CIS 575. Introduction to Algorithm Analysis
Material for January 19, 2024

Insertion Sort: Iterative Implementation

This topic is covered in *Cormen*'s Section 2.1.

1 Bottom-Up Insertion Sort

Recall that we developed a top-down version of *insertion sort*:

```
InsertionSort(A[1..n])
  if n > 1
    InsertionSort(A[1..n − 1])
    InsertLast(A[1..n])
```

By repeatedly "unfolding" the calls to InsertionSort, a call InsertionSort($A[1..n]$) unfolds to

InsertLast($A[1..2]$)
...
InsertLast($A[1..n]$)

This shows that a *bottom-up* version of *insertion sort* can be implemented by the iterative algorithm

```
InsertionSort(A[1..n])
  for i ← 2 to n
    InsertLast(A[1..i])
```

2 Insert Last Element in Proper Place

We can no longer postpone the implementation of InsertLast:

Input: $A[1..n]$ with $A[1..\mathbf{n-1}]$ non-decreasing
Output: $A[1..n]$ is a permutation of its original values such that $A[1..\mathbf{n}]$ is non-decreasing

```
InsertLast(A[1..n])
  if n > 1 and A[n] < A[n − 1]
    A[n] ↔ A[n − 1]
    InsertLast(A[1..n − 1])
```

We have used another convenient bit of notation: $x \leftrightarrow y$ for the *swapping* of the contents of x and y, which in most languages will require a temporary variables, and 3 (one-way) assignments.

The above implementation is recursive, and we have just argued that recursive functions may have issues with stack use. But observe that the recursive call to InsertLast is a *tail call* in that there is nothing to return to, and hence there is no need to save anything in the stack! (In CIS505 we discuss the concept of *tail recursion* in much more detail.) And in fact, we can easily convert into an equivalent iterative algorithm:

InsertLast($A[1..n]$)
  $j \leftarrow n$
  while $j > 1$ and $A[j] < A[j-1]$
    $A[j] \leftrightarrow A[j-1]$
    $j \leftarrow j-1$

3 Iterative Insertion Sort

Collecting the pieces, we end up with

InsertionSort($A[1..n]$)
  for $i \leftarrow 2$ **to** n
    $j \leftarrow i$
    while $j > 1$ and $A[j] < A[j-1]$
      $A[j] \leftrightarrow A[j-1]$
      $j \leftarrow j-1$