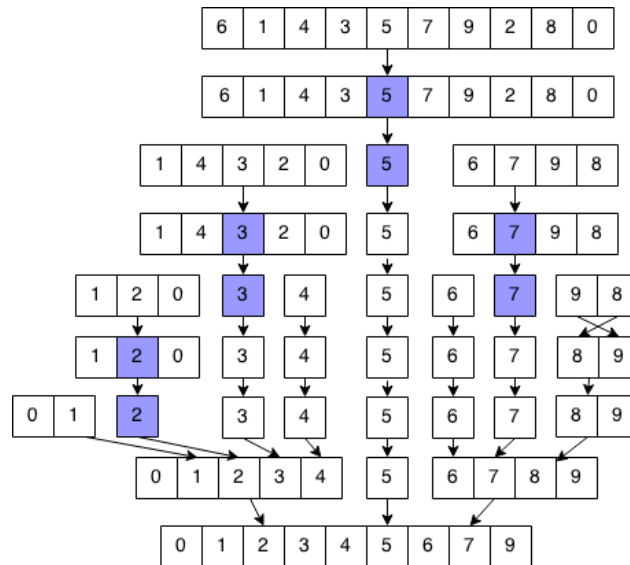


## Exercise: A faster QuickSort

In this exercise, you will use your knowledge of the Task Parallel Library to make a C# implementation of QuickSort run faster.

Quick Sort is a fast sorting algorithm that uses divide and conquer. In quick sort, an element is selected as a "pivot". The list is then divided into two sublists: a list of elements less than (or equal to) the pivot and a list of elements greater than the pivot. Each sublist is sorted and then appended together along with the origin pivot.



You can familiarize yourself with QuickSort on its Wikipedia page: <https://en.wikipedia.org/wiki/Quicksort>

Source of the text and image above is [http://www.thecshandbook.com/Quick\\_Sort](http://www.thecshandbook.com/Quick_Sort)

### Exercise 1:

A single-threaded C# implementation of QuickSort is provided on Blackboard.

Examine how the implementation works and **make it run faster** using your knowledge of the Task Parallel Library.

*How big a speedup (in percentage) can you achieve?*

(Remember to run in release mode and take the average time of multiple runs)

We have discussed, that Tasks should not be too small... whatever "too small" means....

So maybe you should stop spawning Tasks and just run the algorithm in a single Task, when the size of the block to sort becomes smaller than a given threshold?