What is Tree Traversal?

When we consider a tree, there are many possible ways to traverse the nodes in that tree. The most general classes of tree traversal are:

- Breadth-first Traversal
- Depth-first Traversal
  - Pre-order Traversal
  - In-order Traversal
  - Post-order Traversal
Breadth-first Traversal

In a *breadth-first traversal* one considers all siblings of a particular node before considering its children. That is, we traverse the tree in a “layered” manner in which we first consider the root, followed by all its children, followed by all its children’s children, etc.
Depth-first Traversal

In a *depth-first traversal* one considers all the children of a particular node before considering its siblings. That is, we traverse the tree by following a single path as deeply as possible into the tree, before backtracking and considering other paths.

A depth-first traversal in a binary tree can be broken down further based on the order the children are considered with respect to the current node.

**Pre-order Traversal**  First consider the current node, then consider the left child, then consider the right child.

**In-order Traversal**  First consider the left child, then consider the current node, then consider the right child.

**Post-order Traversal**  First consider the left child, then consider the right child, then consider the current node.
Depth-first Traversal

Pre-order Traversal

1
2
3 4
5
6 7
Depth-first Traversal

In-order Traversal

```
1 2 3 4 5 6 7
```

Dr. J. Pfoh
Depth-first Traversal

Post-order Traversal

```
1 2
3
4 5
6
7
```
Searching in an unsorted space means that you will have to visit each member to perform the search.
However, if we sort this space, the search requires fewer visits of the nodes 2, 5, 7, 12, 17, 20, 33, 34, 50, 65.
A binary search is a method of searching a sorted space (e.g., array). Assume we want to determine whether the number 33 is in our array.

We can begin with the middle element and determine whether it is equal to our target, is greater than our target, or is less than our target and effectively reduce our search space by half.
As $20 < 33$, we can ignore the current node and all elements to the left (less than the current element).

At this point, we can again consider the middle element of what is left over and proceed in the same fashion.
As $50 > 33$, we can ignore the current node and all elements to the right (greater than the current element) and continue our process.
Finally, we land on the element (33) we are searching for after having only considered 4 of the 10 elements.
There are many methods by which to sort elements, we will consider three of them:

- Bubblesort
- Mergesort
- Quicksort
A bubblesort is one of the more simple sorting algorithms. While it is not terribly efficient, it is a good place to start.

The general idea is that we iterate over all elements in order and if the next element is less than the current, we switch the two. We continuously iterate over the entire array until we can iterate over it without performing any switches at which point the array is sorted.
Bubblesort
First Iteration

Consider the following, unsorted array. We will begin with our first iteration.

```
12  2  65  34  5  20  17  50  7  33
```

This iteration begins by considering the first element (12) and comparing it with the next element (2). Since, 12 > 2, we switch the two and consider the next element.

```
2  12  65  34  5  20  17  50  7  33
```
Bubblesort

First Iteration

We continue our first iteration...

\[
\begin{array}{cccccccccc}
2 & 12 & 65 & 34 & 5 & 20 & 17 & 50 & 7 & 33 \\
\end{array}
\]

As \(12 < 65\), we do not perform a switch and simply consider the next element...

\[
\begin{array}{cccccccccc}
2 & 12 & 65 & 34 & 5 & 20 & 17 & 50 & 7 & 33 \\
\end{array}
\]
Bubblesort
First Iteration

We continue our first iteration...

\[
\begin{array}{cccccccccc}
2 & 12 & 65 & 34 & 5 & 20 & 17 & 50 & 7 & 33 \\
\end{array}
\]

As 65 > 34, we perform a switch and consider the next element...

\[
\begin{array}{cccccccccc}
2 & 12 & 34 & 65 & 5 & 20 & 17 & 50 & 7 & 33 \\
\end{array}
\]
Bubblesort
First Iteration

We continue our first iteration...

2  12  34  65  5  20  17  50  7  33

As 65 > 5, we perform a switch and consider the next element...

2  12  34  5  65  20  17  50  7  33
We continue our first iteration...

2 12 34 5 65 20 17 50 7 33

As 65 > 20, we perform a switch and consider the next element...

2 12 34 5 20 65 17 50 7 33
Bubblesort
First Iteration

We continue our first iteration...

As $65 > 17$, we perform a switch and consider the next element...
Bubblesort
First Iteration

We continue our first iteration...

```
  2  12  34  5  20  17  65  50  7  33
```

As 65 > 50, we perform a switch and consider the next element...

```
  2  12  34  5  20  17  50  65  7  33
```
Bubblesort
First Iteration

We continue our first iteration...

\[
\begin{array}{cccccccccc}
2 & 12 & 34 & 5 & 20 & 17 & 50 & 65 & 7 & 33 \\
\end{array}
\]

As 65 > 7, we perform a switch and consider the next element...

\[
\begin{array}{cccccccccc}
2 & 12 & 34 & 5 & 20 & 17 & 50 & 7 & 65 & 33 \\
\end{array}
\]
Bubblesort

First Iteration

We continue our first iteration...

\[
\begin{array}{cccccccc}
2 & 12 & 34 & 5 & 20 & 17 & 50 & 7 & 65 & 33 \\
\end{array}
\]

As \(65 > 33\), we perform a switch and are finished with our first iteration

\[
\begin{array}{cccccccc}
2 & 12 & 34 & 5 & 20 & 17 & 50 & 7 & 33 & 65 \\
\end{array}
\]

However, the array is not yet sorted, we must continue iterations until a single iteration causes no more switches.
## Bubblesort

### Second Iteration

Here, the second iteration...

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>12</th>
<th>34</th>
<th>5</th>
<th>20</th>
<th>17</th>
<th>50</th>
<th>7</th>
<th>33</th>
<th>65</th>
<th>no switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>12</td>
<td>34</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>50</td>
<td>7</td>
<td>33</td>
<td>65</td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>34</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>50</td>
<td>7</td>
<td>33</td>
<td>65</td>
<td>switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>34</td>
<td>20</td>
<td>17</td>
<td>50</td>
<td>7</td>
<td>33</td>
<td>65</td>
<td>switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>34</td>
<td>17</td>
<td>50</td>
<td>7</td>
<td>33</td>
<td>65</td>
<td>switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>50</td>
<td>7</td>
<td>33</td>
<td>65</td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>50</td>
<td>7</td>
<td>33</td>
<td>65</td>
<td>switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>50</td>
<td>33</td>
<td>65</td>
<td>switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
<td></td>
</tr>
</tbody>
</table>
Bubblesort

Third Iteration

and the third iteration...

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>12</td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td><strong>12</strong></td>
<td>5</td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td><strong>12</strong></td>
<td>20</td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td><strong>20</strong></td>
<td>17</td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td><strong>20</strong></td>
<td>34</td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td><strong>34</strong></td>
<td>7</td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td><strong>34</strong></td>
<td>33</td>
<td>50</td>
<td>65</td>
<td>switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td><strong>34</strong></td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td><strong>50</strong></td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
</tbody>
</table>
Bubblesort

Fourth Iteration

now the fourth iteration...

<table>
<thead>
<tr>
<th>2</th>
<th>5</th>
<th>12</th>
<th>17</th>
<th>20</th>
<th>7</th>
<th>33</th>
<th>34</th>
<th>50</th>
<th>65</th>
<th>no switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>7</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>7</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>7</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>7</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>17</td>
<td>7</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
</tbody>
</table>
**Bubblesort**

**Fifth Iteration**

you guessed it, the fifth iteration...

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>7</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>7</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>7</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>7</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>switch</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
<td></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
<td><strong>12</strong></td>
<td><strong>17</strong></td>
<td><strong>20</strong></td>
<td><strong>33</strong></td>
<td><strong>34</strong></td>
<td><strong>50</strong></td>
<td><strong>65</strong></td>
<td>no switch</td>
<td></td>
</tr>
</tbody>
</table>
Bubblesort

Sixth Iteration

can I get a sixth iteration?

<table>
<thead>
<tr>
<th>2</th>
<th>5</th>
<th>12</th>
<th>7</th>
<th>17</th>
<th>20</th>
<th>33</th>
<th>34</th>
<th>50</th>
<th>65</th>
<th>no switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
<td>no switch</td>
</tr>
</tbody>
</table>
**Bubblesort**

**Seventh Iteration**

lucky number seven...

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td><strong>5</strong></td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td><strong>7</strong></td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td><strong>12</strong></td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td><strong>17</strong></td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td><strong>20</strong></td>
<td>33</td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td><strong>33</strong></td>
<td>34</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td><strong>34</strong></td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td><strong>50</strong></td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>20</td>
<td>33</td>
<td>34</td>
<td>50</td>
<td><strong>65</strong></td>
</tr>
</tbody>
</table>
Mergesort is a divide and conquer approach to sorting.

The general idea is that we will split the array into two equal halves, sort the halves, then simply merge the two sorted halves together to create a sorted array.
Mergesort

split...

| 12 | 2  | 65 | 34 | 5  | 20 | 17 | 50 | 7  | 33 |

sort halves...

<table>
<thead>
<tr>
<th>12</th>
<th>2</th>
<th>65</th>
<th>34</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>17</td>
<td>50</td>
<td>7</td>
<td>33</td>
</tr>
</tbody>
</table>

merge...

<table>
<thead>
<tr>
<th>2</th>
<th>5</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>17</td>
<td>20</td>
</tr>
</tbody>
</table>

| 2  | 5  | 7  | 12 | 17 | 20 | 33 | 34 | 50 | 65 |
Sounds great, but how do we sort the halves?
Sounds great, but *how do we sort the halves?*

Obviously, we split each half into two equal, further halves, sort those halves, then simply merge those two sorted halves together to create two sorted halves.

This is starting to sound a lot like a recursive definition...

*OK, then what is our base case?*
Mergesort

Sounds great, but *how do we sort the halves?*

Obviously, we split each half into two equal, further halves, sort those halves, then simply merge those two sorted halves together to create two sorted halves. This is starting to sound a lot like a recursive definition...

*OK, then what is our base case?*

Well, single element arrays are pretty easy to sort...
Mergesort

mergesort([12,9,1,17])

12 9 1 17

mergesort([12,9])

12 9

mergesort([12])

12

mergesort([9])

9

mergesort([1,17])

1 17

mergesort([1])

1

mergesort([17])

17
Mergesort

merge([9,12],[1,17])

merge([12],[9])
merge([1],[17])

merge([9,12],[1,17])
Quicksort is another divide and conquer approach to sorting.

The general idea here is that we choose a pivot element and put everything less than or equal to the pivot element in one array and everything greater than the pivot element in another, then sort those and concatenate the two parts.

This, again, begs the question, “how we sort each part?” Well, you guessed it, we simply perform the same steps on each part.
Quicksort

choose pivot...

split based on pivot...

choose new pivots in each part...

split based on pivot...
Quicksort

choose new pivots in each part...

3 5 7 9

split based on pivot...

3 5 7 9

now we concatenate back up the chain...

3 5 7 9

more concatenation...

3 5 7 9

one more concatenation...

3 5 7 9

and done!
Quicksort

Quicksort([5,7,3,9])

3 5
Quicksort([3,5])

9 7
Quicksort([9,7])

3
Quicksort([3])

5
Quicksort([5])

QuickSort([])

7 9
Quicksort([7,9])

7
Quicksort([7])

9
Quicksort([9])

Dr. J. Pfoh
Algorithms - Sorting
WS 2013-2014 23.01.2014 39 / 40
Quicksort

3 5 7 9

3 5
7 9
cat([3,5],[7,9])

cat([3],[5])
cat([7,9],[7])
cat([3,5],[7,9])
cat([3],[])