Class Announcements/Reminders:

- HW1 due 11 PM tonight. No late submissions accepted.
- P1 extension by one day, minor corrections to documentation and code.
- ODT files for outlines
- Eclipse, Java resources on class homepage

Last class:

- Primitives vs. References (finish)
- Command-Line Java
- Exceptions

Today:

- Exceptions – Behind the Scenes
- Complexity
  - Concepts, Definitions, Assumptions, Examples
  - Growth of functions and notation
Exceptions – Behind the Scenes

Normal execution mode

Exception-handling mode

- If in a try block

- If not in a try block

Back to normal execution mode – but where?
Complexity - Intro

Quantifying “efficiency”

Examples

Resources: number of “basic” operations (time) and memory used (space)

Model of computation

Complexity: Predicting how resources required grow with problem size
Complexity: Examples

Number-guessing game

Array assignment loop:

```java
for (int i = 0; i < n; i++) {
    a[i] = i;
}
```
Complexity: Insertion sort

Pseudocode:

\begin{verbatim}
INSERTION-SORT(A)
1   for (j = 1 to A.length - 1)
2       key = A[j]
3       i = j - 1
4       while (i >= 0 and A[i] > key)
5           A[i + 1] = A[i]
6           i = i - 1
7       A[i + 1] = key
\end{verbatim}

Best-case running time

Worst-case running time
Complexity: Simplifying Assumptions
Complexity: How functions grow

Higher order terms compared to rest

Example functions:
Complexity: How functions grow (cont’d)

<table>
<thead>
<tr>
<th>n</th>
<th>nlog(n)</th>
<th>n^2</th>
<th>2^n</th>
<th>n!</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>33.2</td>
<td>100</td>
<td>1024</td>
<td>3628800</td>
</tr>
<tr>
<td>20</td>
<td>86.4</td>
<td>400</td>
<td>1048576</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>664.4</td>
<td>10000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>9965.8</td>
<td>1000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Growth of common functions with n

- **1**
- **log(n)**
- **n**
- **nlog(n)**
- **n^2**
Asymptotic efficiency: Theta and Big-O notation

Definitions:

Example:

Abusing notation