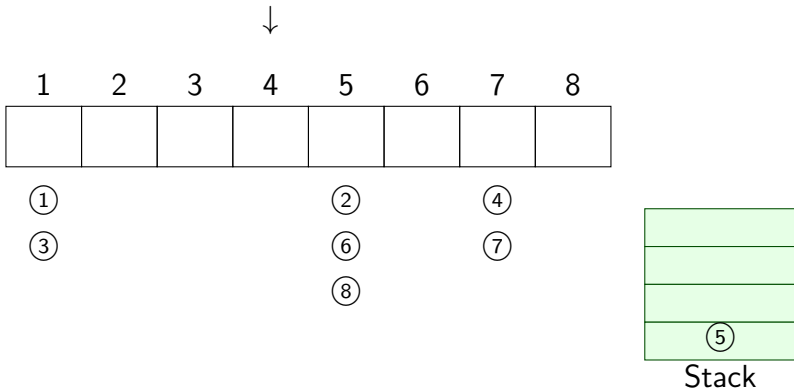


start→

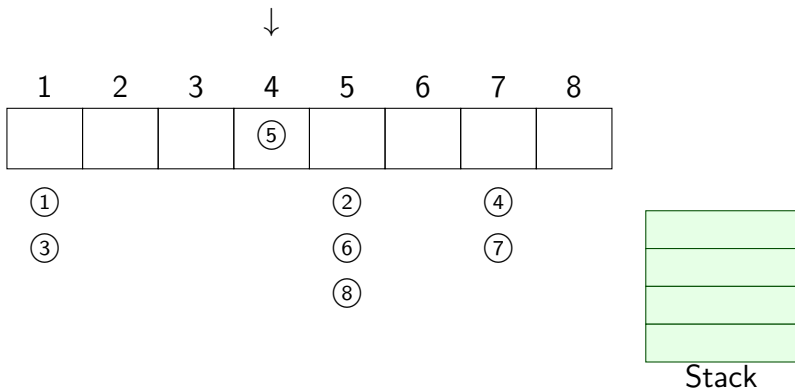
# 解法1



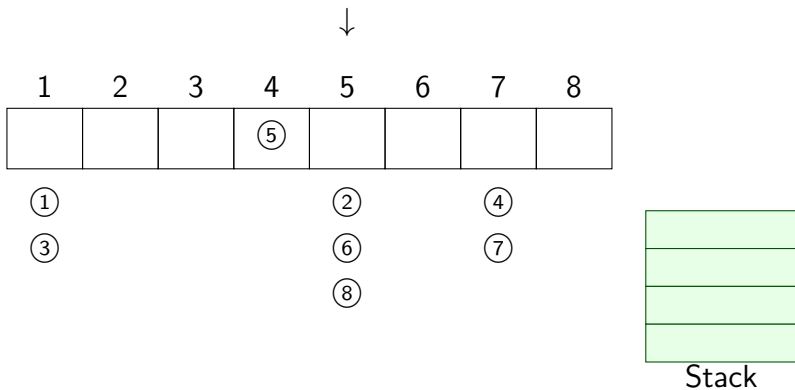
# 解法1



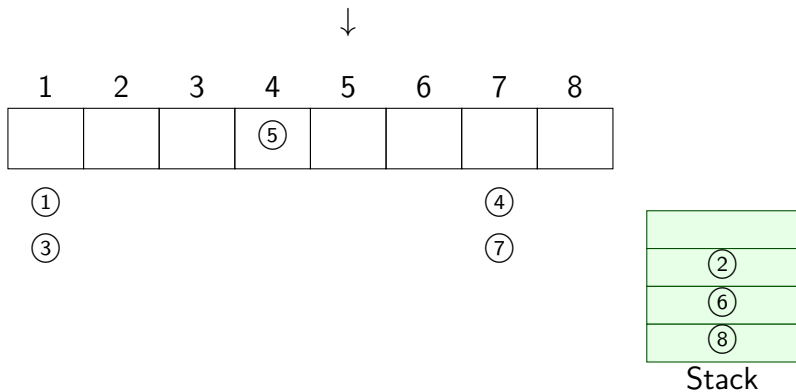
# 解法1



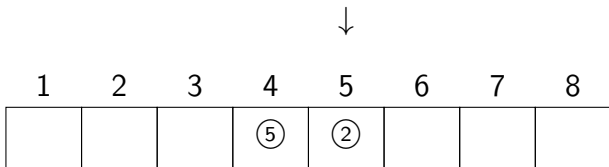
# 解法1



# 解法1



# 解法1

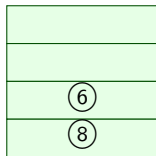


①

④

③

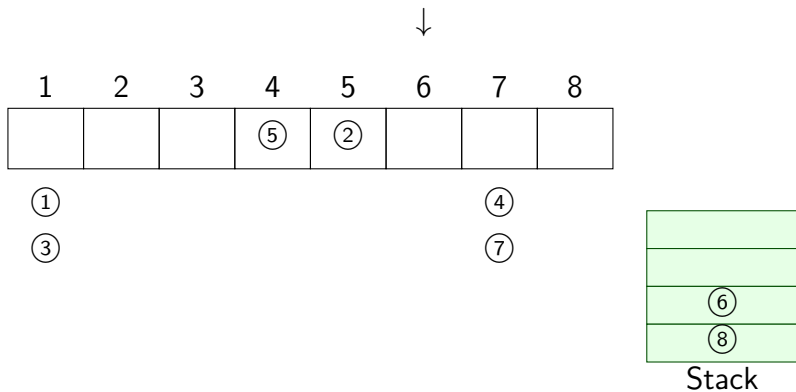
⑦



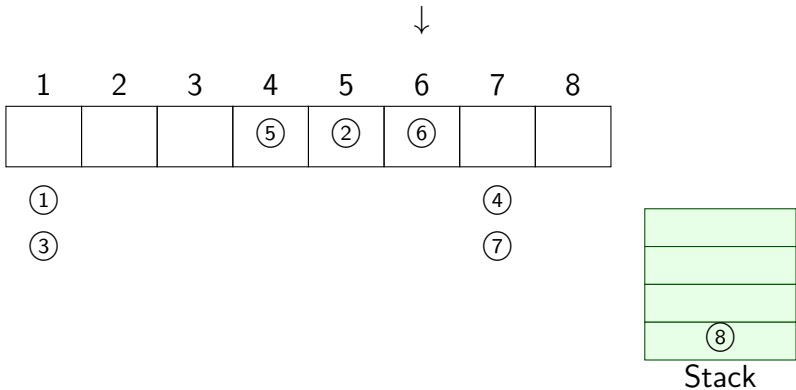
Stack



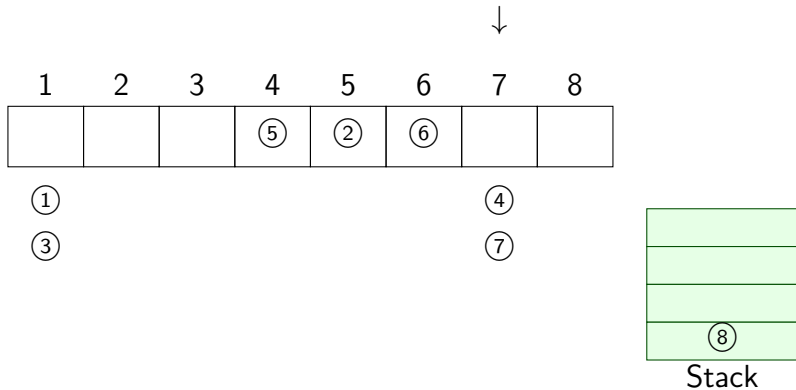
# 解法1



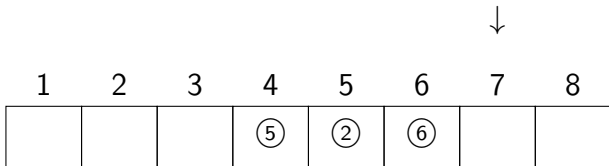
# 解法1



# 解法1

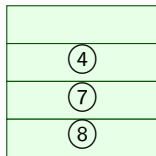


# 解法1



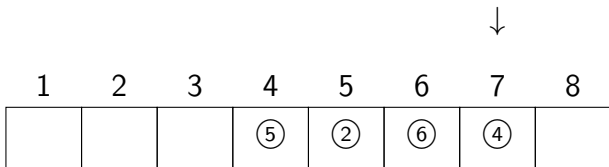
①

③



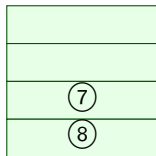
Stack

# 解法1



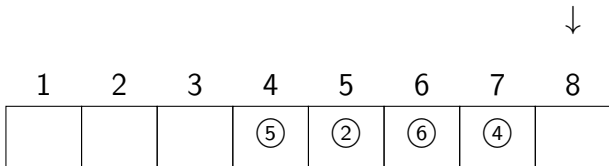
①

③



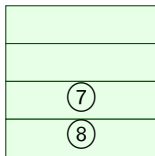
Stack

# 解法1



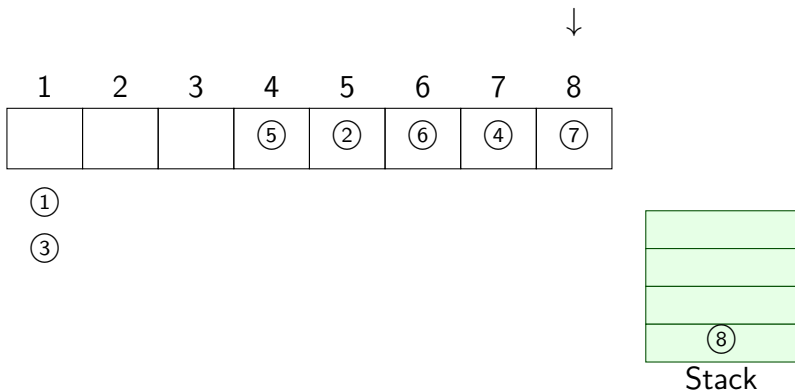
①

③

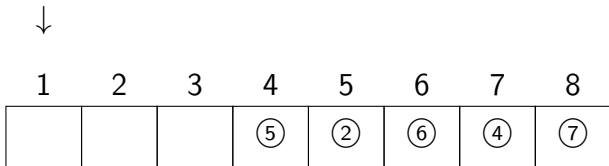


Stack

# 解法1

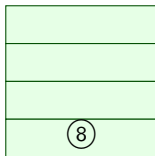


# 解法1



①

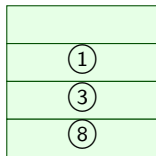
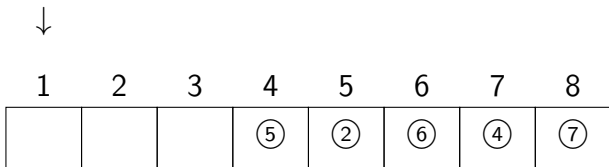
③



Stack

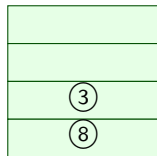
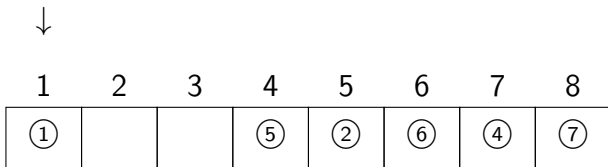


# 解法1



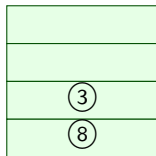
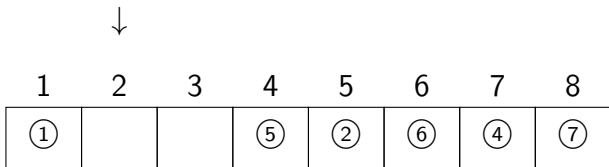
Stack

# 解法1



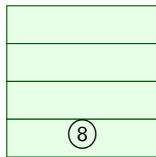
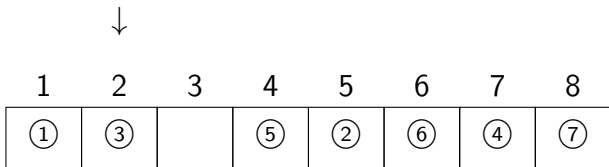
Stack

# 解法1



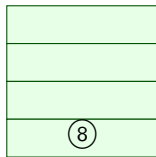
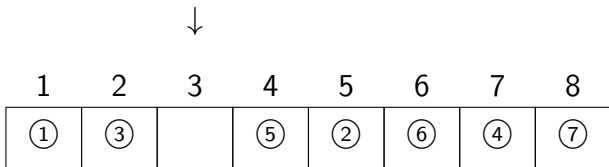
Stack

# 解法1



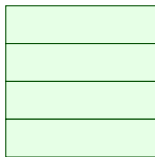
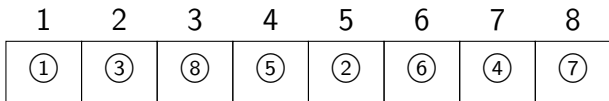
Stack

# 解法1



Stack

# 解法1



Stack

## 解法2

- シミュレーション
- 列ごとにまとめて動かしていく
- 任意の順序で列を動かしてOK

## 解法2

- 各列について以下の操作を行う必要がある
  1. 最寄りの空席の位置を探す
  2. 2つの列をマージする  
(一方の列をもう一方の列の後ろに挿入する)
- これらは例えばstd::setとstd::listを組み合わせると $O(\log N)$ で実現することができる
- $N$ 人についてこの操作を行うと $O(N \log N)$ となる



# 結果

- Onsite
  - First Submission: caffe チーム (113 min)
  - First Accepted: caffe チーム (113 min)
- Online
  - First Submission: ei1333 さん (49 min)
  - First Accepted: caffe チーム (113 min)
- Success Rate (Accepted / Submission)  
12.00 % (3 / 25)

# ジャッジ解

aoba	C++	57行
arrows	C++	49行
arrows	Java	38行
gacho	C++	25行
haji	C++	43行
kawa	C++	17行
kzyKT	C++	20行
sate	C++	36行
uku	C++	66行