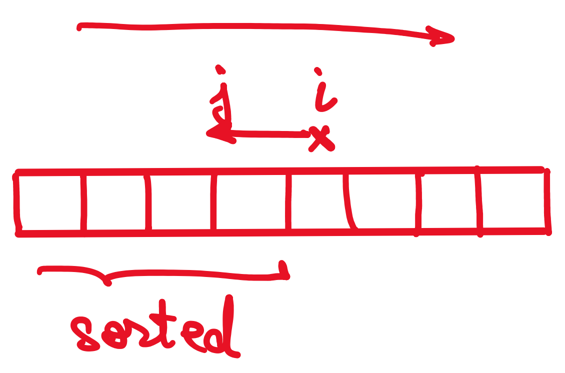
Insertion sort

Monday, 3 February 2025 11:36



loop invariant:

the subarray arr[0 → i−1] is always sorted

Loop invariants allow us to prove if an algorithm is correct.

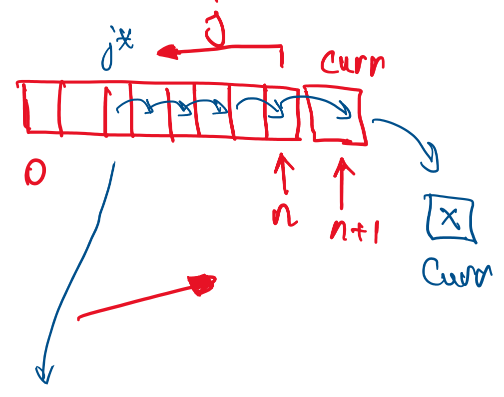
To use a loop invariant for proof:

- initialization: the invariant is true prior to the 1st loop
- maintenance: $P(n) \to P(n+1)$
  if true prior to an iteration, it is also true after the iteration
- termination: $P(N)$ true implies the algorithm corectness (check the reason why it got terminated → $N_{(cond)}$)

For insertion sort

- init. $i=1 \Rightarrow A[0 \to 1]$ has 1 element, which is also sorted (unique permutation)
- Maintenance $P(n) \to P(n+1)$

  Assume $P(n)$: $A[0 \to n-1]$ is sorted



$arr[j^*] \leq curr < arr[j^*+1]$ for the first time

$arr[j^*+1] \leftarrow arr[j^*]$

⇓

At loop term:

$\widehat{arr}[j^*+2] > curr \geq \underbrace{\widehat{arr}[j^*]}_{\text{correct}} = \underbrace{\widehat{arr}[j^*+1]}_{\text{shifted 1 pos}}$

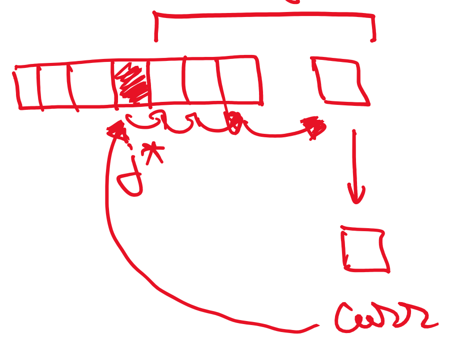
By assigning $arr[j^*+1] \leftarrow curr$:

- all numbers $arr[0 \to n+1]$ present in $\widehat{arr}[0 \to n+1]$

Use $P(n)$:

- $arr[j^*+2] > arr[j^*+1] \geq arr[j^*]$

  $\hat{a}[j^*+1 \to n+1] = a[j^* \to n]$ Sorted!



$\hat{a}[:j^*-1] = a[:j^*-1]$ Sorted!

$\Rightarrow \hat{a}[1:n]$ sorted, contains all numbers in $a[1:n]$ $\Rightarrow$ $P(n+1)$ true

$\therefore P(n) \to P(n+1)$

- Termination: $n = \text{len(nums)} = N$

  $P(N)$ true $\Rightarrow \hat{a}[1:N]$ contains all numbers in N, $\hat{a}_1 \leq \hat{a}_2 \leq \dots \leq \hat{a}_N$ $\Rightarrow$ Array is sorted