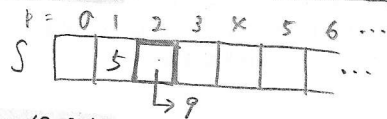
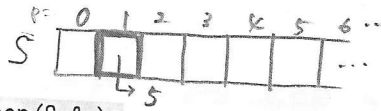


pop(S, &p);



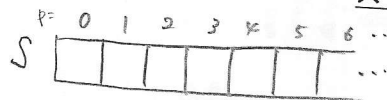
この時点での先頭である 9 を出力して削除
↓ p = 1 減算 先頭は S[1] の 5

pop(S, &p);



この時点での先頭である 5 を出力して削除
↓ p = 1 減算 p = 0

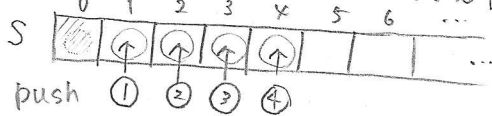
pop(S, &p);



スタックは空です。取り出す要素はありません。

S には何も入っていません。

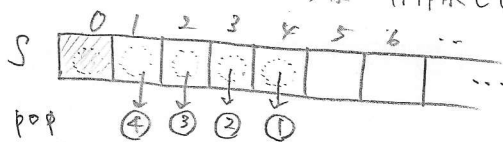
- 下の図のようにスタックは逆列 S[i] の i が小さいものから順に要素が入れられていき (push)



push ① ② ③ ④

→ push の流れ

- 要素が入れられている S[i] 中で i が一番大きいものを先頭として、先頭の要素を出力して今まで入っていたその要素を削除していく (pop) データ構造である。

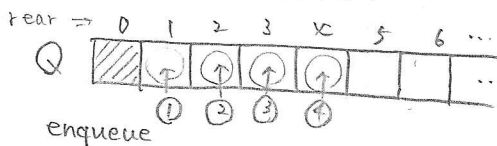


pop ④ ③ ② ①

← pop の流れ

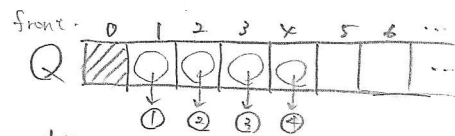
- このような図を使えば、スタックの動きがよくわかる。

- ここで、キューに関して



enqueue

→ enqueue の流れ



dequeue

→ dequeue の流れ

- 上の図を用いて理解をすすめると、

enqueue により $Q[\text{rear}]$ に要素が入ると rear に 1 加算することをつけて、 $Q[1], Q[2], \dots$ に要素を入れていくと、 rear は最初 0 にしておけば、一番左端に入れられた要素を取り出す dequeue は $\text{front} = 1$ を用意して $\text{front} \leq \text{rear}$ とすれば $Q[\text{front}]$ を表示し front に 1 加算すればよいとわかる。

スタックとキューでは要素を取り出すときの向きが反対となる。