

# Parcours en largeur d'un arbre binaire

| LIV David & ROUSSEL Matthieu

## Fonctions implémentées

❖ Toutes les fonctions dans le fichier.h sur le e-learning ont été ajoutées.

❖ Seules quelques fonctions auxiliaires ont été ajoutées, les voici :

```
int powe(int n, int p);
```

➤ Fonction qui calcule tout simplement  $n^p$

```
int hauteur_arbre(Arbre a);
```

➤ Fonction qui calcule la hauteur d'un arbre **COMPLET**

```
void free_arbre(Arbre a); // Fonction annexe
```

➤ Fonction qui libère l'espace mémoire d'un arbre

```
int build_filiforme_aleatoire(int h, Arbre *a, int val);
```

➤ Fonction annexe qui appelle la fonction pour construire un arbre filiforme.

```
void Libere_Liste(Liste *l, Liste lst);
```

➤ Fonction qui libère l'espace mémoire d'une liste

## Les difficultés rencontrées

❖ Les fonctions filiformes qui avaient des problèmes de retour de valeurs lorsqu'on a fait les tests sur PLaTon.

❖ La fonction `parcours_largeur_naif()` qui ne renvoyait pas les bonnes valeurs sur PLaTon.

## La répartition du travail au sein du binôme

### ❖ 1) Préliminaires

- Question 1 -> Question 3.b : LIV David
- Question 3.c : ROUSSEL Matthieu

### ❖ 2) Deux méthodes pour réaliser un parcours en largeur

- Question 2.1 : ROUSSEL Matthieu
- Question 2.2 : ROUSSEL Matthieu & LIV David

### ❖ 3) Comparaison des méthodes

- ROUSSEL Matthieu

### ❖ 4) Rapport & Docstrings

- LIV David

## Questions de la fin de la partie "Comparaison des méthodes".

### ❖ 3.a

- La fonction *parcours\_largeur\_V2* semble être plus efficace car :
  - Elle parcourt moins de noeud que le *parcours\_largeur\_naif\_V2*.  
Par exemple sur un arbre complet de hauteur 20 :
    - *parcours\_largeur\_V2* = 2097151 noeud visités
    - *parcours\_largeur\_naif\_V2* = 6324158 noeud visités

### ❖ 3.b

- La fonction *parcours\_largeur* à une complexité égale à  $O(n)$ .

❖ 3.c

- La fonction `parcours_largeurs_naif` sur un arbre filiforme à une complexité égale à  $O(n^2)$ .

❖ 3.d

- La fonction `parcours_largeurs_naif` sur un arbre complet à une complexité égale à  $O(n^2)$ .