

# Quelques conseils

BCPST 2 2019-2020

## 1 Choix des structures de données, bibliothèques

Une même notion (graphe, arbre) admet de nombreuses représentations. Par exemple, les graphes peuvent être représentés par des matrices, des listes. Le choix de cette structure peut simplifier la syntaxe de votre projet. De même, l'utilisation d'une bibliothèque peut aussi être pratique.

Il est important de pouvoir respecter les quelques règles énoncées ci-dessus.

1. Toutes les fonctions écrites, toutes nouvelles structures ou bibliothèques utilisées doivent être comprises.
2. Pouvoir justifier l'utilisation d'une structure donnée (surtout si celle-ci ne semble pas naturelle). Par exemple, pour les graphes, dans le cas où les graphes qui vous intéressent ont peu d'arêtes, il vaut mieux considérer une représentation par des listes d'adjacence.
3. Si une fonction vous semble compliquer à rédiger, décomposer celle-ci en différentes parties plus simples à écrire.

## 2 Lisibilité du code

Il est important d'écrire un code lisible par tous. Voici quelques conseils pour cela. Commenter le code peut aussi aider à cela. En particulier, mentionner quels sont les types de vos arguments et quels sont les types des valeurs de retour et/ou quels sont les modifications effectuées par une fonction donnée.

1. Les noms de variables, de fonctions doivent être choisis de manière appropriés.
2. Suivant votre problème, on peut donner un nom à des constantes utilisés tout au long de votre code.
3. Commenter vos fonctions. En Python, il existe deux types de commentaires. L'une avec `"""`, est utilisée pour expliquer ce que fait une fonction donnée, indiquer le type des paramètres en entrées et les valeurs de sorties. L'autre avec `#` est en général utilisé pour un commentaire d'une ligne.
4. Garder des conventions : par exemple, les constantes sont écrites en majuscules, les fonctions en minuscules. En cas de nom à rallonge, on peut utiliser un `_`.
5. On écrit en général au maximum sur 80 colonnes.