

AMRITA VISHWA VIDYAPEETHAM
CHENNAI CAMPUS

Second Year. B.TECH

(Computer Science and Engineering)

(DAA Lab Work)

Name: M CYNTHIA SHREE

RollNo.: CH.SC.U4CSE24110

Department: CSE-B

Academic Year: 2025-26

1. Quick Sort with all Cases

CODE:

```
//ch.sc.u4cse24110
//quick sort with all cases
#include <stdio.h>
#include <stdlib.h>

void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int partition(int a[], int l, int r) {
    int pivot = a[r];
    int i = l - 1;

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

    swap(&a[i+1], &a[r]);
    return i + 1;
}

int pivot_first(int l, int r) {
    return l;
}

int pivot_last(int l, int r) {
    return r;
}

int pivot_random(int l, int r) {
    return l + rand() % (r - l + 1);
}
```

```

void quicksort(int a[], int l, int r, int (*pickPivot)(int,int)) {
    if(l < r) {
        int p = pickPivot(l, r);
        swap(&a[p], &a[r]);
        int pi = partition(a, l, r);
        quicksort(a, l, pi - 1, pickPivot);
        quicksort(a, pi + 1, r, pickPivot);
    }
}

int main() {
    int a[] = {157,110,147,122,111,149,151,141,123,112,117,133};
    int n = sizeof(a)/sizeof(a[0]);

    //quicksort(a, 0, n-1, pivot_first);
    //quicksort(a, 0, n-1, pivot_last);
    quicksort(a, 0, n-1, pivot_random);

    for(int i = 0; i < n; i++)
        printf("%d ", a[i]);

    return 0;
}

```

OUTPUT:

```

C:\Users\mrgns\Desktop\daa\task5>gcc quicksortcases.c -o qsc
C:\Users\mrgns\Desktop\daa\task5>.\qsc
110 111 112 117 122 123 133 141 147 149 151 157

```

WORKING:

Quick Sort: Last element Pivot

Array is

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

swap (157, 110)

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

swap (157, 122)

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

swap (147, 111)

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

swap (157, 123)

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

swap (147, 112)

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

swap (149, 117)

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

swap (151, 133)

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

< 133

> 133

pivot = 117

pivot = 151

left

array:

swap (122, 111)

(117)

[110, 111, 122, 123, 112, 117]

swap (122, 112)

[110, 111, 112, 123, 122, 117]

swap (123, 117)

[110, 111, 112, 117, 122, 123] \Rightarrow sorted

23, 112, 133, 157]

112, 133, 157]

3, 157]

1, 157]

7

Right array: (151)

swap (153, 147)

[141, 147, 157, 149, 151]

swap (157, 149)

[141, 147, 149, 157, 151]

swap (157, 151)

[141, 147, 149, 151, 157] \Rightarrow sorted

merge them

Final sorted array:

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

First element pivot (157)

swap it to last (157, 133)

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

all elements less than 157 so no inner swap
final pivot stays

Pivot = 133

swap (133, 117)

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

swap (147, 122)

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

Swap(147, 111)

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

Swap(147, 123)

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

Swap(149, 112)

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

Swap(151, 133)

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

~~continue recursively~~ now pivot = 117

Swap(117, 112)

[112, 110, 122, 111, 123, 117]

Swap(122, 111)

[112, 110, 111, 122, 123, 117]

Swap(122, 117)

[112, 110, 111, 117, 123, 122]

pivot = 112

Swap(112, 110)

[110, 112, 111]

Swap(111, 110)

[110, 111, 112]

pivot = 123

Swap(123, 122)

[122, 123]

now all sorted so merge

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

Random Pivot
swap (111, 133) (111)

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

swap (157, 110)

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

swap (157, 111)

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

Random pivot = 122 swap to 157 then

swap (147, 122)

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

swap (157, 117)

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

swap (133, 122)

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

swap (157, 117)

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

swap (133, 122)

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

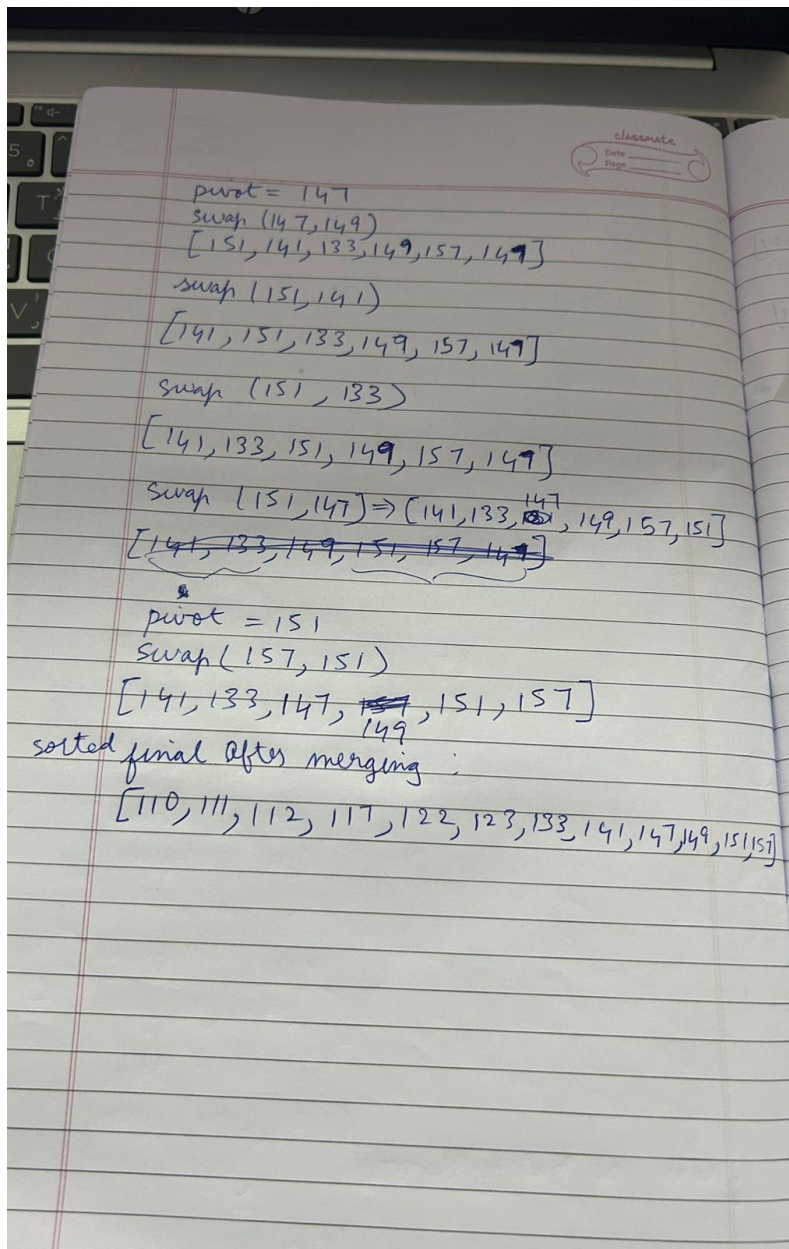
sorted

pivot = 123

swap it to last then

swap (149, 123)

[123, 151, 141, 133, 147, 157, 149]



TIME COMPLEXITY: $O(n \log n)$ on average and best case, $O(n^2)$ in the worst case due to unbalanced partitions.

SPACE COMPLEXITY: Due to recursion stack, worst case is $O(n)$.