## INSERTION SORT

**Algorithm, Complexity**

- **Insertion Sort: Intro**
  - One of the simplest (to code) sorting algorithms
  - Consists of N passes
  - Consider a pass $p = 1$ through $N - 1$,
    - Ensure: elements 0 through $p$ are in sorted order
      - Use: elements 0 through $p - 1$ are already sorted
    - How?
      - Move element in position $p$ left, until its correct place is found among first $p + 1$ elements
In-class Problem

- What is the run-time complexity of the following code?

```cpp
sum = 0;
for(i = 1; i < n; i++)
    for(j = 1; j < i*i; j++)
        if (j % i == 0)
            for(k = 0; k < j; k++)
                sum++;
```

- What's the complexity with the above if statement added?

Insertion Sort: Example

<table>
<thead>
<tr>
<th>Original</th>
<th>34</th>
<th>8</th>
<th>64</th>
<th>51</th>
<th>32</th>
<th>21</th>
</tr>
</thead>
</table>

**Figure 7.1** Insertion sort after each pass
Insertion Sort: Code

- Sorting an array of integers:
  ```
  int j;
  for (int p = 1; p < a.length; p++) {
    int tmp = a[p];
    for (j = p; ((j>0) && tmp < a[j-1])); j--)
      a[j] = a[j-1];
    a[j] = tmp;
  }
  ```

- Online demo:
  - [http://www.sorting-algorithms.com/insertion-sort](http://www.sorting-algorithms.com/insertion-sort)