Quicksort

(int) indices

Alg (pseudocode, sort of):

```
if(right > left)
quicksort(array, left, right) {

    // pick a pivot

    pivot = partition(array,
                      left, right,
                      pivot)
```

also as index, value

can change

quicksort(array, left,
pivot − 1)

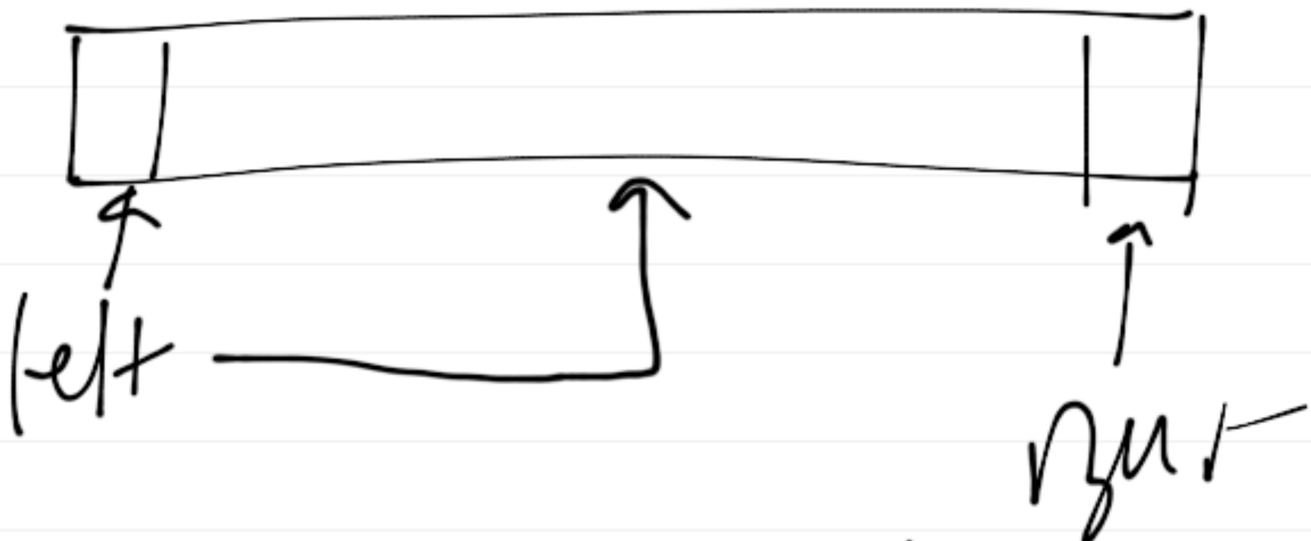
quicksort(array, pivot + 1,
right)

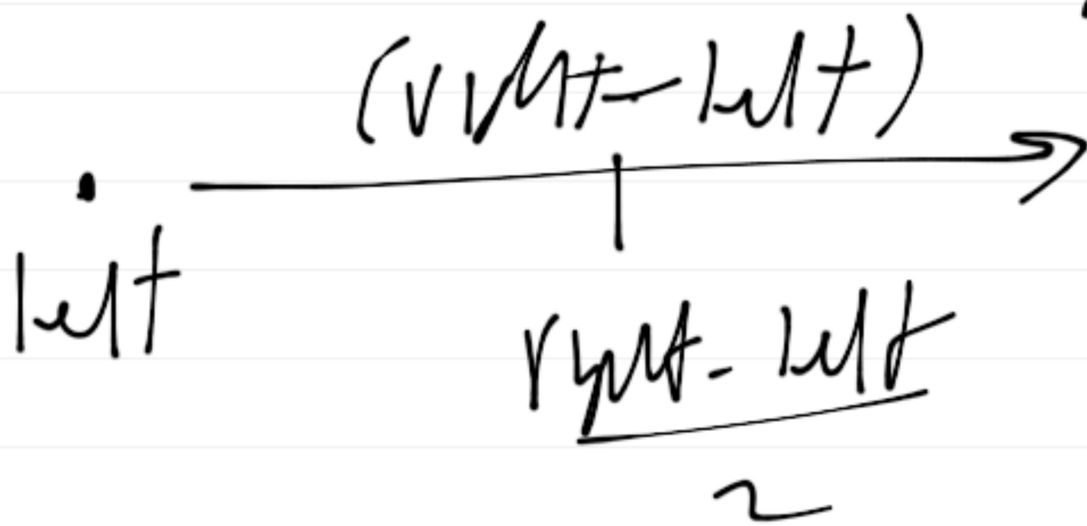
}

- picking a pivot:

ideally, pivot should be index of **median** element

$$pivot = left + (right - left)/2$$





Analysis:

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + T(\text{partition})$$

partition(array, left, right, pivot)

1. store data at pivot location

   the_pivot = array[pivot]

2. <u>move</u> pivot element to end of array

   swap(array[pivot], array[right])

3. go thru array, from left to right

   <u>moving</u> values $\leq$ the_pivot

   to the left of pivot index,

   advancing left index

```
for(i=left; i<right; i++){
    if(array[i] <= the_pivot){
        swap(arr[left], arr[i])
        left++;
    }
}
```

4. move pivot element to its final location

```
swap(array[left], array[right])
```

5. return pivot's new location

return (left)

(note: 'left' argument must be pass-by-value)

3 7 8 $\boxed{5}$ 2 1 9 5 4

left pivot right

i

3 7 8 4 2 1 9 5 $\boxed{5}$

i .. pivot

3 4 8 7 2 1 9 5 $\boxed{5}$

3 4 2 7 8 1 9 5 $\boxed{5}$

3 4 2 1 8 7 9 5 $\boxed{5}$

3 4 2 1 5 7 9 8 $\boxed{5}$

at end of for loop:

3 4 2 1 5 7 9 8 $\boxed{5}$

left

last step puts pivot in left location

3 4 2 1 5 $\boxed{5}$ 9 8 7

left list

right list

Note: Quicksort won't degenerate to bubblesort provided pivot is chosen as the median element in $O(n)$ time

linear time median search

Quicksort worst case analysis

$$T(n) = T(i) + T(n-i-1) + \underbrace{cn}_{\text{moving elements into position}}$$

$i = |S_1|$ the no. of elements in one side of the array

worst case: $i = 0$

Solve for $T(n)$

$k=0$

$T(n) = cn + T(n-1)$

$k=1$

$= cn + (c(n-1) + T(n-2))$

$= cn + cn - c + T(n-2)$

$k=2$

$= cn + cn - c + (c(n-2) + T(n-3))$

$= cn + cn + cn - c - 2c + T(n-3)$

$k=3$

$$= cn + cn + cn - c - 2c$$
$$+ \left(c(n-3) + T(n-4) \right)$$

$$= 4cn - c\sum_{k=0}^{n} k + T(n-4)$$

$$= c(k+1)n - \boxed{c\sum_{k=0}^{n} k} + T(n-k+1)$$

$$\sum_{k=0}^{n} k = \frac{n(n+1)}{2}$$

$$T(n) = c(k+1)n - c\frac{n(n+1)}{2}$$

$$+T(n-k+1)$$

let $k = n$

$$T(n) = c(n^2+n) - c\frac{n^2}{2}$$

$$-\frac{cn}{2} + T(1)$$

$$\frac{cn^2}{2} - \frac{cn}{2} \in O(n^2)$$