Brute force

Change the programming language to C

q2.

```
#include <stdio.h>

void sort(int *arr, int n) {
    for(int i = 0; i < n; i++) {
        int min_val = arr[i];
        int min_idx = i;
        for(int j = i + 1; j < n; j++) {
            if(arr[j] < min_val) {
                min_val = arr[j];
                min_idx = j;
            }
        }
        int temp = arr[i];
        arr[i] = arr[min_idx];
        arr[min_idx] = temp;
    }
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    sort(arr, n);

    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

q3.

```
#include <stdio.h>

void sort(int *arr, int n) {
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n - 1; j++) {
            if(arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    sort(arr, n);

    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

q5.

```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>

// Complete the cmp function to solve the problem
bool cmp(int* a, int* b) {
    double distA = sqrt(a[0]*a[0] + a[1]*a[1]);
    double distB = sqrt(b[0]*b[0] + b[1]*b[1]);

    if(distA > distB) {
        return true;
    } else if(distA == distB) {
        if(a[0] > b[0]) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

void sort(int arr[][2], int n) {
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n - 1; j++) {
            if(cmp(arr[j], arr[j + 1])) {
                // Swap logic
                for(int k = 0; k < 2; k++) {
                    int temp = arr[j][k];
                    arr[j][k] = arr[j + 1][k];
                    arr[j + 1][k] = temp;
                }
            }
        }
    }
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n][2];
    for(int i = 0; i < n; i++) {
        scanf("%d %d", &arr[i][0], &arr[i][1]);
    }

    sort(arr, n);

    for(int i = 0; i < n; i++) {
        printf("%d %d\n", arr[i][0], arr[i][1]);
    }

    return 0;
}
```

q6.
N and N^2