
Cours IA

Python pour l'IA

3^{ème} Année Lic.

Dr. A. LAKHFIF
2022/2021

Python : Exercices

Exercice 01 : Vérifier qu'un nombre est un Palindrome

Ecrire un programme python qui vérifie que le nombre en entrée est un palindrome.
Exemple : 1234321, 3457543, 84548 sont des nombres palindromes

Exercice 02 : Extraire les chiffres d'un nombre (dans l'ordre inverse)

Ecrire un programme python qui extrait dans l'ordre inverse chaque chiffre à partir d'un nombre entier.

Par exemple, pour l'entier **7536**, la sortie doit être **"6 3 5 7"**, avec une espace entre les chiffres.

Exercice 03 : Afficher 05 lignes d'étoiles (*)

Afficher 05 lignes d'étoiles (*), qui forment une pyramide orientée vers le bas :

**

*

Exercice 04 : Calculer le volume d'une sphère de rayon =6
Le volume d'une sphère est donné par l'équation : $V = \frac{4}{3} \times \pi \times r^3$.

Exercice 05 : Afficher la date et l'heure actuelle
Afficher la date et l'heure actuelle.

Recherche En Profondeur (DFS)

La recherche en profondeur (DFS) est un algorithme fondamental de parcours de graphe.

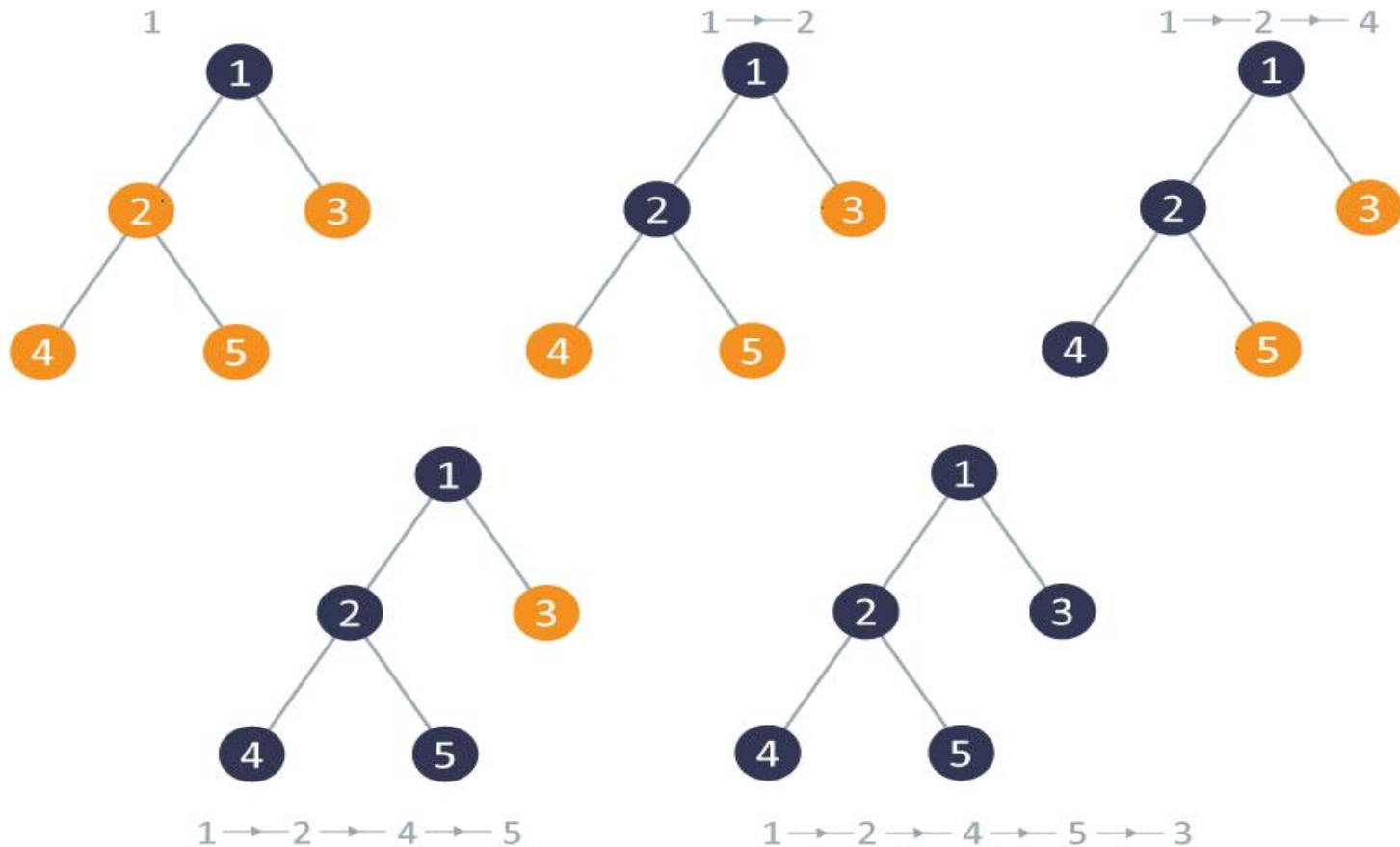
- L'algorithme DFS traverse en explorant chaque chemin le plus loin possible avant de devoir revenir en arrière (**Backtracking ou Retour en arrière**)..
- cette propriété permet de mettre en œuvre l'algorithme succinctement à la fois sous forme itérative et récursive.

Etapes:

- Ajouter le nœud de départ dans la pile et marquez comme visité.
- Tant qu'il y a un nœud dans la pile:
 1. **Prendre le nœud** au dessus de la pile.
 2. **Ajouter dans la pile tous les voisins disponibles dans l'ordre, noter le parent et marquer comme visité.**

Recherche En Profondeur (DFS)

DFS



Implémentation récursive

```
explorer(graphe G, sommet s)
    marquer le sommet s
    afficher(s)
    pour tout sommet t voisin du sommet s
        si t n'est pas marqué alors
            explorer(G, t);
```

Implémentation récursive

- ✓ parcours en profondeur d'un graphe G

parcoursProfondeur(graphe G)

pour tout sommet s du graphe G

si s n'est pas marqué *alors*

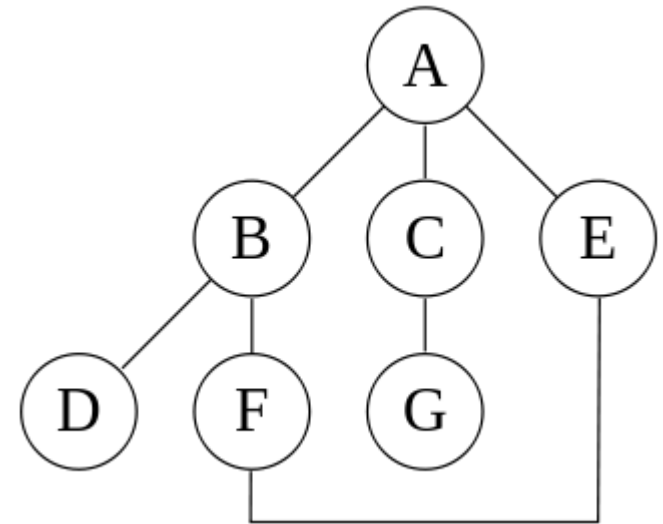
explorer(G, s)

Implémentation récursive

✓ **parcours en profondeur d'un graphe G**

➤ **Avec marquage des sommets :**

on aura l'ordre de visite suivant: A, B, D, F, E, C, G.



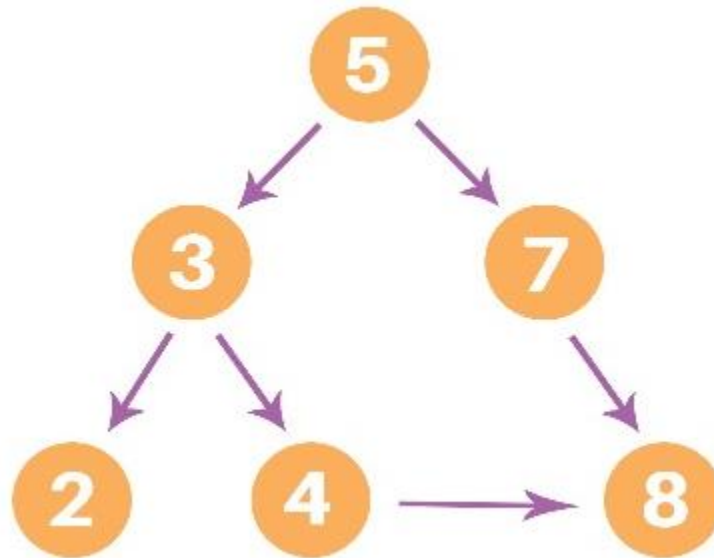
➤ **Sans marquage des sommets :**

on aura l'ordre de visite des sommets suivants : A, B, D, F, E, A, B, D, F, E, ...,etc indéfiniment, car l'algorithme ne peut sortir de la boucle A, B, D, F, E et n'atteindra donc jamais C ou G.

Python : Application

Exercice :

- ✓ parcourir en profondeur le graphe G suivant:



Python : Application

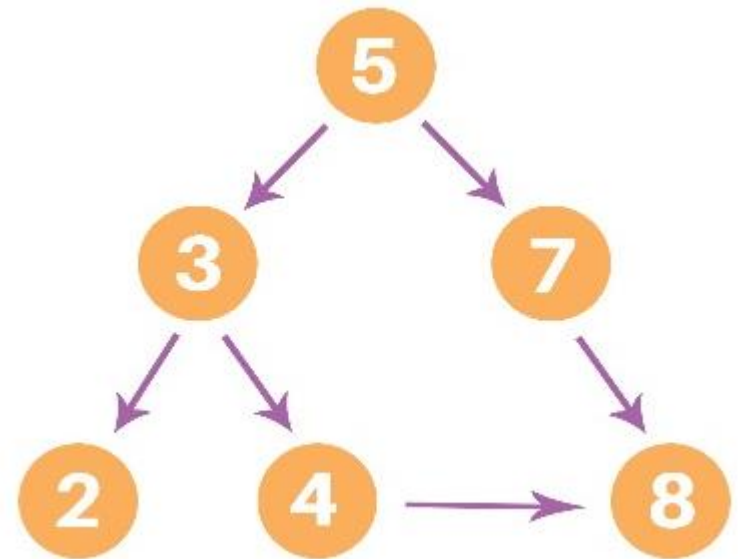
Exercice :

✓ parcourir en profondeur le graphe G suivant:

■ Astuces :

✓ Créer un dictionnaires des adjacents

```
graph = {  
    '5' : ['3','7'],  
    .....  
    '2' : [],  
    .....  
}
```



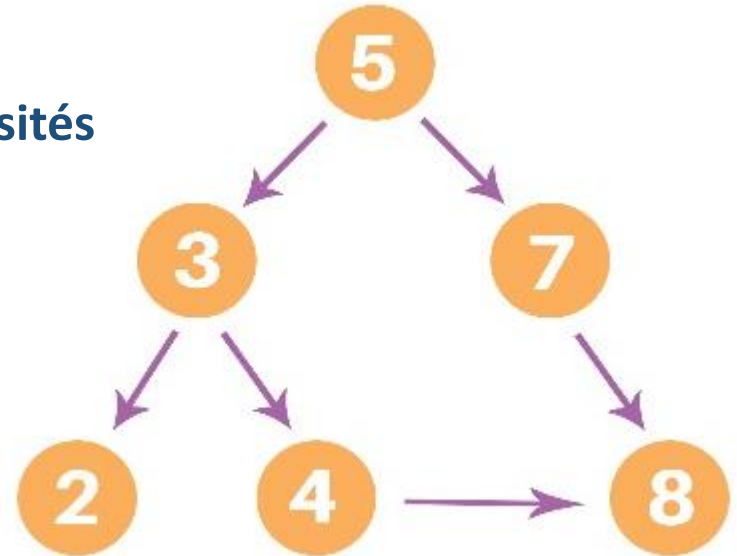
Python : Application

✓ parcourir en profondeur le graphe G suivant:

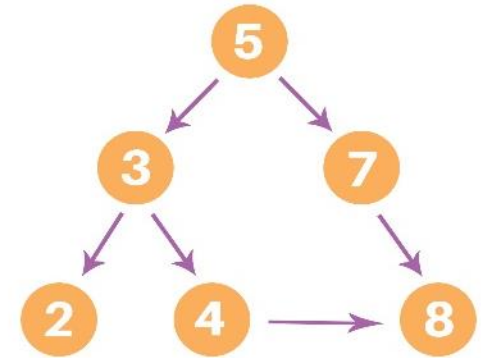
- **Astuces :**

- ✓ Créer un ensemble `set()` pour les nœuds visités

`visited = set()`



Python : Application



✓ parcourir en profondeur le graphe G suivant:

■ Pseudo -code :

DFS(G, u)

u.visited = true

for each v ∈ G.Adj[u]

if v.visited == false

DFS(G,v)

init() {

For each u ∈ G

u.visited = false

For each u ∈ G

DFS(G, u)

}