

Esercitazione 8

Programmazione Avanzata ed Elementi di Ingegneria del Software

Esercizio 1:

Scrivere una funzione di ordinamento che riordini un vettore di punti secondo la loro distanza dall'origine degli assi. Un punto è un dato di tipo:

```
struct Point {  
    float x;  
    float y;  
}
```

La funzione deve ordinare il vettore usando l'`insertion sort` per i primi `N` elementi del vettore e il `selection sort` per gli elementi rimanenti.

Opzionalmente: valutare il caso del classificatore `KNearestNeighbors`, che classifica un punto come appartenente ad una determinata classe in base alle distanze tra i suoi vicini.¹ In relazione alla complessità computazionale, serve ordinare tutto il vettore o esistono altre strategie?

Esercizio 2:

Scrivere una funzione che, date due matrici in ingresso, determini quale delle due ha somma magica maggiore. Nel caso in cui una delle due matrici non sia un quadrato magico, la funzione deve ritornare un codice di errore.²

¹ https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

² https://it.wikipedia.org/wiki/Quadrato_magico

Esercizio 3:

Data la funzione di ordinamento offuscata riportata in basso, identificare l'algoritmo che implementa:

```
void f(int *a, int v0) {
    int v1, v2;
    if (v0 > 1) {
        f(a, v0 - 1);
        v1 = *(a + v0 - 1);
        v2 = v0 - 2;
        for(; v2 > -1; v2--) {
            if (*(a + v2) <= v1) { break; }
            else { *(a + v2 + 1) = *(a + v2); }
        }
        *(a + v2 + 1) = v1;
    }
}
```