1 Insertion Sort without Sentinel

1.1 Pseudocode

Algorithm 1 Insertion Sort without Sentinel

```
for $i = 2$ to $n$ do
    $t \leftarrow A[i]$
    $j \leftarrow i - 1$
    while $j > 0$ and $A[j] > t$ do
        $A[j+1] \leftarrow A[j]$
        $j \leftarrow j - 1$
    end while
    $A[j+1] \leftarrow t$
end for
```

1.2 Analysis of Comparisons

Worst Case

Worst case is when the array is reverse sorted, and every ith iteration of the loop must compare against all previous $(i-1)$ elements.

$$\begin{aligned}\sum_{i=2}^{n}\sum_{j-1}^{i-1} 1 = \sum_{i=2}^{n}(i-1) &= \sum_{i=2}^{n} i - \sum_{i=2}^{n} 1 \\ &= \frac{(n+1)n}{2} - (1-(n-1)) \\ &= \frac{(n-1)n}{2}\end{aligned}$$

Best Case

Best case is when the array is already sorted, so there will just be 1 comparison per iteration of the outermost loop. (Same as Insertion Sort with Sentinel.)

$$\sum_{i=2}^{n} 1 = (n-2) + 1 = n - 1$$

Average Case

On average, the sentinel costs $\sum_{i=2}^{n} \frac{1}{i}$ comparisons. This is simply just the Harmonic Series. In other words, the sentinel costs $H_n - 1$ comparisons, so Insertion

Sort without Sentinel is

$$\frac{(n+4)(n-1)}{4} - (H_n - 1) \approx \frac{(n+4)(n-1)}{4} - \ln n$$

1.3 Analysis of Exchanges

Removing the sentinel adds no new exchanges so the best, worst, and average cases are all the same as Insertion Sort with Sentinel.