Quick sort dry run (working out with pen and paper)

4 || 1 8 9 3 8 1 9 3

4 1 3 1 3 8 9 9 8

Let’s bring the pivot to the correct place
3 1 3 1 4 8 9 9 8

pivot = 4

curr = 0
Swap arr[0] with arr[0] -> 1 remains at arr[0]

curr++ -> curr = 1

arr[1] -> 8 -> do nothing
arr[2] -> 9 -> do nothing

arr[3] -> 3 -> swap arr[curr] with arr[3] -> arr[1] with arr[3],
curr++ -> curr = 2

arr[4] -> 8 -> do nothing

arr[5] -> 1 -> swap arr[curr] with arr[5] -> arr[2] with arr[5],
curr++ -> curr = 3

arr[6] -> 9 -> do nothing

arr[7] -> 4 -> swap arr[curr] with arr[7] -> arr[3] with arr[7],
Curr++ -> curr = 4

Exercises
- Stable sort or not?
10 15(Mitali) 5 20 18 15(Parveen)
- Take a sorted array of size 10 and use quick sort to sort it (9, 8, 7, 6, ….1)

Next session - problem solving

- Given a an array, instead of selecting a fixed index for partitioning the array, use a random index for partitioning step to bring randomization in Quick Sort Algorithm

- Given an array and a number k , where k is smaller than the size of the array, we need to find the k’th smallest element in the given array. It is given that all array elements are distinct.

- Given a set of n nuts of different sizes and n bolts of different sizes. There is a one-one mapping between nuts and bolts. Match nuts and bolts efficiently.

Constraint: Comparison of a nut to another nut or a bolt to another bolt is not allowed. It means nut can only be compared with bolt and bolt can only be compared with nut to see which one is bigger/smaller.

Another way of asking this problem is, given a box with locks and keys where one lock can be opened by one key in the box. We need to match the pair.

Input :

```
char nuts[] = {‘@’, ‘#’, ‘$’, ‘%’, ‘^’, ‘&’}
char bolts[] = {‘$’, ‘%’, ‘&’, ‘^’, ‘@’, ‘#’}
```

Output :

```
Matched nuts and bolts are :
# $ % & @ ^
# $ % & @ ^
```