

Identification de polygones

vendredi 7 avril 2017 14:39

1)

```
from itertools import combinations
for p1, p2 in combinations(polygones, n = 2):
    if p1 == p2:
        print(p1 + " est identique à " + p2)
```

2)

```
T = {}
for p in polygones:
    sommets = p.nombre_sommets
    if sommets not in T:
        T[sommets] = []
    T[sommets].append(p)
for sous_polygones in T.values():
    question1(sous-polygones)
```

3) On peut utiliser les aires et les angles.

Elimination de recouvrements

vendredi 7 avril 2017 14:43

4) Pour un segment sur une droite d'équation $y = ax + b$, on peut utiliser comme clé de hachage le couple (a, b)

```
6)
T = defaultdict(int)
for s in segments:
    for valeur, p in zip(s.points,
(1, -1))
        T[p] += valeur
sommets = sorted(T.keys())
cpt = 0
entree = None
for s in sommets:
    cpt += T[sommet]
    if cpt == 1:
        if entree is None:
            entree = s
    else:
        if entree is not None:

resultats.append(segment(entree, s))
        entree = None
```

Bilan Dictionnaires

vendredi 7 avril 2017 15:15

8)

Dictionnaire	Insert	Suppr	Recherche	Voisins	Boucle sur tous
ABR AVL	$\log n$	$\log n$	$\log n$	$\log n$	n
Vecteur Liste chaînée	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Vecteur trié Liste triée	$O(n)$	$O(n)$	$\log n$	1	n
Sorted container	$\sqrt[3]{n}$	$\sqrt[3]{n}$	$\log n$	1	n
Vecteur indicé par les clefs Table de hachage	1	1	1	n	n

9)

1 : Liste chaînée

2 : ABR + Table de hachage