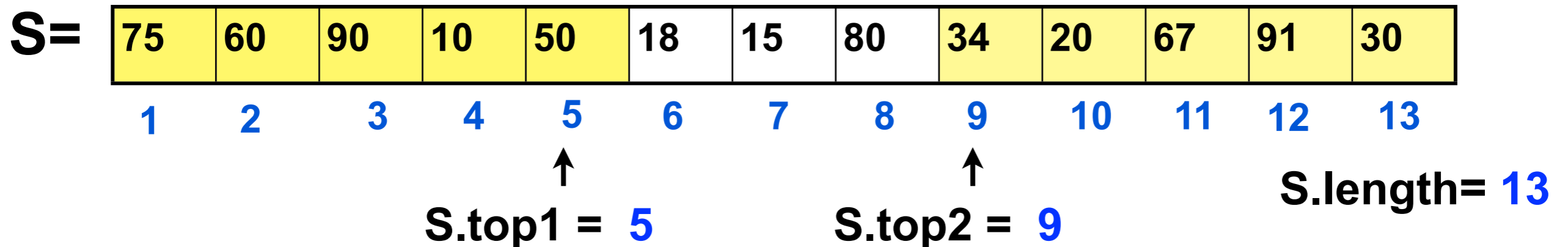


# Esercizio pila

(Es. 10.1-2 di [CLRS10]) Come si possono implementare due pile su un array di  $n$  elementi, in modo tale che non si verifichi la condizione di pila piena a meno che il numero totale degli elementi delle pile sia  $n$ . Le operazioni push e pop delle due pile devono essere eseguite in tempo costante.



# Esercizio coda

**Es. 10.1-6.**

**Mostrate come implementare una pila usando due code. Bisogna definire le operazioni Top, push e pop usando enqueue e dequeue.**

**Si analizzi la complessità asintotica delle operazioni push e pop in questa nuova implementazione.**

# Sol 1. Esercizio coda

Nella coda Q1 gli elementi della pila sono inseriti in modo tale che l'ultimo accodato sia l'elemento in cima quindi

`push(Q1,x) = enqueue(Q1,x)`

La coda Q2 serve per realizzare l'operazione pop:

`pop(Q1)`

**input:** Q1 è una coda

**prec:** Q1 è utilizzata per implementare una pila in cui l'elemento in cima è l'ultimo accodato.

**post:** Q1 contiene gli elementi che aveva tranne l'ultimo accodato

`x = dequeue(Q1)`

**while** not Queue-empty(Q1) **do**

**invariante:** tutti i precedenti di x nella coda Q1 sono accodati nella coda Q2

`y = x`

`enqueue(Q2,y)`

`x = dequeue(Q1)`

**while** not Queue-empty(Q2) **do**

`enqueue(Q1,dequeue(Q2))`

**return** x

# Sol.1 Esercizio coda

Nella coda Q1 gli elementi della pila sono inseriti in modo tale che l'ultimo accodato sia l'elemento in cima quindi

`push(Q1,x) = enqueue(Q1,x)`

La coda Q2 serve per realizzare l'operazione top:

`top(Q1)`

**input:** Q1 è una coda

**prec:** Q1 è utilizzata per implementare una pila in cui l'elemento in cima è l'ultimo accodato.

**post:** Q1 non è cambiata e l'elemento in coda è restituito all'ambiente

`x = dequeue(Q1)`

**while** not Queue-empty(Q1) **do**

**invariante:** tutti i precedenti di x nella coda Q1 sono accodati nella coda Q2

`y = x`

`enqueue(Q2,y)`

`x = dequeue(Q1)`

**while** not Queue-empty(Q2) **do**

`enqueue(Q1,dequeue(Q2))`

`enqueue(Q1,x)`

**return** x

# Sol 2. Esercizio coda

Nella coda Q1 gli elementi della pila sono inseriti in modo tale che il primo inserito sia l'elemento in cima quindi

`Q1.top = Q1.head`

`pop(Q1) = dequeue(Q1)`

La coda Q2 serve per realizzare l'operazione push:

`push(Q1,x)`

**input:** Q1 è una coda e x un elemento dello stesso tipo di quelli in Q1

**prec:** Q1 è utilizzata per implementare una pila in cui l'elemento in cima è quello in testa alla coda.

**post:** Q1 contiene gli elementi di che aveva con x in testa

`enqueue(Q2,x)`

**while** not Queue-empty(Q1) **do**

**invariante:** x e gli elementi che precedono x nella coda Q1 sono accodati nella coda Q2

`x = dequeue(Q1)`

`enqueue(Q2,x)`

**while** not Queue-empty(Q2) **do**

`enqueue(Q1,dequeue(Q2))`

**return**