You need a partner, and at least half a laptop. These slides are posted.
Maps can:
• Make an empty one (*constructor*)
• Add a key-value pair (*put*)
• Check if a key is in the map (*containsKey*)
• Get the value for a key (*get*)
• Tell you its size (*size*)
One way to do a map

ArrayList of Pairs

m =

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
</tbody>
</table>

System.out.println(m.get("CA"));
One way to do a map

ArrayList of Pairs

\[ m = \begin{array}{c|c|c}
\text{AK} & \text{NC} & \text{CA} \\
49 & 12 & 31 \\
\end{array} \]

\[ m.\text{put(”NM”, 47);} \]
One way to do a map

ArrayList of Pairs

\[
\begin{array}{c|c|c|c}
\text{Key} & \text{Value} \\
\hline
AK & 49 \\
NC & 12 \\
CA & 31 \\
NM & 47 \\
\end{array}
\]

\( m = \) \[ m.put("NM", 47); \]
Snarf Sep 12 InClass

ReadArrayListMap.java
(and Pair.java)
CountUniqueWords.java
0. Time *War & Peace* for
   - 100 words
   - 500 words
   - 2500 words
   - 12,500 words
   - 62,500 words
   - 312,500 words
   - 565,460 words (the total)

Once you have five datapoints:
1. Plot your five points.
2. Have a group member put a hand up.
3. Keep working on six and seven!

Ask: *How does this scale?*

1. Plot the timings using your favorite tool.
2. Enter them in
My timings

Two minutes!

Ask: Why?
So, what’s going on?

Map Value

Key
Value

Comparisons: 0
Operations: 0

Code to run next

```java
m.put("NY", 11);
```
So, what’s going on?

Map Value

Key Value

NY

11

Comparisons: 0
Operations: 1

Code to run next

```java
m.put("NC", 12);
```
So, what’s going on?

Map Value

Key
Value

NY 11
NC 12

Comparisons: 1
Operations: 2

Code to run next

```java
m.put("AK", 49);
```
So, what’s going on?

Map Value

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
</tbody>
</table>

Comparisons: 3
Operations: 3

Code to run next

```
m.put("CA", 31);
```
So, what’s going on?

Map Value

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
</tbody>
</table>

Comparisons: 6
Operations: 4

Code to run next

```java
m.put("NM", 47);
```
So, what’s going on?

Map Value

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
<tr>
<td>NM</td>
<td>47</td>
</tr>
</tbody>
</table>

Comparisons: 10
Operations: 5

Code to run next

```java
m.put("WA", 42);
```
So, what’s going on?

Map Value

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
<tr>
<td>NM</td>
<td>47</td>
</tr>
<tr>
<td>WA</td>
<td>42</td>
</tr>
</tbody>
</table>

Comparisons: 15  
Operations: 6

Code to run next

```java
m.put("DC", 51);
```
So, what’s going on?

Map Value

Key Value

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
<tr>
<td>NM</td>
<td>47</td>
</tr>
<tr>
<td>WA</td>
<td>42</td>
</tr>
<tr>
<td>DC</td>
<td>51</td>
</tr>
</tbody>
</table>

Comparisons: 21
Operations: 7
1 + 2 + 3 + ... + N

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
<tr>
<td>NM</td>
<td>47</td>
</tr>
<tr>
<td>WA</td>
<td>42</td>
</tr>
<tr>
<td>DC</td>
<td>51</td>
</tr>
</tbody>
</table>

![Bar graph](image)

This story might be true.
The sum of the first $N$ natural numbers is given by:

$$1 + 2 + 3 + \ldots + N = \sum_{i=1}^{n} i = \frac{n(n + 1)}{2} = \frac{n^2 + n}{2}$$

Where $N$ is the number of insertions. This formula indicates that $N$ insertions run in time quadratic in $N$.

### Key

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
<tr>
<td>NM</td>
<td>47</td>
</tr>
<tr>
<td>WA</td>
<td>42</td>
</tr>
<tr>
<td>DC</td>
<td>51</td>
</tr>
</tbody>
</table>

### Graph

- The graph shows a trend of increasing values with $N^2$.
- The trend indicates a quadratic relationship.
Was this fair?

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>11</td>
</tr>
<tr>
<td>NC</td>
<td>12</td>
</tr>
<tr>
<td>AK</td>
<td>49</td>
</tr>
<tr>
<td>CA</td>
<td>31</td>
</tr>
<tr>
<td>NM</td>
<td>47</td>
</tr>
<tr>
<td>WA</td>
<td>42</td>
</tr>
<tr>
<td>DC</td>
<td>51</td>
</tr>
</tbody>
</table>
Different Case

Map Value

Key
Value

Comparisons: 0
Operations: 0

Code to run next

```java
m.put("NY", 11);
```
Different Case

Map Value

Key      Value
NY      11

Comparisons: 0
Operations: 1

Code to run next

```java
m.put("NY", 12);
```
Different Case

Map Value

Key
Value

NY
12

Comparisons: 1
Operations: 2

Code to run next

m.put(“NY”, 13);
Different Case

Map Value

Key
Value

NY
13

Comparisons: 2
Operations: 3

Code to run next

m.put("NY", 14);
Different Case

Map Value

Key
Value

NY
14

Comparisons: 3
Operations: 4
War & Peace

Comparisons vs. Insertions

- Worst Case
- Your Case
- Best Case

Tuesday, September 11, 12
To Do: complete the graph

(in ms)

0. Time cheese.txt and random.txt for
• 100 words
• 500 words
• 2500 words
• 12,500 words
• 62,500 words
• 312,500 words
• 565,460 words

random is suffering by now; best to not try the bigger ones...

1. Plot the timings using your favorite tool.
2. Enter them in

http://www.cs.duke.edu/courses/fall12/compsci201/charts/sep12.html