Entrées-Sorties en Python

I Entrée standard 1

I Entrée standard

Pour gérer l'entrée d'information d'un utilisateur en `Python`, le plus simple est de passer par l'entrée standard. Il s'agit d'une sorte de fichier ouvert où on lit des informations lignes par lignes.

Quand on lance le programme, soit l'utilisateur va effectivement taper des entrées ligne par ligne, soit l'entrée va provenir d'un fichier que l'on va rediriger sur l'entrée standard.

On utilise ainsi la fonction `input()` qui renvoie une ligne lue et passe à la suite.

Par exemple, dans le problème (HIGHWAYS)[https://www.spoj.com/problems/HIGHWAYS/] on nous donne un format de la forme :

```
2
4 2 1 4
1 2 5
3 4 5
4 4 1 4
1 2 5
2 3 5
3 4 5
4 2 6
```

où on vous dit que la première ligne est le nombre d'instances de problèmes à résoudre. On peut le stocker dans une variable `n` entière ainsi :

```python
n = int(input())
```

Ensuite, on trouve n problèmes comportant une ligne : `sommets aretes x y` de quatre entiers suivis de `aretes` lignes de la forme `a b w` indiquant une arête de poids `w` entre `a` et `b`.

Pour lire trois entiers (x, y, z) écrits sous la forme `x y z` on va utiliser `split` :

```python
x, y, z = map(int, input().split())
```

Donc ici, on pourrait faire une fonction `distance(g, x, y)` qui prend un graphe et deux sommets et renvoie la plus petite distance de `x` à `y`.

Pour lire et afficher le résultat, il suffit alors d'écrire :

```python
n = int(input())

for _ in range(n):
  sommets, aretes, x, y = map(int, input().split())
  g = [ [] for _ in range(sommets) ]
  for _ in range(aretes):
    a, b, w = map(int, input().split())
    g[a].append( (b, w) )
    g[b].append( (a, w) )
```

```python
d = distance(g, x, y)
if d == float('inf'):
  print('NONE')
else:
  print(d)
```

Python