Study Write-Up: Mergesort

Mergesort is a good algorithm for sorting data. It is considered to be a divide-and-conquer algorithm. That is to say, it works by recursively breaking down the problem into several sub-problems until the problem can be solved directly. It then takes that solution and combines the solutions together to give the solution to the original problem. In this way, Mergesort is like binary search or Quicksort. Here is a simple demonstration of Mergesort:

Mergesort goes through the following sort of process to sort a given array of data:

1. Divide the unsorted list into *n* sub-lists, each containing 1 element (a list of 1 element is considered sorted).
2. Repeatedly merge sub-lists to produce new sorted sub-lists, by comparing values and making sure they're in order, until there is only 1 sublist remaining. This will be the sorted list.

Here is an example of Mergesort in Ruby:

```ruby
def merge_sort(lst)
  if lst.length <= 1
    lst
  else
    mid = (lst.length / 2).floor
    left = merge_sort(lst[0..mid - 1])
    right = merge_sort(lst[mid..lst.length])
    merge(left, right)
  end
end
```

```ruby
def merge(left, right)
  if left.empty?
    right
  elsif right.empty?
    left
  elsif left.first < right.first
    [left.first] + merge(left[1..left.length], right)
  else
    [right.first] + merge(left, right[1..right.length])
  end
end

merge_sort([5, 9, 3, 5, 1, 0, 2]) # returns [0, 1, 2, 3, 5, 5, 9]
```

Unfortunately for Mergesort, though, it is often beaten by one of its main competitors, Quicksort. It can be slower than Quicksort by several times, so Quicksort is often favored instead of it. It is, however, better to use than Quicksort in some cases, i.e., when there is not constant time to access any piece of memory on demand. Nevertheless, it is a good sorting algorithm, used by the likes of the Perl language and the Spidermonkey JavaScript engine. It's used by our computers to sort data everyday in all sorts of environments.