29/4/21

Q> Bubble sort and selection sort time complexity programs.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void swap (int *x, int *y) {
    int temp;
    temp = (*x);
    (*x) = (*y);
    (*y) = temp;
}
void bubble (int a[], int n) {
    int i = 0, j = 0;
    int swapped;
    while (i != n - 1) {
        swapped = 0;
        for (j = 0; j < n - i - 1; j++) {
            if (a[j] > a[j + 1]) {
                swap (&a[j], &a[j + 1]);
                swapped = 1;
            }
        }
        i++;
        if (swapped == 0) { return; }
    }
}
```

```
void selection(int a[], int n) {
    int i, j, min;
    for(i = 0; i < n-1; i++) {
        min = i;
        for(j = i+1; j < n; j++) {
            if(a[j] < a[min]) {
                min = j;
            }
        } swap(&a[min], &a[i]);
    }
}
void display(int a[], int n) {
    int i;
    for(i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }
}
int main() {
    int a[5000], i, c, n = 0, j, k;
    printf("1. Selection sort \n2. Bubble sort");
    scanf("%d", &i);
    printf("Enter no. of times u want to
        sort : ");
    scanf("%d", &j);
    for(n; n < j; n+1) {
```

```
printf("Enter size : ");
scanf("%d", &K);
for(c = 0; c < K; c++) {
    a[c] = rand();
}

double ss_time = 0.0;
clock_t begin = clock();
if(i == 1) { selection(a, K); }
else if(i == 2) { bubble(a, K); }
else { exit(0); }
display(a, K);
clock_t end = clock();
ss_time += (double)(end - begin) / CLOCKS_PER_SEC;
printf("n = %d : %f", K, ss_time);
}

return 0;
}
```