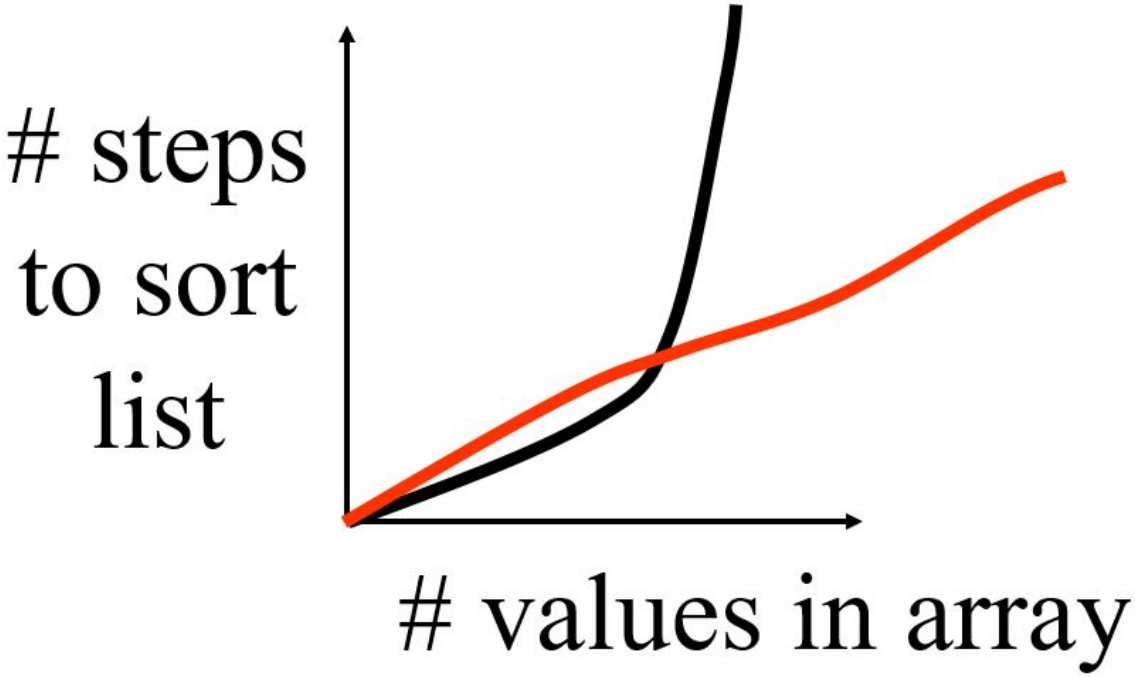
Recursive Merge Sort

Mr. Poole
Java

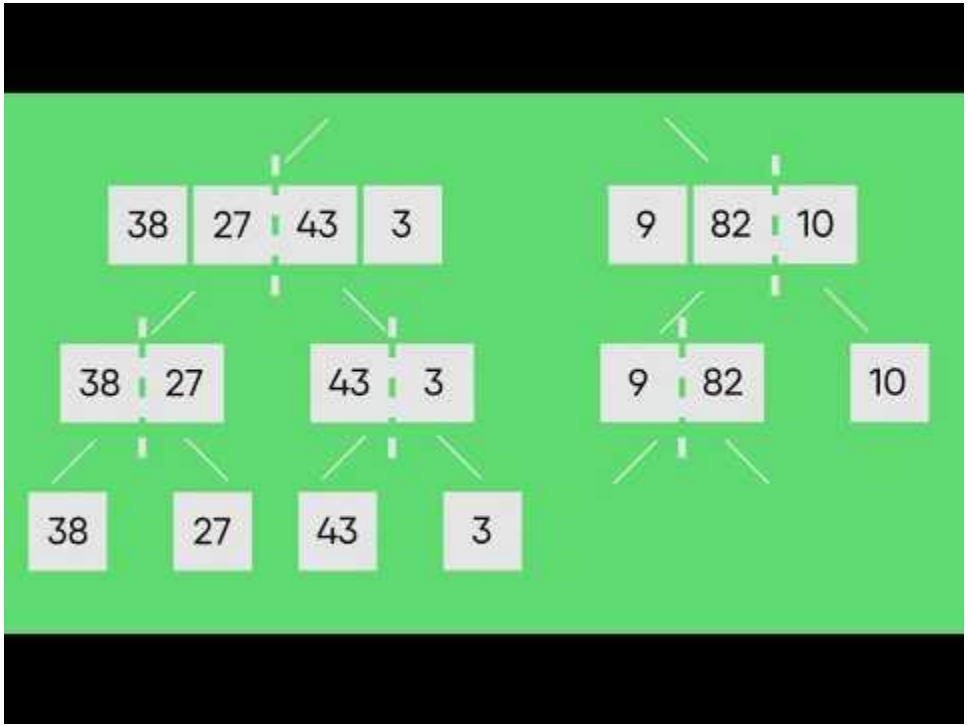
Quadratic sorting algorithms are nice but...



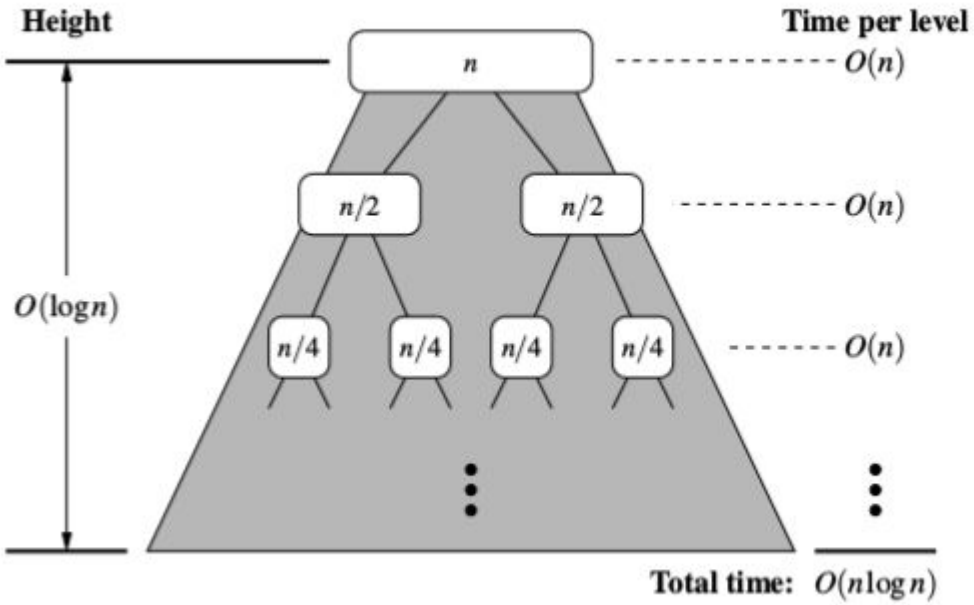
Merge Sort Concept

- **First Divide and Conquer Algorithm**
- • **O(log n):** Given a person's name, find the phone number by picking a random point about halfway through the part of the book you haven't searched yet, then checking to see whether the person's name is at that point. Then repeat the process about halfway through the part of the book where the person's name lies. (This is a binary search for a person's name.)

Sorting - Merge Sort



Runtime of Merge Sort



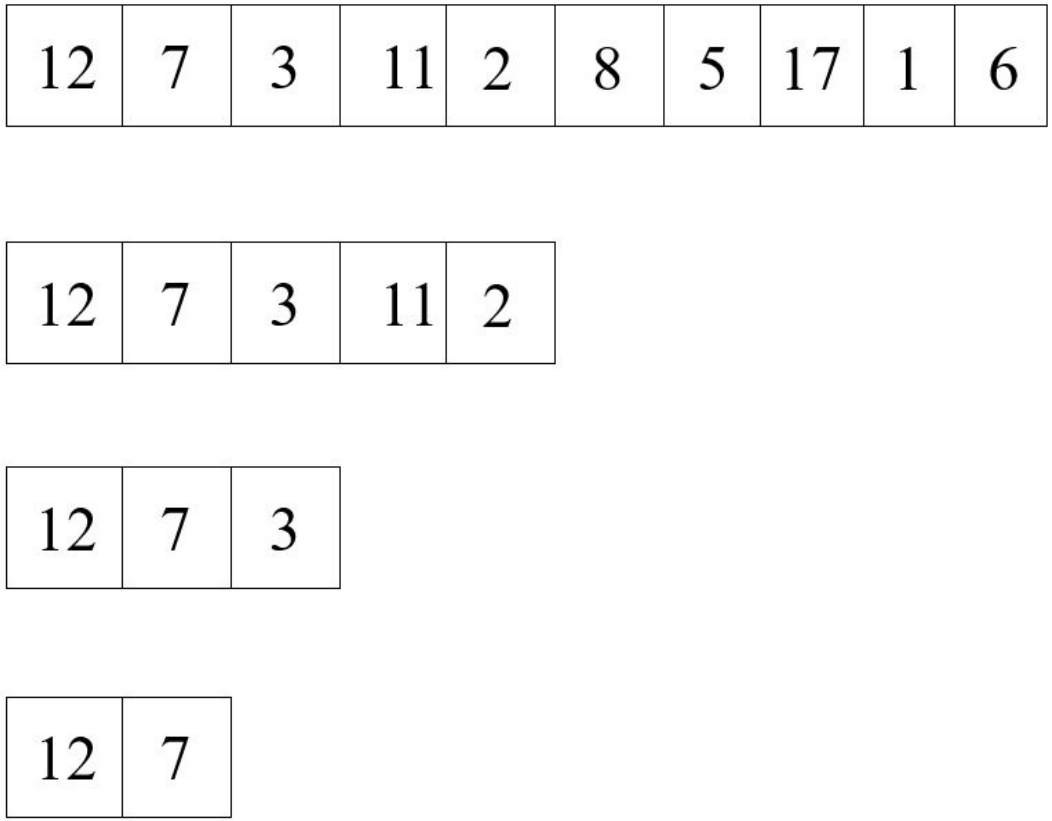
Merge Sort Pseudo Code

```
void mergeSort(int A[], int first, int last)
{
        // find middle index of A
        // sort the first half of A
        // sort the second half of A
        // merge the first and second halves of A
}
```

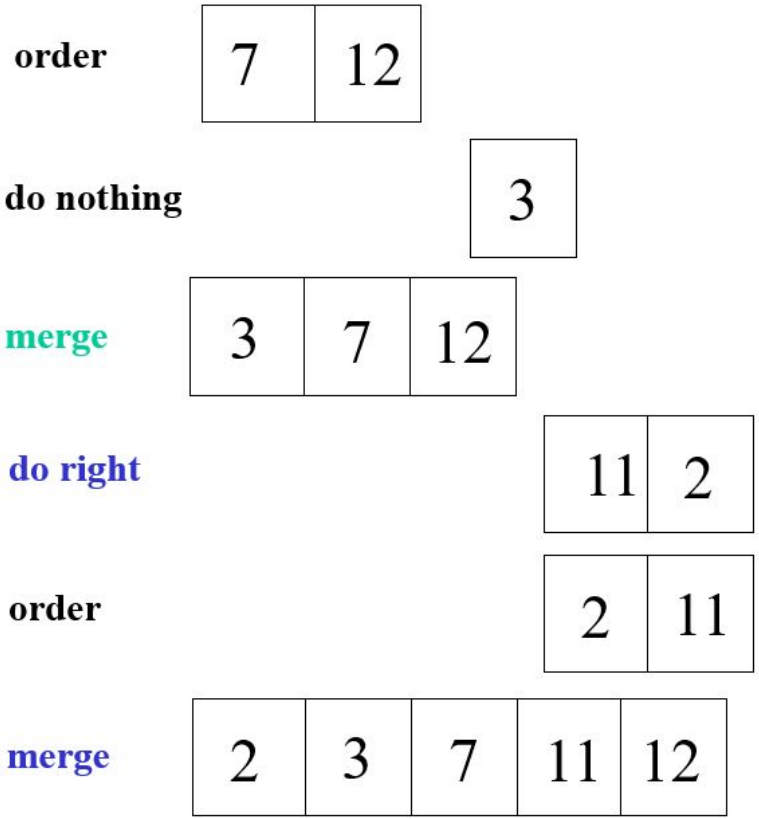
Recursive
Merge Sort Pseudo Code

```
void mergeSort(int A[], int first, int last)
{
        if(sublist has only one value)
                do nothing
        else if(sublist has two values)
                sort it if necessary
        else
                find midpoint of current sublist
                call mergeSort and process left sublist
                call mergeSort and process right sublist
                merge left and right sublists
}
```

How About an Example:

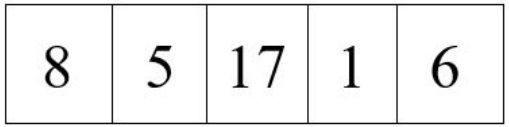


How About an Example:

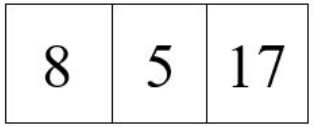


How About an Example:

do right

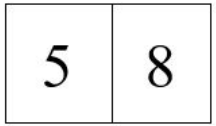


divide



divide

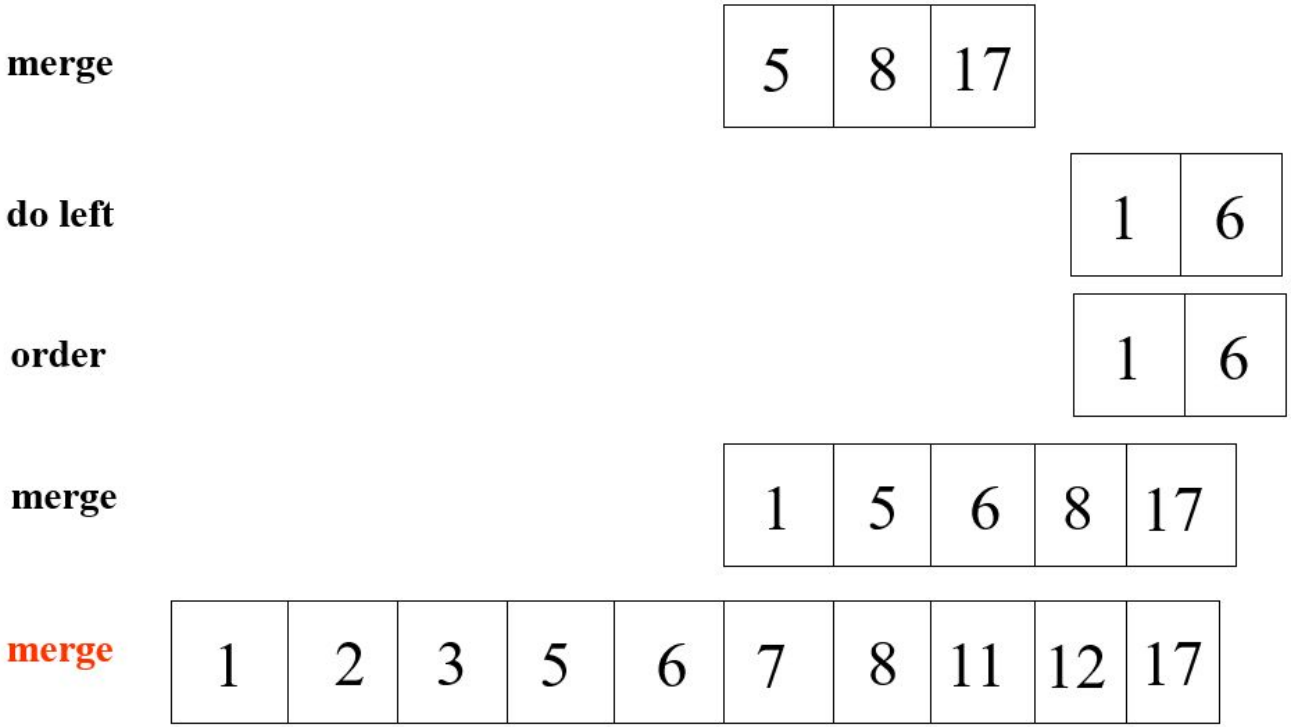
order



do left

do nothing

How About an Example:



Lab: Merge Sort

1. Implement Merge Sort on Arrays.