→ Quick sort

algorithm / implementation here considers the last element as pivot

look, we simply have to place the pivot element in the right place such that all elements left of it are smaller than it & right of it are bigger

how do we achive this?

we need to maintain a variable which will hold the "left most element which is greater than pivot's" 's index.
if we find a element less than pivot after finding a element greater than pivot towards the left. we spaw these two.

after iterating through loop we recurse to implement these on two half's on both sides excuding pivot element

```
void quickSort (int a[], int l, int r)
{
    if (l<r){ //base case.
        return;
    }
    pi = pivotplacer (a, l, r);
    quicksort (a, l, pi-1);
    quickSort (a, l, pi+1);
}
```

• experimental 2 code.

```
void pivotplacer(int a[], int l, int r)
{
    int i = -1;
    pivot = a[r]; // Specific to this implementation
    for(int j = l; j < r; j++)
    {
        if (a[j] > pivot && i == -1)
        {
            i = j;
        }
        if (a[j] < pivot && i != -1)
        {
            swap(a, j, i);
            i = j;
        }
    }
    swap(a, i, l); return (i);
}
```

```
Void pivotplacer(int a[], int l, int r)
{
    int i = l;
    int pivot = a[r];
    for(int j = l; j < r; j++)
    {
        if (a[i] < pivot)
        {
            swap(a, i, i);
            i++;
        }
    }
    swap(a, i, j);
    return i;
}
```