Quick sort

- Quick sort orders a list of values by partitioning the list around one element (the pivot), then sorting each partition

- The algorithm
  1. choose one element in the list to be the partition element
  2. organize the elements so that all elements less than the partition element are to the left and all greater are to the right
  3. apply the quick sort algorithm (recursively) to both partitions

- Nice if the partition element divides the list roughly in half
- Quick sort has two main methods
  - quickSort – performs recursive algorithm
  - partition – rearranges elements into two partitions

Example of Quicksort

- Input array
  - First element becomes the pivot
  - Pivot is placed to its final position during partition
  - Two sub-arrays are sorted recursively
  - First elements become pivots
  - Pivots are placed in final position
  - Four sub-arrays are sorted recursively
quickSort

quickSort(data, min, max) {
    if(min < max) {
        pivot = partition(data, min, max) // make partitions
        quickSort(data, min, pivot-1) // sort left partition
        quickSort(data, pivot+1, max) // sort right partition
    }
}
Heapsort(A)
  Build-Max-Heap(A)
  for i = A.length downto 2
  A.heap-size = A.heap-size - 1
  Max-Heapify(A, 1)