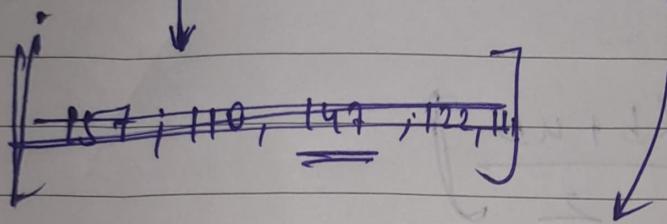


Quick sort :

pivot (random element)

[157, 110, 147, 122, 111, 149, 151, 141, 123, 112, 117, 133]

0 lib ub len(l)



[111, 123, 112, 117, 133]
[110, 147, 122, 111, 149, 157, 151]

3 [141, 123, 112, 117, 133, 110, 1157, 122, 111, 149, 157, 151]

← →
less than more
than

Pivot for left half

[123, 112, 117, 110, 122, 111] [133, [141, 147]] [151, 157]

Pivot

[110, 112, 111, 117, 122, 123] [141, 147] [151, 157]

[110, 112, 111, 117, 122, 123, 141, 147, 151, 157]

quick sort code:

```
#include <stdio.h>
int partition (int a[], int low, int high) {
    int p = a[high], i = low - 1, j;
    for (j = low; j < high; j++) {
        if (a[j] < p) {
            i++;
            int temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    return i + 1;
}
```

y

```
int temp = a[i+1];
a[i+1] = a[high];
a[high] = temp;
return i+1;
```

y

```
void quicksort (int a[], int low, int high) {
    if (low < high) {
        int pi = partition (a, low, high);
        quicksort (a, low, pi - 1);
        quicksort (a, pi + 1, high);
    }
}
```

y.

```
int main() {  
    int n, a[100], i;  
    scanf ("%d", &n);  
    for (i = 0; i < n; i++) scanf ("%d", &a[i]);  
    quicksort(a, 0, n - 1);  
    for (i = 0; i < n; i++) printf ("%d ", a[i]);
```