

Annotated QuickSort

Example, sort:
10, 4, 8, 1, 7

```
void quickS( int start, int finish, int[] array)
{
  int pivot, left, right, temp;
  left = start;
  right = finish;
  pivot = array[ (left + right) / 2];
  while (right > left)
  {
```

0 and 4 and the array, in the example

$0 + 4 = 4 / 2 = 2$, which stores 8 in the example

So we don't cross over when working in from the ends.
(And actually the last time right and left do cross over,
and so then the outer while stops.)

```
    while (array[left] < pivot) {left = left + 1;}
```

```
    while (pivot < array[right]) {right = right - 1;}
```

Right being greater than left was already checked above.
But here right = to left is checked as well. This serves the purpose
of getting the left and right to cross over each other in preparation
for the recursive calls.

```
    if (left <= right)
    {
```

```
        temp = array[left];
        array[left] = array[right];
        array[right] = temp;
```

So swap the present left and right

```
        left = left + 1;
        right = right - 1;
```

Move both left and right in one, in order to set up
the next checks by the two whiles above.

```
    }
```

```
    if (start < right) quickS(start, right, array);
```

So for the left side, start stays start (which will become left).
Right is sent to finish, which will become right,
so right stays right, as it ended**.

```
    if (left < finish) quickS(left, finish, array);
```

And for the right side, left is sent to start, which will become left
so left stays left, as it ended.**
Finish stays finish (which will become right).

```
  }
```

The following two lines are to
get us to appropriate values to
swap. They will leave numbers
in place if they are less than the
pivot and left of it or greater
than the pivot and to the right
of it.

This recursively does the left,
since start stays start, and right
is now left of the pivot.

And this recursively does the
right, since finish stays finish,
& left is now right of the pivot.

**Remember that right ended up left of the pivot,
and left ended up right of the pivot.