CSCI 241

Scott Wehrwein

Insertion Sort
Selection Sort

Goals

- Be able to execute insertion sort and selection sort on paper.

- Be prepared to implement insertion sort and selection sort.

Why are we talking about sorting, I thought this was a data structures course?

	(not sorted) PilingDrawer	(sorted) FilingDrawer
Add	O(1)	O(N)
Find	O(N)	O(log N)

Insertion Sort

Insert A[i] into the sorted sublist A[0..i-1].

Selection Sort

Find the smallest element in A[i..n] and place it at A[i].

Demo: https://visualgo.net/bn/sorting

Insertion Sort

Insert A[i] into the sorted sublist A[0..i-1].

i

Invariant: A | sorted | ? |

Selection Sort

Find the smallest element in A[i..n] and place it at A[i].

i

Invariant: A | sorted, <= A[i..n] | ? |

```
insertionSort(A):
  i = 0;
  while i < A.length:
    // bubble A[i] to its sorted position by repeatedly
    //    swapping with the element to its left
    // increment i
```

Invariant: A

| sorted | ? |

(with `i` marking the boundary between sorted and ?)

```
selectionSort(A):
  i = 0;
  while i < A.length:
    // find minimum value in A[i..A.length]
    // swap it with A[i]
    // increment i
```

Invariant: A

| sorted, <= A[i..n] | ? |

(with `i` marking the boundary between sorted and ?)

Demo: https://visualgo.net/bn/sorting

Developing Insertion Sort

Insert A[i] into the sorted sublist A[0..i-1].

Pre: A — i, sorted, ?

Invariant: A — sorted, j, i, ?

Post: A — sorted, i = A.length
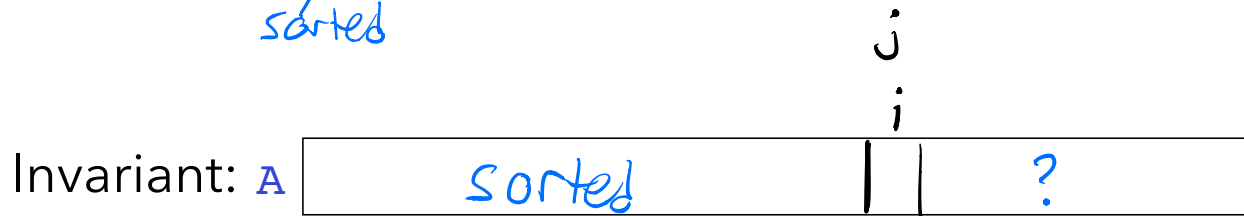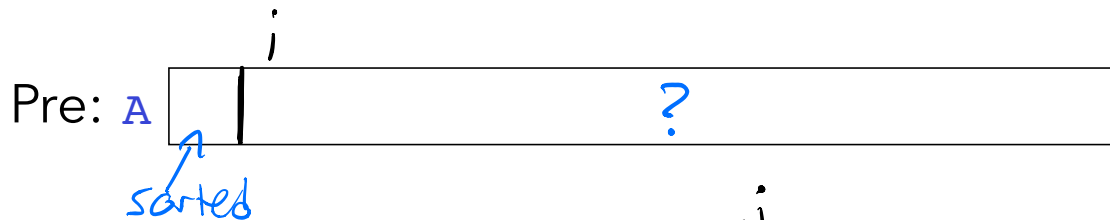
```
i = 1
While  i < A.length :
    j = i
    While ( j > 0 and A[j-1] > A[j])
        swap (A, j, j-1)
        j--
    i++
```