

Arbres

December 4, 2023

Soit un ensemble fini T , un élément $r \in T$ et une fonction $p : T \setminus \{r\} \rightarrow T$. Si pour tout $x \in T \setminus \{r\}$, il existe un entier k tel que $p^k(x) = r$ (où nous avons noté $p^k(x) = p(p^{k-1}(x))$ avec $p^0(x) = x$), alors T muni de la fonction p est un **arbre**. Nous disons que r est la **racine** de T et que $p(x)$ est le **père** de x et que x est le **fil** de $p(x)$. Par exemple, avec l'ensemble $T = \{1, 2, \dots, 9\}$ et $p(x) = \lfloor \frac{x}{2} \rfloor$, le père de $x = 8$ sera $p(8) = 4$, celui de 5 sera $p(5) = 2$, et la racine de l'arbre sera 1.

Plus généralement, pour $k \geq 1$, $p^k(x)$ est un **ancêtre** de x et x est un **descendant** de $p^k(x)$. Un élément sans descendant est une **feuille** de l'arbre. Dans l'exemple précédent, 8 est une feuille car il n'existe aucun nombre dans l'ensemble $T = \{1, 2, \dots, 9\}$ tel que $p(x) = 8$. De plus 2 est un ancêtre de 8 car $p^2(8) = 2$.

Etant donné $x \in T$, sa **hauteur** sera 0 si x est une feuille, et cela sera la valeur maximale de k tel qu'il existe x et y avec $x = p^k(y)$. En particulier, la hauteur de T est -1 si $T = \emptyset$, et sinon cela sera la hauteur de sa racine r . Par exemple, dans l'exemple précédent, la hauteur est de 3 car $p^3(9) = 1$.

Vous allez construire des fonctions permettant la représentation et la manipulation d'arbres. Un arbre sera implémenté sous la forme d'un tableau.

1. Implémenter la fonction père décrit en exemple $p(x) = \lfloor \frac{x}{2} \rfloor$.
2. Ecrire une fonction `matriceTree(T,r,p)` avec T l'ensemble fini, r une racine et p une fonction père, calcule tous les éléments de l'arbre et retourne une matrice M de dimension $|T| \times |T|$ où, pour tout $i, j \in T$:

$$M_{i,j} = \begin{cases} 1 & \text{si } i = p(j) \\ -1 & \text{si } j = p(i) \\ 0 & \text{sinon} \end{cases}$$

M sera appelée **matrice d'adjacence** de (T, r, p) .

3. Testez votre algorithme avec `T = list(range(1,20))` et la fonction père défini en question 1.
4. Testez votre algorithme avec la fonction père $p(x) = x - 1$.

5. Ecrire une fonction qui, étant donné une matrice d'adjacence M , calcule la racine de l'arbre.
6. Ecrire une fonction `profondeur(T,r,p,x)` qui calcule la profondeur d'un élément x dans l'arbre.
7. Ecrire une méthodes récursive qui calcule la hauteur d'un arbre (T, r, p) . Faites une méthode utilisant la matrice d'adjacence, et une ne l'utilisant pas.
8. Testez votre algorithme sur l'exemple de la question 3.