Insertion Sort Algorithm Proof

Kelin Luo

1 Preliminary

In this note, we will demonstrate how to prove correctness using an induction proof. Specifically, we will prove that the invariant "after iteration j of the outer loop, $A[1...j]$ is the sorted array for the original $A[1...j]$" holds true at all times. This invariant is crucial in establishing the algorithm's correctness.

2 Proof for Insertion Sort Algorithm

Lemma 1. *After iteration j of the outer loop, $A[1...j]$ is the sorted array for the original $A[1...j]$.*

Proof. We will proceed with a formal proof by induction.

- Base case: This is obviously true when $i = 1$ because $A[1...1]$ consists of a single number, which is naturally sorted.
- Inductive case: Assume that the inductive hypothesis holds for $j - 1$, meaning that $A[1...j-1]$ is sorted after the $(j-1)$-th iteration. We aim to show that $A[1...j]$ is sorted after the j-th iteration.

  Consider the $key = A[j]$. Let i be the largest index in $1, 2, ..., j-1$ such that $A[i] \leq key$. In the inner loop, we find the largest index i in $\{1, 2, ..., j-1\}$ such that $A[i] \leq key$, or $i = 0$ if $A[1] > key$. The list is then updated to: $A[1], ..., A[i], A[i+1] = key, A[i+2], ..., A[j-1], A[j]$.

  By the inductive hypothesis, we know that $A[1] \leq ... \leq A[i] \leq A[i+2] \leq ... \leq A[j-1] \leq A[j]$. We claim that the resulting list with key is sorted. Since i is the largest index in the original sorted list such that $A[i] \leq key$, we have:

  $$A[1] \leq ... \leq A[i] \leq key \text{ (since } i \text{ is the largest index satisfying } A[i] \leq key\text{)}$$

  $$key < A[i+2] \leq ... \leq A[j] \quad \text{(since key is smaller than } A[i+2]\text{)}.$$

  Thus, key is in its correct position. All other numbers were already in their correct positions, proving the claim. Therefore, after the j-th iteration, $A[1...j]$ is sorted, so this proves the inductive case.

By induction, we conclude that the lemma holds. $\square$