

2020-AT-01 Arbres digitaux

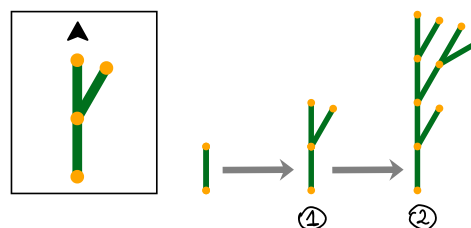
6yo–8yo: —	8yo–10yo: —	10yo–12yo: —	12yo–14yo: —	14yo–16yo: —	16yo–19yo: hard
<i>Answer Type:</i> multiple choice with images					
<i>Computer Science Areas:</i> <input type="checkbox"/> algorithms and programming <input type="checkbox"/> data structures and representations			<input type="checkbox"/> computer processes and hardware <input type="checkbox"/> communication and networking <input type="checkbox"/> interactions, systems and society		
<i>Computational Thinking Skills:</i> <input type="checkbox"/> abstraction <input type="checkbox"/> algorithmic thinking			<input type="checkbox"/> decomposition <input type="checkbox"/> evaluation <input type="checkbox"/> pattern recognition		

Body

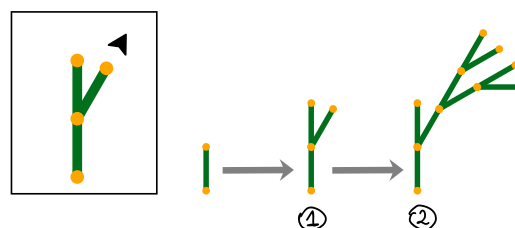
Un arbre digital est fait de tronçons d'arbre comme celui-ci: .

Il pousse étape par étape d'après une règle de croissance définie.

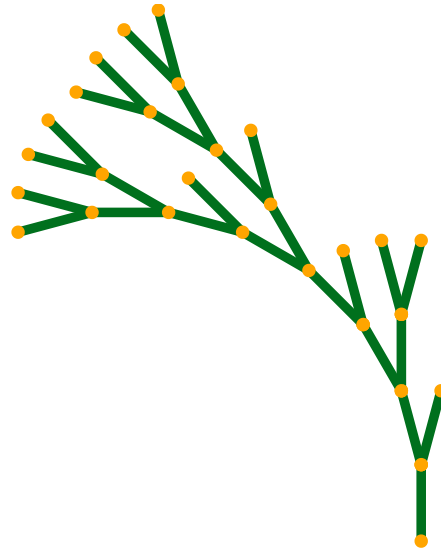
La règle de croissance indique de quelle manière un tronçon est remplacé par une structure composée de nouveaux tronçons. Lors de chaque étape, chaque tronçon est remplacé de cette manière. Une pointe de flèche indique où et dans quelle direction les tronçons sont assemblés.



Les exemples à droite montrent deux règles de croissance et les deux premières étapes de croissance correspondantes.



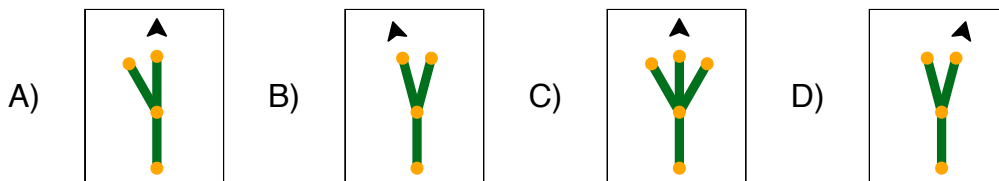
L'arbre suivant a poussé en trois étapes:



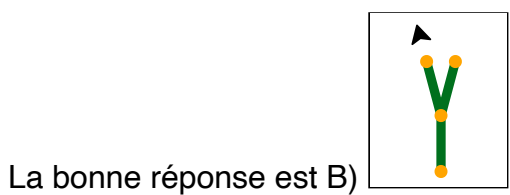
Question/Challenge

D'après quelle règle de croissance l'arbre digital a-t-il poussé?

Answer Options/Interactivity Description



Answer Explanation



Règle de croissance

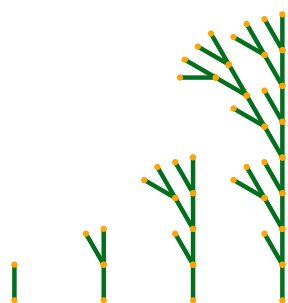
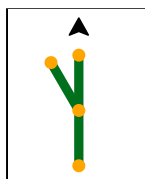
Trois étapes de croissance

Description

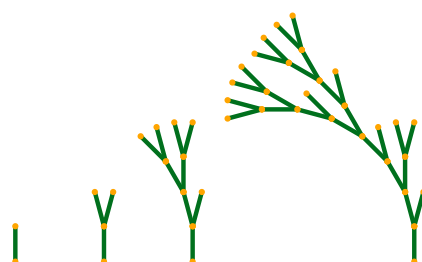
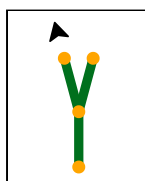
Règle de croissance

Trois étapes de croissance

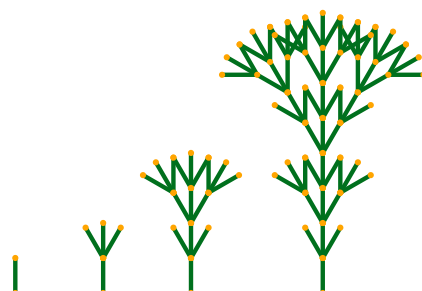
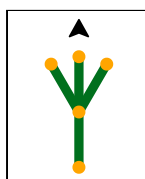
Description



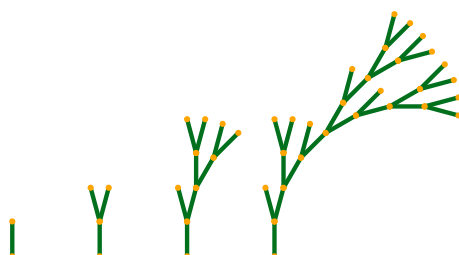
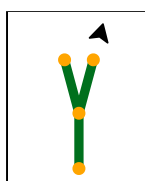
Le reste de l'arbre est toujours ajouté à la branche dirigée tout droit vers le haut. Il se forme ainsi un tronc droit avec des branches toutes orientées à gauche.



Le reste de l'arbre est toujours ajouté à la branche dirigée vers la gauche et le haut. L'arbre est donc penché vers la gauche.



Le reste de l'arbre est toujours ajouté à la branche du milieu. Les deux branchements à gauche et à droite génèrent une structure régulière et symétrique.



Le reste de l'arbre est toujours ajouté à la branche dirigée vers la droite et le haut. L'arbre est donc penché vers la droite.

It's Informatics

Dans cet exercice, on voit comment l'application répétée de règles simples peuvent générer des structures compliquées. De telles figures formées de parties semblables à la figure complète sont aussi appelées des *fractales*. En informatique, on recourt très souvent aux fractales, par exemple pour créer des paysages ou des effets spéciaux pour des films.

En biologie, on utilise ce qu'on appelle des *systèmes de Lindenmayer* (ou *L-systèmes*) pour simuler la croissance des plantes. Ce système génère également des fractales. Dans cet

exercice, nous avons vu des exemples très simples de L-systèmes.

Les arbres dans cet exercice sont générés par l'application d'une règle sur chaque tronçon d'arbre, puis à nouveau sur chaque tronçon ainsi généré, et ainsi de suite. De tels procédés sont appelés *récur­sifs*. Le concept de la récursivité est important en informatique. Grâce à la récursivité, il est possible de décrire de manière très simple beaucoup de choses compliquées.

Keywords and Websites

- Fractale: <https://fr.wikipedia.org/wiki/Fractale>
- L-système: <https://fr.wikipedia.org/wiki/L-Système>, <http://paulbourke.net/fractals/lsys/>
- Récursivité: <https://fr.wikipedia.org/wiki/Récursivité>

Wording and Phrases

(Not reported from original file)

Comments

(Not reported from original file)

Contributors

- Wilfried Baumann, baumann@ocg.at, Austria (author, graphics)
- Gerald Futschek, gerald.futschek@tuwien.ac.at, Austria (author)
- Alisher Ikramov, ikramov.alisher@list.ru, Uzbekistan (contributor)
- Vipul Shah, v.shah@tcs.com, India (contributor)
- Valentina Dagienė, valentina.dagiene@mif.vu.lt, Lithuania (contributor)
- Wolfgang Pohl, pohl@bwinf.de, Germany (contributor, graphics)
- Susanne Datzko, susanne.datzko@inf.ethz.ch, Switzerland (contributor, graphics)
- Fabian Frei, fabian.frei@inf.ethz.ch, Switzerland (contributor, translation from English into German)
- Elsa Pellet, elsa.pellet@gmx.net, Switzerland (translation from German into French)

Support Files

- *.svg by Wilfried Baumann, modified by Susanne Datzko and Wolfgang Pohl
- python_code/** by Wilfried Baumann

License



Copyright © 2020 Bebras – International Contest on Informatics and Computer Fluency. This work is licensed under a Creative Commons Attribution – ShareAlike 4.0 International License. <https://creativecommons.org/licenses/by-sa/4.0/>