

---

CPE Lyon - 3IRC Année 2021/2022  
Structures de données et algorithmes avancés  
**Séance 5 - Arbres**

---



### Exercice 1. Arbre binaire de recherche (TD)

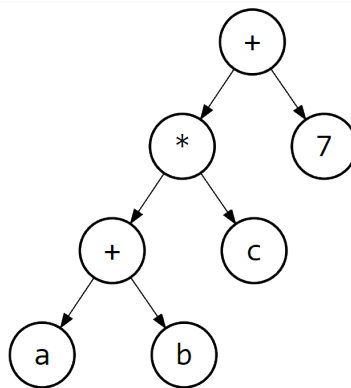
Construisez l'arbre binaire de recherche à partir de la liste suivante (prise dans cet ordre) :

25,60,35,10,5,20,65,45,70,40,50,55,30,15

Construisez ensuite l'arbre obtenu en supprimant successivement les valeurs 35 et 25.

### Exercice 2. Arbre d'expression / Notation polonaise inverse (TD / TP)

Les arbres sont utilisés dans les interpréteurs et les compilateurs pour valider et évaluer les expressions écrites dans un langage. Par exemple, un arbre peut servir à vérifier qu'une formule mathématique est correctement écrite (c'est-à-dire *syntactiquement correcte*). Dans un tel arbre, appelé **arbre d'expression**, les nœuds internes correspondent aux *opérateurs* et les feuilles correspondent aux *opérandes*. Par exemple, la formule  $(a + b) \times c + 7$  peut être représentée par l'arbre d'expression suivant :



1. Construisez l'arbre d'expression de la formule  $((5 + z) / - 8) * (4^2)$
2. Ecrivez le résultat du parcours infixe de l'arbre obtenu. S'agit-il exactement de la même formule qu'à la question précédente? Que faudrait-il rajouter?

La *notation polonaise inverse (NPI)* ou *notation post-fixée* consiste à écrire l'opérateur **après** les opérandes. Par exemple, en NPI, l'expression  $3 + 4$  s'écrit  $3\ 4\ +$ . Cette notation a (entre autres) l'avantage de pouvoir écrire sans ambiguïté une formule mathématique *sans utiliser de parenthèses* : par exemple, la formule  $((1 + 2) \times 4) + 3$  peut s'écrire en NPI  $1\ 2\ +\ 4\ \times\ 3\ +$ .

3. Quel parcours d'arbre permet de retrouver une formule écrite en NPI?
4. Comment adapter l'algorithme de ce parcours pour calculer le résultat d'une expression mathématique écrite en NPI? Implémentez cet algorithme en Python.

### Exercice 3. Une question d'entretien chez Facebook (TP)

L'*opérateur ternaire* ? : est un opérateur que l'on retrouve dans de nombreux langages de programmation, et qui permet d'écrire de manière concise des conditions. Ainsi, au lieu d'écrire `if a then b else c`, on écrira plus simplement `a?b:c`.

**Exercice** : implémentez une structure d'arbre, puis construisez l'arbre correspondant à une expression ternaire (pouvant être imbriquée) donnée sous forme de chaîne de caractères.

Exemple :

Expression en entrée = `a?b:c`

Arbre à construire :

```
      a
     / \
    b   c
```

Expression en entrée = `a?b?c:d:e`

Arbre à construire :

```
      a
     / \
    b   e
   / \
  c   d
```

### Exercice 4. Arbre binaire de recherche (TP)

Implémentez un type de données appelé **ABR** permettant de réaliser les opérations classiques (insertion, suppression, recherche) d'un arbre binaire de recherche **équilibré**.

### Exercice 5. Akinator (TP)

Vous connaissez sans doute Akinator, le Génie du Web (<https://fr.akinator.com/>) ? Dans ce jeu, Akinator, à travers un certain nombre de questions, doit essayer de deviner l'identité du personnage que l'utilisateur a en tête. Le but de cet exercice est d'écrire une version simplifiée d'Akinator. Pour cela, le programme interagit avec l'utilisateur pour construire un arbre de questions et de noms d'animaux. Voici un exemple d'exécution :

```
Pensez-vous à un animal ? o
Est-ce un oiseau ? n
Quel est le nom de l'animal ? chien
Quelle question distinguerait un chien d'un oiseau ? Peut-il voler
Si l'animal était un chien, quelle serait la réponse à cette question ? n
```

```
Pensez-vous à un animal ? o
Peut-il voler? n
Est-ce un chien ? n
Quel est le nom de l'animal ? chat
Quelle question distinguerait un chat d'un chien ? Aboie-t-il
Si l'animal était un chat, quelle serait la réponse à cette question ? n
```

```
Pensez-vous à un animal ? o
Peut-il voler? n
Aboie-t-il? o
Est-ce un chien ? o
J'ai trouvé !
```

```
Pensez-vous à un animal ? n
```