webcast: username: cmps-101
password: algorithm

Sorting Algorithms

notation: \( A = (A_1, A_2, ..., A_n) \)
where \( n = \text{length}[A] \).

\[ A[i...j] = (A_i, A_{i+1}, ..., A_j) \]
**Insertion Sort** \((A)\)

1.) for \(i = 2\) to \(n\)
2.) \(\text{temp} = A_{j}\)
3.) \(i = i - 1\)
4.) while \(i > 0\) and \(\text{temp} < A_{i}\)
5.) \(A_{i+1} = A_{i}\)
6.) \(i = i - 1\)
7.) \(A_{i+1} = \text{temp}\)

\[A_{1}, A_{2}, \ldots, A_{j-1}, A_{j}, A_{i+1}, \ldots, A_{n}\]

- **sorted**
- **unsorted**
Invariant:

\[ A[1 \ldots (i-1)] \] is sorted before the \( i \)th iteration of outer for loop.

Goal: find run time (cost) of

\[ \text{InsertionSort}(A): \ \Omega(n) \]

let \( c_k = \text{cost of step } k \) \((1 \leq k \leq 2)\)

let \( t_j = \# \text{ of executions of while loop test on } j\text{th iteration of outer for loop} \)