

ENSAE TD noté, mardi 26 octobre 2022

1

On souhaite calculer pour chaque rue de France le médecin le plus proche (environ 100.000 médecins généralistes). On suppose qu'on dispose des coordonnées en deux dimensions des rues X et de celles des médecins Y .

1) Tirer deux matrices aléatoires pour X et Y , de tailles différentes. La fonction retourne deux matrices de deux colonnes : `def tirage_alea(n_rues, n_med):`

2) Calculer toutes les distances entre X et Y . On ne pourra pas utiliser la fonction `cdist` (pairwise distances) : `def pairwise_distance(X, Y):`

3) Ecrire un test unitaire qui vérifie le résultat lorsque X et Y sont la matrice identité 2×2 ? (`def test_distance():`)

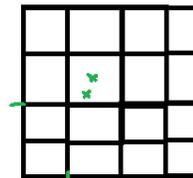
4) Quel est le coût de l'algorithme (en nombre d'opérations) en fonction des dimensions de X et Y ?

5) Ecrire une fonction qui retourne l'indice du médecin le plus proche pour chaque rue `def plus_proche_medecin(X, Y):`

6) On crée une grille 10x10 pour quadriller l'espace. Ecrire une fonction qui calcule les coordonnées des points dans la grille pour plusieurs points. On propose la signature suivante où `p_min` et `p_max` sont les coins inférieur et supérieur de la grille. La fonction doit vérifier le test unitaire ci-dessous.

```
def grille(p, p_min, p_max, n=10):

def test_grille():
    r = grille(numpy.array([[0.41, 0.71], [0.39, 0.99]]),
               p_min=numpy.array([0, 0]),
               p_max=numpy.array([1, 1]), n=10)
    assert r.tolist() == [[0.4, 0.7], [0.3, 0.9]]
```



7) Ecrire une fonction qui crée un dictionnaire où les clés sont les coordonnées d'une case de la grille, les valeurs, l'ensemble des points dans cette case. Cette fonction doit vérifier le test unitaire ci-dessous. Il faudra choisir `pmin` et `pmax` judicieusement.

```
def map_grille_indices(x, n=10):

def test_map():
    x = numpy.arange(8).reshape((-1, 2))
    assert map_grille_indices(x, 2) == {(0.0, 0.0): [0, 1], (0.5, 0.5): [2], (1.0, 1.0): [3]}
```

8) Modifier la fonction `pairwise_distance` pour ne calculer que les distances des points dans la même case.

9) Mesurer le temps pour deux matrices X et Y de dimensions 1000×2 et 100×2 . Laquelle est plus rapide et pourquoi? On pourra notamment comparer les coûts algorithmiques. Imaginez une configuration dans laquelle les deux fonctions ont approximativement le même coût.

10) On revient au problème initial : trouver le médecin le plus proche. On utilise cette fonction plus rapide dans la fonction `def plus_proche_medecin(X, Y):`. Mais ça ne retourne pas toujours le bon résultat. Pourquoi?