On the Limits of Computing

- Reasons for Failure
  1. Runs too long
     - Real time requirements
     - Predicting yesterday's weather
  2. Non-computable!
  3. Don't know the algorithm
- Complexity, N
  - Time
  - Space
- Tractable and Intractable
- Study of a Sorting Algorithm
  - Sorting, Ordering
  - Alphabetizing

Sorting Example

- Selection Sort
  - N items in an array named Data
  - Find smallest of elements 1 thru N of Data
  - Interchange this with 1st element of array Data
  - Find smallest of elements 2 thru N of Data
  - Interchange this with 2nd element of array Data
  - ... Find smallest of elements K thru N of Data
  - Interchange this with Kth element of array Data
  - Done when K = N

Analysis of Sorting Example

- How Many Operations?
  - Comparisons
  - N-1 comparisons in first pass
  - N-2 comparisons in first pass
  - ... 1 comparisons in last pass
  - N-1 + N-2 + N-3 + ... 2 + 1
  - N*(N-1)/2 = N*N/2 - N/2 (Gauss)
- What does Order N Square Mean?
  - Examples
### Polynomial Time

- **Linear Time Algorithms**
  - Add elements of an array
  - Single loop algorithms
  - $t = A \cdot N$

- **Cubic Time Algorithms**
  - Matrix multiplication
  - $t = A \cdot N^3$

- **Polynomial Time**
  - $t = A \cdot N^K$
  - ... and in-between
  - Faster machines make a lot of difference

- **Quicksort**
  - $t = A \cdot N \cdot \log(N)$
  - Logarithmic behavior

### Polynomial Time

- What does Order log(N) or $N\log(N)$ Mean?
  - Various values of $N$

<table>
<thead>
<tr>
<th>$N$</th>
<th>log(N)</th>
<th>$N\log(N)$</th>
<th>$N^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>24</td>
<td>64</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>1K</td>
<td>10</td>
<td>10K</td>
<td>1M</td>
</tr>
<tr>
<td>2K</td>
<td>11</td>
<td>22K</td>
<td>4M</td>
</tr>
<tr>
<td>8K</td>
<td>13</td>
<td>1M</td>
<td>64M</td>
</tr>
<tr>
<td>1M</td>
<td>20</td>
<td>20M</td>
<td>1T</td>
</tr>
<tr>
<td>2M</td>
<td>21</td>
<td>42M</td>
<td>4T</td>
</tr>
</tbody>
</table>

- $K = 1024; \ M = K \cdot K; \ G = K \cdot M; \ T = K \cdot G$

### Tractable Algorithms

- **Graphs Showing Complexity**
  - Polynomial = *Tractable*

- **Binary Search**
  - Assumes Sorted
  - Like telephone book lookup
  - *Logarithmic Time*
  - $t = A \cdot \log(N)$

- **Intractable Algorithms**
  - Computer "crawls" or seems to come to halt for large $N$
  - Large problems essentially unsolved
  - May never be able to compute answer for some obvious questions