

3 Jeu de frites - Code Python (40 points)



Kevin est allé dîner chez McDonald's avec son ami. Quelques minutes après avoir commandé, Kevin a reçu sa commande et est allé s'asseoir à une table en attendant que son ami ait à son tour la sienne. Cependant, Kevin va devoir attendre longtemps, car les employés confus du McDonald's ont accidentellement donné les articles de la commande de son ami à d'autres clients qui avaient commandé les mêmes choses, puis ont supprimé sa commande de leur liste, croyant qu'elle avait été complétée.

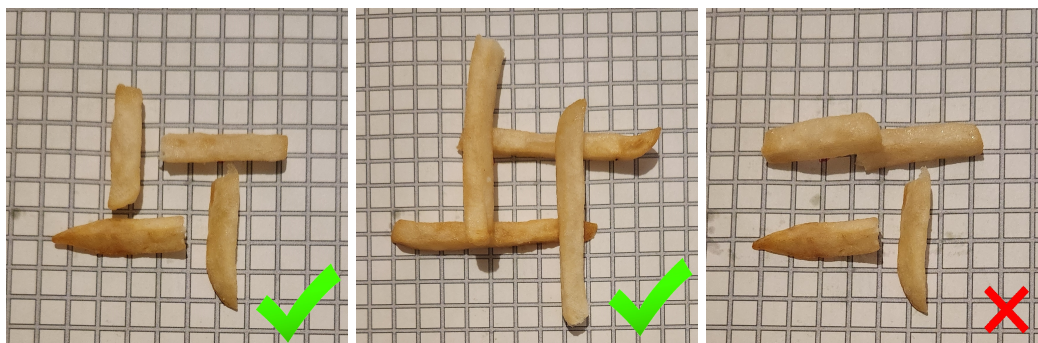
Pendant que Kevin attend après son ami qui doit se battre¹ avec les employés du McDonald's, il décide de faire passer le temps en jouant à un petit jeu avec ses frites². Son jeu consiste à fixer $n \in \mathbb{N}$ points sur une grille (qu'il dessine au verso du napperon en papier qu'il a dans son cabaret). Ensuite, Kevin prend n frites qui sont **toutes de la même longueur** $2k$ (où $k \in \mathbb{N}$), puis il essaie de placer une frite sur chaque point, avec les restrictions suivantes :

- Le centre de la frite est placé sur le point de la grille. Autrement dit, comme la frite est de longueur $2k$, chaque extrémité de la frite est à distance k du point.
- Les frites sont placées soit à l'horizontale, soit à la verticale.
- Deux frites qui ont la même orientation **ne peuvent pas** se toucher ou se chevaucher (et ce même si deux frites ne se touchent qu'à leurs extrémités)
- Deux frites qui ont des orientations différentes (donc une verticale et une horizontale) **peuvent** se toucher ou se croiser.

¹Au sens figuré. Malgré sa frustration, son ami reste tout de même respectueux envers les employés.

²Malgré qu'il n'est généralement pas recommandé de jouer avec de la nourriture.

Avec les points $(0,0)$, $(0,3)$, $(3,0)$, $(3,3)$, et des frites de demi-longueur $k = 2$, Kevin pourrait placé ses frites à l'horizontale sur $(0,0)$ et $(3,3)$, et à la verticale sur $(0,3)$ et $(3,0)$ (image à gauche). Par contre, s'il avait plutôt voulu mettre la frite de $(0,3)$ à l'horizontale, cela n'aurait pas été valide, puisque la condition selon laquelle deux frites ayant la même orientation ne peuvent pas se toucher ne serait pas respectée (image à droite). Remarquez aussi que la solution (de l'image à gauche) aurait aussi fonctionné avec des frites de longueur arbitrairement grande, et pas seulement avec $k = 2$. Par exemple, sur l'image au centre, on a une solution avec $k = 4$.



Avec les points $(0,0)$, $(0,4)$, $(7,0)$, $(7,4)$, $(10,0)$, $(10,4)$ et des frites de demi-longueur $k = 4$, Kevin pourrait placer ses frites de la façon suivante :



Cependant, avec $k = 5$ (ou plus), les points ci-haut n'admettraient pas de solution.

Étant donné les points de la grille, Kevin se pose la question suivante : quelle serait la plus grande valeur de k pour laquelle il existe une façon de placer les frites de sorte à respecter toutes les contraintes de son jeu? Aidez Kevin en rédigeant un algorithme (en Python) qui va calculer le k maximal (sur le premier exemple, la réponse aurait été ∞ , tandis que sur le deuxième exemple, la réponse aurait été 4).

L'entrée est donnée dans un fichier .txt. La première ligne contient la nombre de points n . Les n lignes suivante contiennent chacun des n points, données comme deux entiers séparés par un espace. Vous pouvez supposer que chacun des n points est unique.

En sortie, vous devez produire un fichier .txt où la première ligne contient la valeur de la demi-longueur k optimale. S'il serait possible de placer des frites de longueur arbitraire (et donc qu'il n'y a pas de k maximal), le fichier devrait contenir "infini". S'il est impossible de placer des frites pour tous les $k \geq 1$, le fichier devrait contenir 0.

Indice : Essayer d'abord d'implémenter une fonction la plus efficace possible qui, pour un k fixé, vérifie s'il est possible de placer les frites de demi-longueur k sur les points. Inspirez-vous ensuite d'un algorithme vu en cours pour trouver le k optimal, en faisant plusieurs appels à votre fonction.

Exemples :

Exemple 1 :

```
4
0 0
0 3
3 0
3 3
--> doit donner
infini
```

Exemple 2 :

```
6
0 0
0 4
7 0
7 4
10 0
10 4
--> doit donner
4
```

Exemple 3 :

```
6
0 0
0 1
1 0
1 1
2 0
2 1
--> doit donner
0
```

Code

Votre code doit lire un fichier spécifié en premier argument qui contient les détails du problème et écrire la solution dans un fichier spécifié en second argument.

Exemple d'appel :

```
python3 frites.py input1.txt output1.txt
```

Un fichier *test_frites.py* et des fichiers d'entrées vous sont fournis pour vous aider à tester votre code.

Exemple d'appel :

```
python3 test_frites.py
```

Remise

Compléter le fichier *frites.py* fournis, et ne remettre **uniquement** que ce fichier. Ne **pas** remettre le fichier *test_frites.py*, ou n'importe quel autre fichier de test.

Dans votre fichier pdf : discutez brièvement de la complexité de votre algorithme.