

# 第8回のキーワード

1

## アルゴリズム関係

- スタック (stack)
- LIFO: Last In, First Out / 後入れ先出し (FILO: First In, Last Out)
- プッシュ (push)
- ポップ (pop)
- トップ (top)
- スタックポインタ (stack pointer)
- キュー / 待ち行列 (queue)

- FIFO: First In, First Out / 先入れ先出し
- エンキュー (enqueue)
- デキュー (dequeue)
- リングバッファ (ring buffer)
- $O(1)$

## Java関係

- $i = (i + 1) \% a.length$
- throw, throws

# スタックとキュー

2

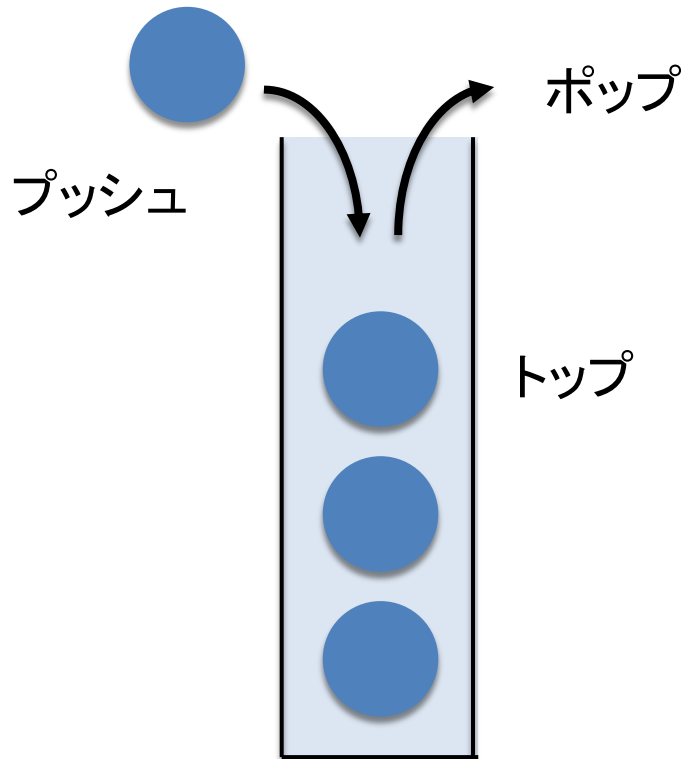
- スタック (stack)
  - ▣ 物を積むように、後から入れたデータから順に取り出せる「後入れ先だし」(LIFO)のデータ構造
  - ▣ スタックポインタを用いることで  $O(1)$  での処理が可能
  
- キュー (queue)
  - ▣ 行列に並ぶように、先に入れたデータから順に取り出せる「先入れ先出し」(FIFO)のデータ構造
  - ▣ リングバッファによって実現すると  $O(1)$  での処理が可能
  
- 両端キュー (double-ended deque)
  - ▣ データ列の両端から出し入れができるデータ構造
  - ▣ スタックとキューの機能をあわせ持つ

# スタックとキュー

3

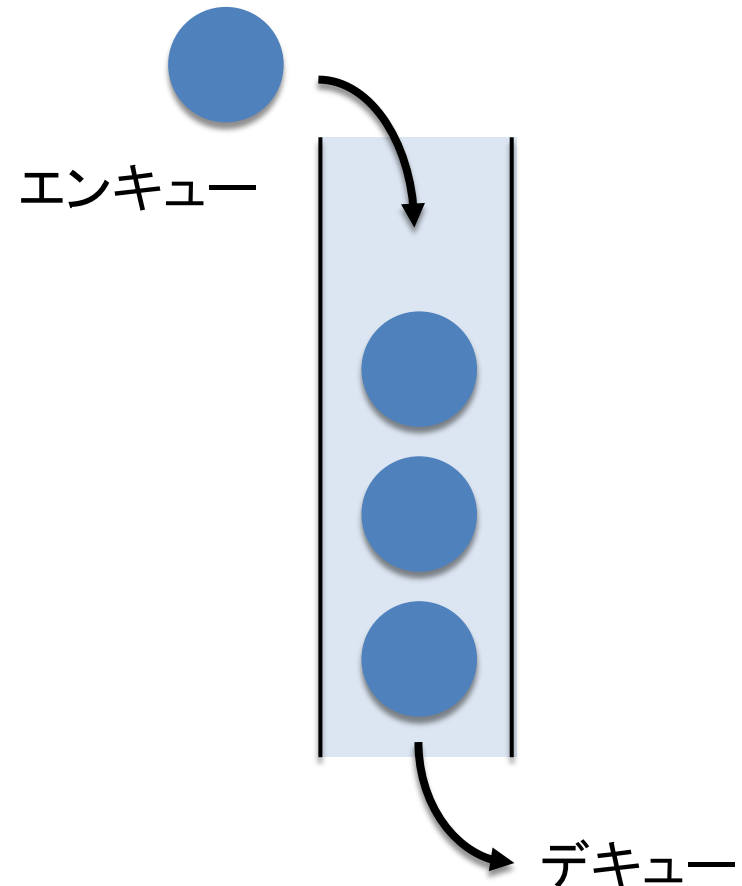
## □ スタック

- LIFO: 後入れ先出し



## □ キュー

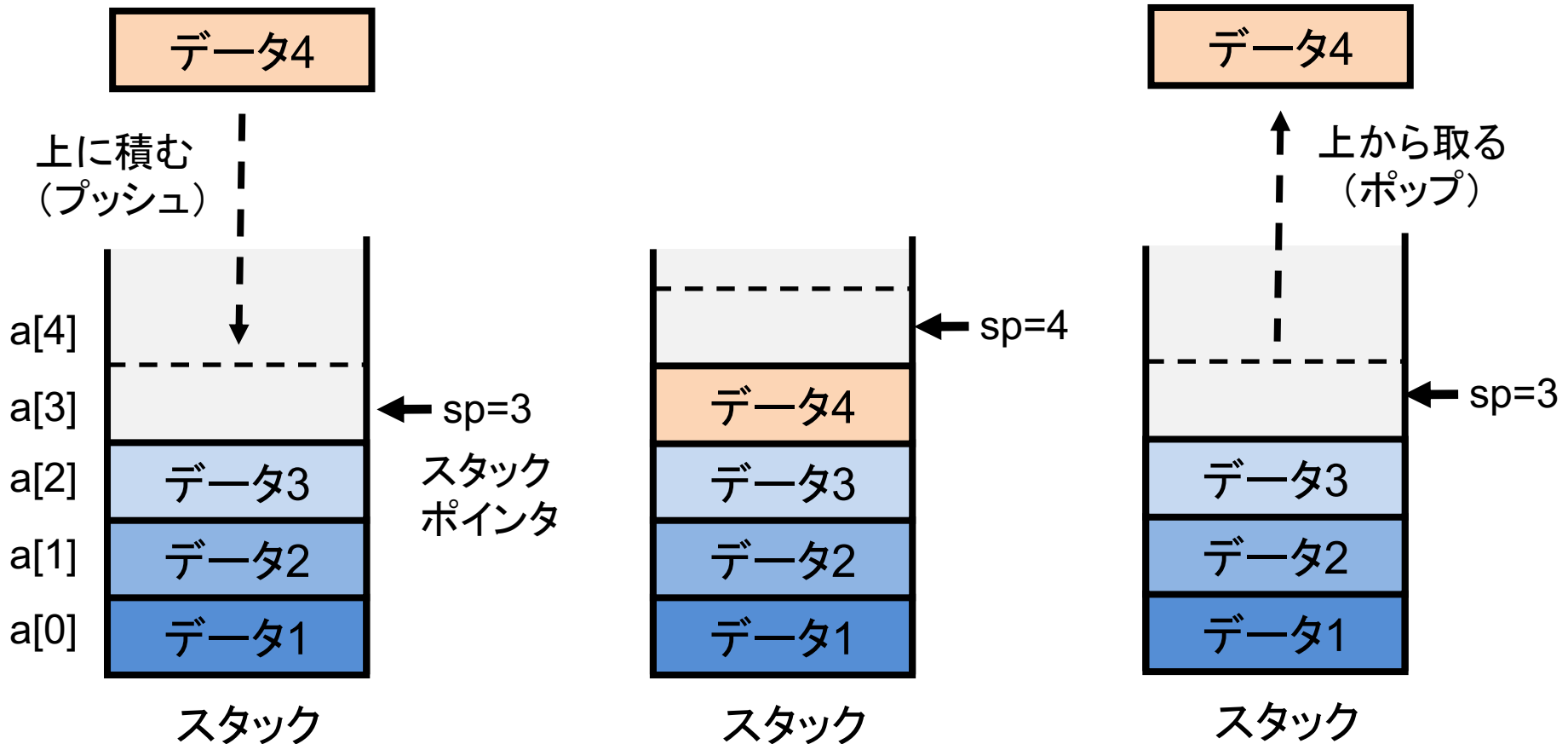
- FIFO: 先入れ先出し



# 配列によるスタック

4

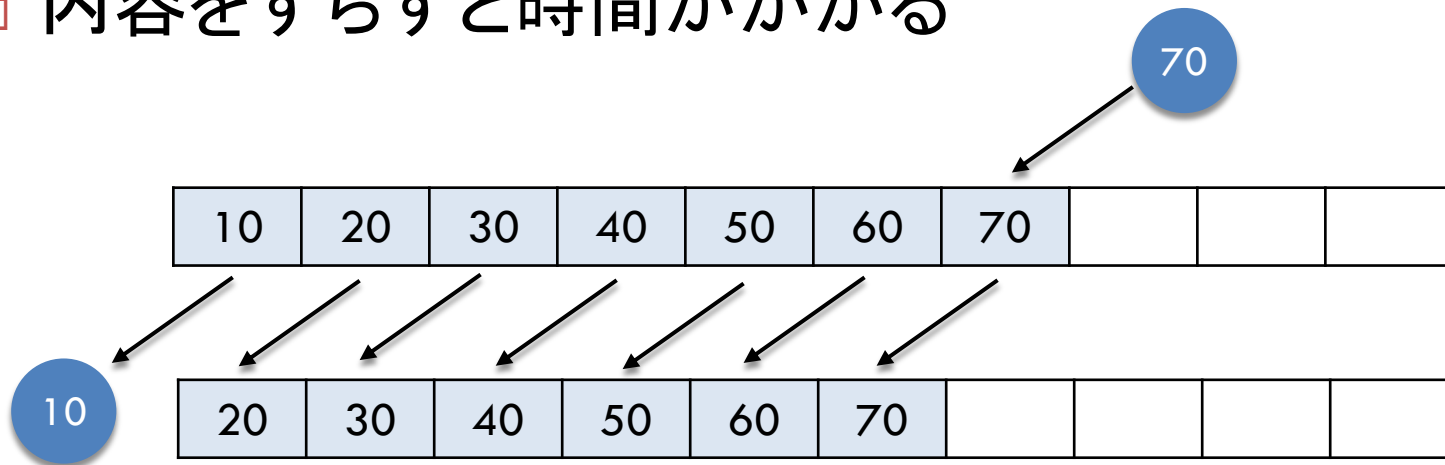
- スタックポインタ(整数)の増減で管理する



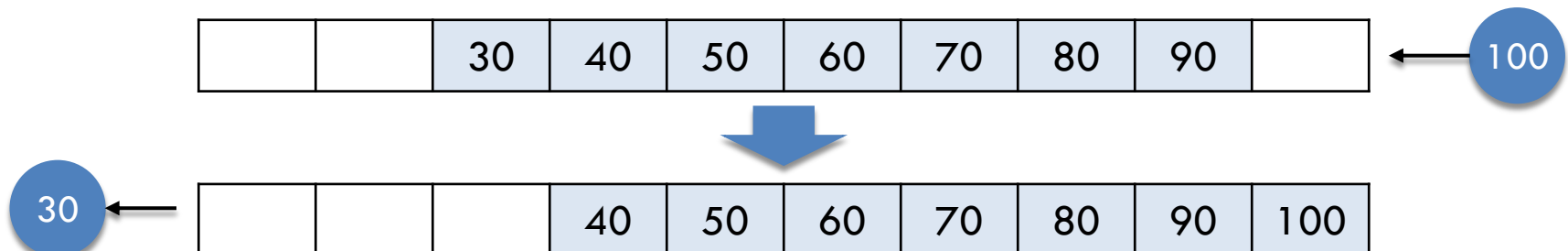
# キューを配列でどう実現するか

5

- 内容をずらすと時間がかかる



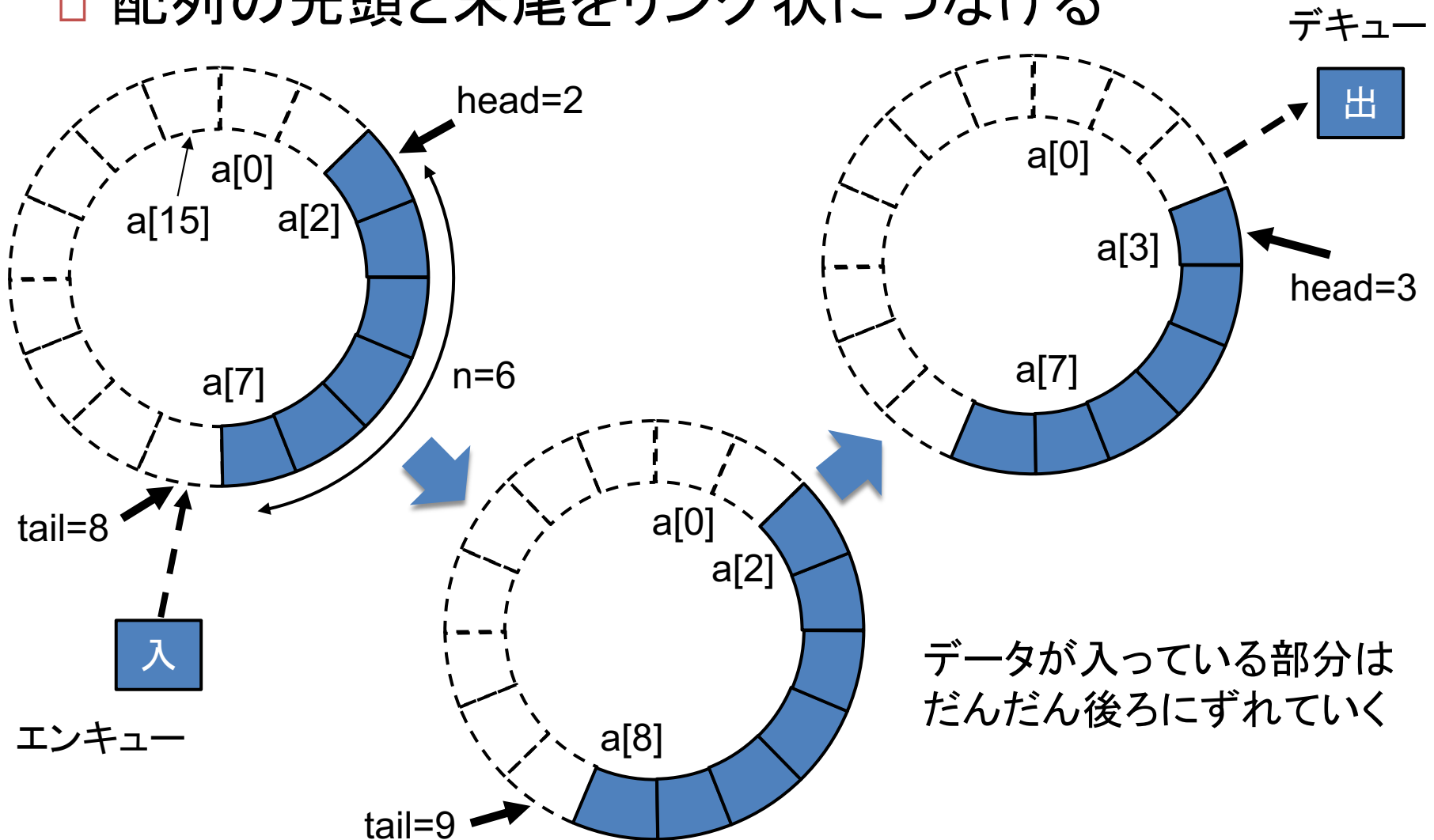
- ずらさないとすぐ満杯になる



# リングバッファ

6

## □ 配列の先頭と末尾をリング状につなげる



# リングバッファ

7

- 配列内で使用要素が循環する
  - 配列の終端まで埋まると、また先頭要素から使用していく
  - 剰余演算を使えば、添字の値を循環させることができる
  - 例) 4で割った余り: 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, ...

