

## Préparation aux oraux CCINP n°5

*k plus proches voisins*

Dans le code accompagnant ce TP, nous fournissons une structure C

```
struct pt { int class; double x; double y; double dist }
typedef struct pt point;
```

- Q1.** Écrire une fonction `double distance(point* p1, point* p2)` calculant la distance entre les points `p1` et `p2` (pour la norme euclidienne de  $\mathbb{R}^2$ ).
- Q2.** Écrire une fonction `void update_distance(point* tab[], int t, point* p)`; prenant en argument un tableau `tab` de points de taille `t` et mettant à jour leur champ `dist` avec la distance à `p`.
- Q3.** Écrire une fonction `void show_distance(point* tab[], int t)` qui affiche les distances des points.
- Q4.** Écrire une fonction `void sort(point* tab[], int t)` triant un tableau `tab` de points de taille `t` par `y` croissante. On pourra utiliser le tri bulle dont l'algorithme est rappelé ci-dessous.

**Entrée.** Un tableau  $T$  de taille  $n$ .

**Sortie.** Le tableau  $T$  trié.

**Pour**  $i$  allant de  $n - 1$  à  $0$  faire

**Pour**  $j$  allant de  $i$  à  $n - 1$  faire

**Si**  $T[j] > T[j + 1]$  alors

            Échanger  $T[j]$  et  $T[j + 1]$

**Fin si**

**Fin pour**

**Fin pour**

**Retourner**  $T$

**Fig. 1.** Algorithme du tri bulle

- Q5.** Écrire une fonction `int knn(point* p, int k, point* tab[], int t)` appliquant l'algorithme des  $k$  plus proches voisins (d'où le nom  $k$ -NN) et renvoyant la classe supposée de `p`.
- Q6.** Expliquer ce qu'est la classification supervisée. Donner d'autres algorithmes permettant la classification supervisée.
- Q7.** Quelle influence a le paramètre  $k$  dans l'algorithme des  $k$  plus proche voisins ?