

بسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

پروژه پژوهشی درس نظریه زبان ها و ماشین ها

L-systems

استاد راهنما: دکتر مهدی دولتی

فاطمه حمدی

بهار 1403

فهرست مطالب

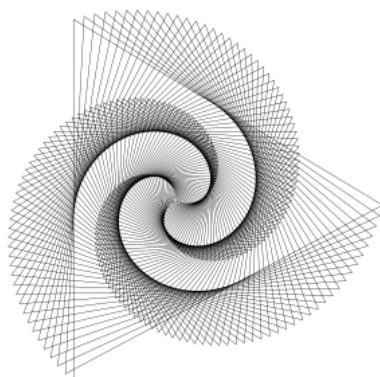
1	مقدمه
1	ساختار L-systems
2	سیستم های L چگونه کار می کنند؟
3	L-systems : جایی که طبیعت = ریاضی !
3	مثال اول: algae
4	مثال دوم: fractal (binary) tree
5	مثال سوم: Cantor set
5	مثال چهارم: Koch curve
6	مثال پنجم: Sierpinski triangle
8	مثال ششم: dragon curve
9	مثال هفتم: fractal plant
10	انواع L-system
11	جمع بندی
12	منابع

مقدمه

سیستم L یا سیستم Lindenmayer یک سیستم بازنویسی موازی و نوعی [formal language](#) است. یک سیستم L شامل الفبا از نمادهایی است که می‌توان از آنها برای ساخت رشته‌ها استفاده کرد، مجموعه‌ای از قوانین تولید که هر نماد را به رشته‌های بزرگ‌تری از نمادها گسترش می‌دهد، یک رشته «axiom» اولیه که از آن شروع به ساخت می‌شود، و مکانیزمی برای تبدیل رشته‌های تولید شده به ساختارهای هندسی. سیستم‌های L در سال 1968 توسط آریستید لیندن مایر ([Aristid Lindenmayer](#))، زیست‌شناس نظری و گیاه‌شناس مجارستانی در دانشگاه [Utrecht](#) معرفی شدند و توسعه یافتند. لیندن مایر از سیستم‌های L برای توصیف رفتار سلول‌های گیاهی و مدل‌سازی فرایندهای رشد گیاه استفاده کرد.

ساختار L-systems

ماهیت بازگشتی قوانین سیستم L منجر به خود-شباهت ([self-similarity](#)) می‌شود و در نتیجه، اشکال فراکتال مانند ([fractal](#)) به راحتی با یک سیستم L توصیف می‌شوند. مدل‌های گیاهی و اشکال ارگانیک با ظاهر طبیعی به راحتی قابل تعریف هستند، زیرا با افزایش سطح بازگشت، شکل به آرامی رشد می‌کند و پیچیده‌تر می‌شود. سیستم‌های Lindenmayer در نسل حیات مصنوعی نیز محبوب هستند.



A spiral drawn with an iterative turtle graphics algorithm



A Fibonacci word fractal

گرامر سیستم‌های L بسیار شبیه [semi-Thue grammar](#) می‌باشد. سیستم‌های L در حال حاضر معمولاً به عنوان parametric L-systems شناخته می‌شوند که به عنوان یک tuple تعریف می‌شوند.

$$G = (V, \omega, P)$$

V (the alphabet) مجموعه‌ای از نمادها است که شامل عناصر قابل جایگزینی و غیرقابل جایگزینی است.

ω (start, axiom or initiator) رشته‌ای از نمادها از V است که وضعیت اولیه سیستم را تعریف می‌کند.

P مجموعه ای از قوانین تولید یا تولیدات (production rules or productions) است که نحوه جایگزینی متغیرها را با ترکیبی از ثابت ها و سایر متغیرها تعریف می کند. یک product از دو رشته predecessor و successor تشکیل شده است.

سیستم های L چگونه کار می کنند؟

فرآیند با یک رشته اولیه به نام axiom شروع می شود. به آن به عنوان بذری فکر کنید که سیستم از آن رشد می کند. هر نماد در رشته طبق قوانین تولید جایگزین می شود. این مرحله برای تعداد مشخصی از تکرارها تکرار می شود. با هر تکرار، رشته پیچیده تر می شود زیرا قوانین به طور مکرر اعمال می شوند. رشته نهایی سپس به یک ساختار هندسی ترجمه می شود که اغلب به صورت یک الگوی گیاهی یا فراکتال تجسم می شود.

قواعد گرامر سیستم L به طور مکرر از حالت اولیه شروع می شود. تا آنجا که ممکن است بسیاری از قوانین به طور همزمان، در هر تکرار اعمال می شوند. این که هر تکرار تا حد امکان قوانین زیادی را به کار می گیرد، یک سیستم L را از یک formal language تولید شده توسط یک formal grammar، که در هر تکرار فقط یک قانون را اعمال می کند، متمایز می کند. اگر قرار بود قوانین تولید فقط یک بار در یک زمان اعمال شوند، به سادگی یک رشته در یک زبان تولید می شد و همه این توالی از برنامه ها زبان مشخص شده توسط گرامر را تولید می کردند. با این حال، رشته هایی در برخی از زبان ها وجود دارد که اگر گرامر به عنوان یک سیستم L به جای یک ویژگی زبان در نظر گرفته شود، نمی توان آنها را تولید کرد. برای مثال، فرض کنید یک قانون $S \rightarrow SS$ در یک دستور زبان وجود دارد. اگر تولیدات یک به یک انجام شوند، با شروع از S، می توانیم ابتدا SS و سپس با اعمال مجدد قانون، SSS دریافت کنیم. با این حال، اگر همه قوانین قابل اجرا در هر مرحله اعمال شوند، مانند یک سیستم L، نمی توانیم این فرم جمله را دریافت کنیم. در عوض، مرحله اول به ما SS می دهد، اما مرحله دوم قانون را دو بار اعمال می کند و به ما SSSS می دهد. بنابراین، مجموعه رشته هایی که توسط یک سیستم L از یک گرامر معین تولید می شود، زیرمجموعه ای از formal language تعریف شده توسط گرامر است، و اگر یک زبان را به عنوان مجموعه ای از رشته ها در نظر بگیریم، به این معنی است که یک سیستم L داده شده در واقع زیرمجموعه ای از formal language تعریف شده توسط گرامر سیستم L است.

اگر هر قانون تولید فقط به یک نماد منفرد و نه به همسایگان آن اشاره داشته باشد، سیستم L مستقل از متن (Context-free L-systems) است. بنابراین، سیستم های L مستقل از متن توسط یک دستور زبان مستقل از متن (context-free grammar) مشخص می شوند. اگر یک قانون نه تنها به یک نماد، بلکه به همسایگان آن وابسته باشد، به آن یک سیستم L وابسته به متن (context-sensitive L-system) می گویند.

اگر دقیقاً یک تولید برای هر نماد وجود داشته باشد، سیستم L قطعی گفته می شود. (deterministic context-free L-system که معمولاً DOL system نامیده می شوند). اگر چندین سیستم L وجود داشته باشد، و هر کدام با احتمال خاصی در طول هر تکرار انتخاب شوند، آنگاه یک سیستم L تصادفی (stochastic L-system) است.

L-systems : جایی که طبیعت = ریاضی !

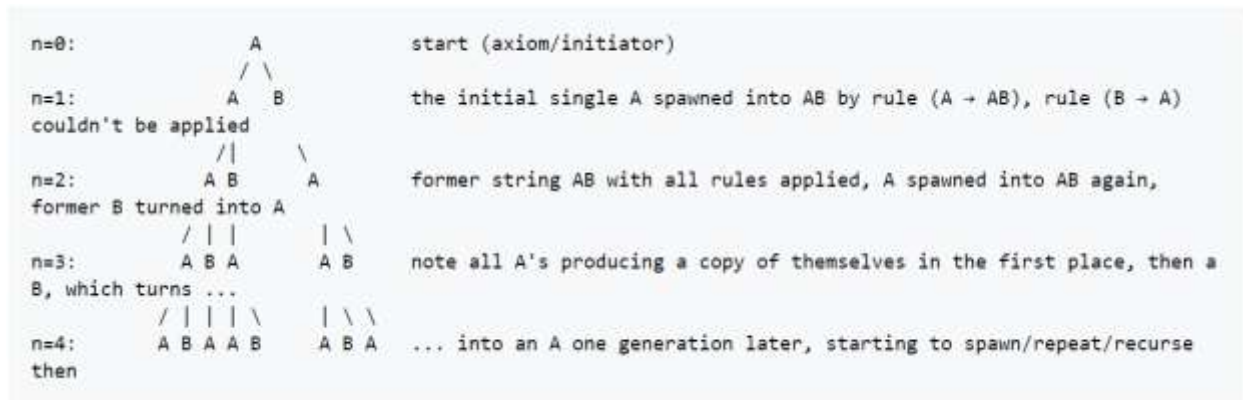
مثال اول: algae

سیستم L ارجینال لیندن مایر برای مدل سازی رشد جلبک ها.

- **variables** : A B
- **constants** : none
- **axiom** : A
- **rules** : $(A \rightarrow AB)$, $(B \rightarrow A)$

which produces:

- $n = 0$: A
- $n = 1$: AB
- $n = 2$: ABA
- $n = 3$: ABAAB
- $n = 4$: ABAABABA
- $n = 5$: ABAABABAABAAB
- $n = 6$: ABAABABAABAABABAABABA
- $n = 7$: ABAABABAABAABABAABAABABAABAABABAABAABABAABAAB



اگر طول هر رشته را بشماریم، دنباله فیبوناچی معروف از اعداد را به دست می آوریم:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

برای هر رشته، اگر موقعیت k -امین را از انتهای سمت چپ رشته بشماریم، مقدار آن با قرار گرفتن مضربی از نسبت طلایی (golden ratio) در بازه $(k-1, k)$ تعیین می شود. نسبت A به B به همین ترتیب به میانگین طلایی همگرا می شود.

این دنباله یک **locally catenative sequence** است، زیرا $G(n) = G(n-1)G(n-2)$. $G(n)$ نشان دهنده نسل n ام است.

مثال دوم: fractal (binary) tree

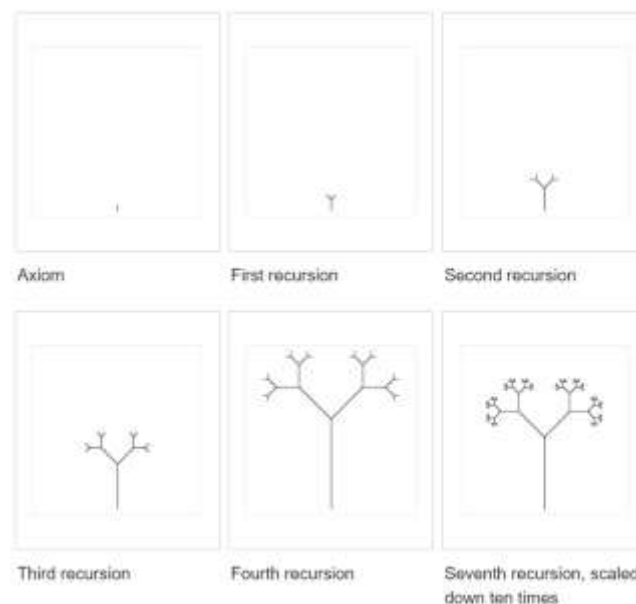
- **variables** : 0, 1
- **constants**: "[", "]"
- **axiom** : 0
- **rules** : $(1 \rightarrow 11), (0 \rightarrow 1[0]0)$

Iterations:

- **Axiom**: 0
- **1st Recursion**: 1[0]0
- **2nd Recursion**: 11[1[0]0]1[0]0
- **3rd Recursion**: 1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

رشته با هر تکرار به سرعت در اندازه و پیچیدگی رشد می کند. با استفاده از گرافیک لاک پشت، رشته را می توان به یک درخت فراکتال ترجمه کرد:

- **0**: Draw a line segment ending in a leaf.
- **1**: Draw a line segment.
- **[**: Push current position and angle, turn left 45 degrees.
- **]**: Pop position and angle, turn right 45 degrees.



عملیات **push** و **pop** از یک پشته **LIFO** برای ذخیره و بازیابی موقعیت و زاویه لاک پشت استفاده می کند و امکان ایجاد ساختارهای انشعاب را فراهم می کند. این مثال نشان می دهد که چگونه قوانین ساده می توانند الگوهای پیچیده و طبیعی را از طریق کاربرد بازگشتی ایجاد کنند.

مثال سوم: Cantor set

- **variables** : A B
- **constants** : none
- **start** : A {starting character string}
- **rules** : $(A \rightarrow ABA)$, $(B \rightarrow BBB)$
Let A mean "draw forward" and B mean "move forward".



این [Cantor's fractal set](#) را روی یک خط راست حقیقی \mathbb{R} ایجاد می کند.

Koch curve: مثال چہارم

گونه ای از منحنی koch که فقط از زاویه های قائم استفاده می کند.

- **variables** : F
- **constants** : + -
- **start** : F
- **rules** : $(F \rightarrow F + F - F - F + F)$

Here, F means "draw forward", + means "turn left 90°", and - means "turn right 90°" (see [turtle graphics](#)).

 $n = 0:$

F

—

$$n = 1:$$
$$F + F - F - F + F$$
 $n = 2:$

F+F-F-F+F+F+F-F-F-F+F-F-F-F+F-F-F+F-F-F+F+F+F-F-F-F+F

 $n = 3:$

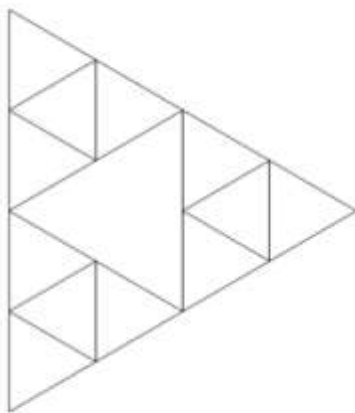
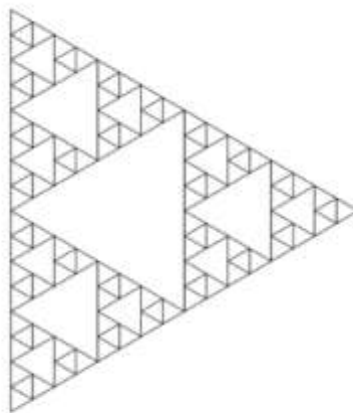
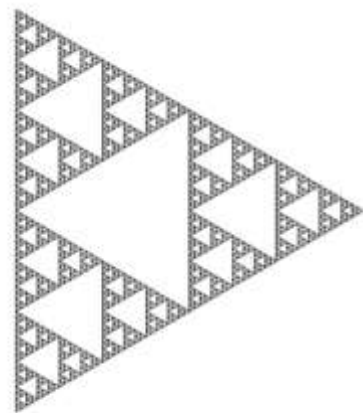
F+F-F-F+F+F-F-F+F-F-F+F-F-F+F-F-F+F+F+F-F-F+F+

F+F-F-F+F+F-F-F+F-F-F-F-F-F-F-F-F-F-F-F-F-F-F-F-

F+F-F-F+F+F+F-F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F-F-

مثلت Sierpinski با استفاده از یک سیستم L رسم شده است.

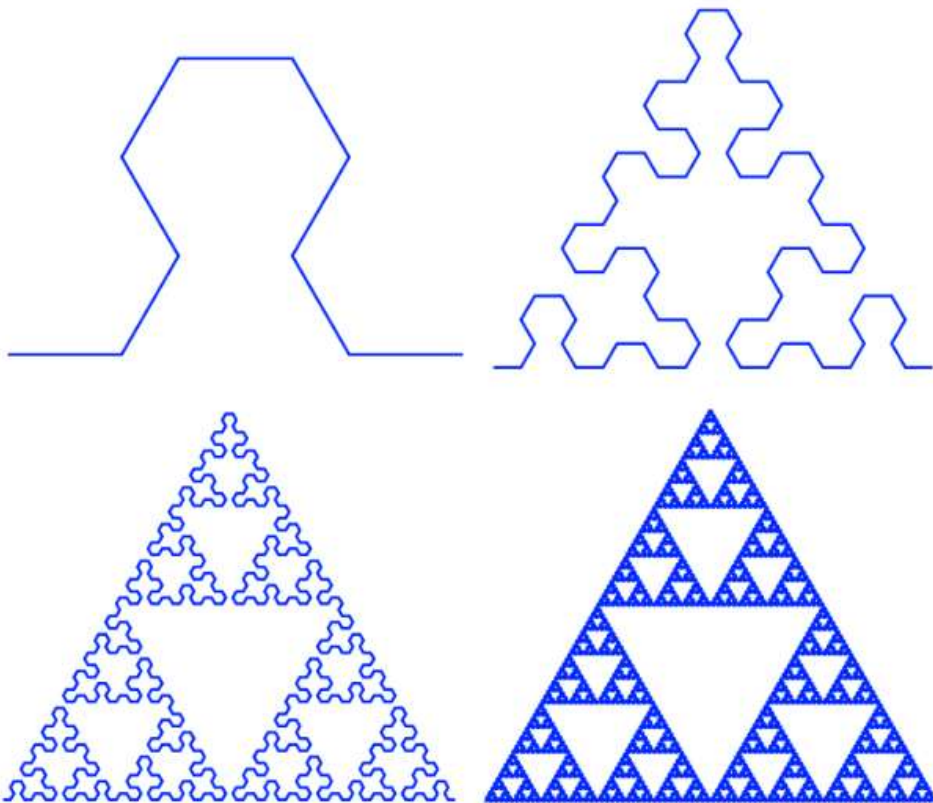
- Here, F and G both mean "draw forward", + means "turn left by angle", and - means "turn right by angle".

 $n = 2$  $n = 4$  $n = 6$

6

- **variables** : A B
- **constants** : + -
- **start** : A
- **rules** : $(A \rightarrow B-A-B)$, $(B \rightarrow A+B+A)$
- **angle** : 60°

Here, A and B both mean "draw forward", + means "turn left by angle", and - means "turn right by angle" (see [turtle graphics](#)).



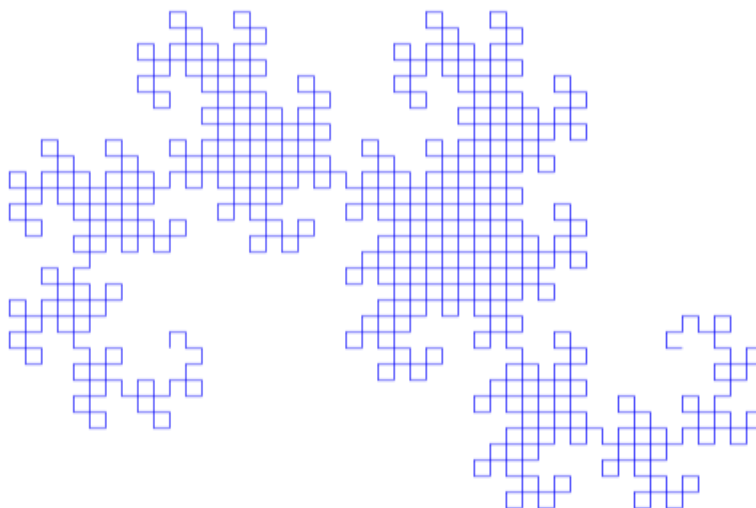
Evolution for $n = 2$, $n = 4$, $n = 6$, $n = 8$

مثال ششم: dragon curve

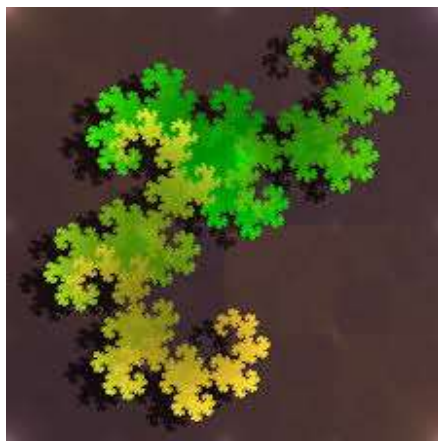
منحنی اژدها (dragon curve) با استفاده از یک سیستم L رسم شده است.

- **variables** : F G
- **constants** : + -
- **start** : F
- **rules** : $(F \rightarrow F+G)$, $(G \rightarrow F-G)$
- **angle** : 90°

Here, F and G both mean "draw forward", + means "turn left by angle", and - means "turn right by angle".



Dragon curve for $n = 10$



مثال هفتم: fractal plant

- **variables** : X F
- **constants** : + - []
- **start** : X
- **rules** : $(X \rightarrow F+[[X]-X]-F[-FX]+X), (F \rightarrow FF)$
- **angle** : 25°

ابتدا باید یک پشته خالی را مقداردهی اولیه شود. این از روش LIFO (آخرین ورود، اولین خروج) برای افزودن و حذف عناصر پیروی می کند. در اینجا، F به معنای "کشیدن به جلو"، - به معنای "25° درجه به راست بپیچید" و + به معنای "25° درجه به چپ بپیچید". X با هیچ عمل ترسیمی مطابقت ندارد و برای کنترل تکامل منحنی استفاده می شود. براکت مربع "[]" مربوط به ذخیره مقادیر فعلی برای موقعیت و زاویه است، بنابراین موقعیت و زاویه را در بالای پشته push می شود، هنگامی که نشانه "[]" مواجه شدیم، پشته، پاپ و موقعیت و زاویه بازنشانی می شود. هر "[]" قبل از هر علامت "[]" می آید.



Fractal plant for $n = 6$

انواع L-system

سیستم های L روی خط حقیقی R:

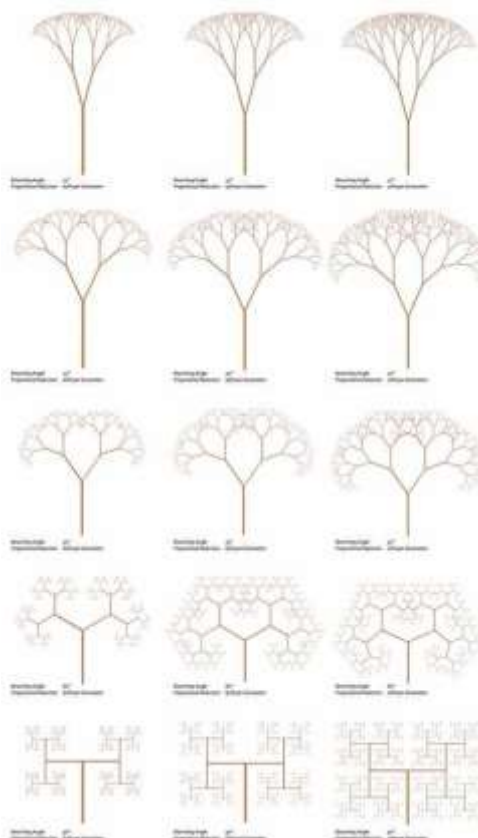
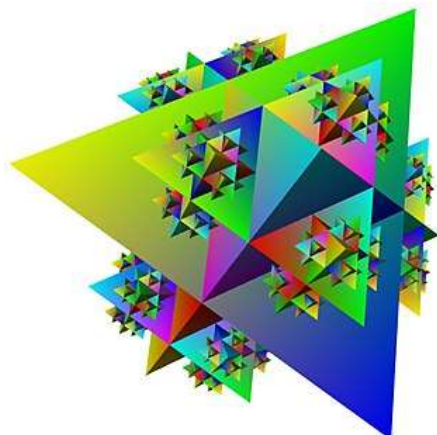
- [Prouhet-Thue-Morse system](#)

سیستم های L معروف در صفحه R2 عبارتند از:

- [space-filling curves](#) ([Hilbert curve](#), [Peano's curves](#), Dekking's church, [kolams](#)),
- median space-filling curves ([Lévy C curve](#), [Harter-Heighway dragon curve](#), Davis-Knuth terdragon),
- tilings ([sphinx tiling](#), [Penrose tiling](#))

جمع بندی

سیستم های L یا سیستم های Lindenmayer ابزاری قدرتمند و همه کاره برای مدل سازی و شبیه سازی فرآیندهای رشد طبیعی هستند. با استفاده از قوانین ساده و بازگشتی، سیستم های L می توانند ساختارهای پیچیده و واقعی ایجاد کنند که هم از نظر علمی ارزشمند و هم از نظر بصری جذاب هستند. کاربردهای آنها در زمینه های مختلفی از زیست شناسی و گرافیک کامپیوتری گرفته تا معماری و هنرهای مولد را در بر می گیرد. از طریق کاربرد تکراری قوانین تولید، سیستم های L می توانند الگوها و فرم های پیچیده ای ایجاد کنند که زیبایی و پیچیدگی طبیعت را تقلید می کند. در نتیجه، آنها همچنان ابزاری ضروری برای محققان، هنرمندان و توسعه دهندگانی هستند که به دنبال درک و تکرار دنیای طبیعی هستند.



منابع

<https://en.wikipedia.org/wiki/L-system>

https://en.wikipedia.org/wiki/Turtle_graphics

<https://www.youtube.com/watch?v=puwhf-404Xc>

<https://en.wikipedia.org/wiki/Fractal>

<https://en.wikipedia.org/wiki/Self-similarity>

copilet

ویدیو ها و تصاویر بیشتر:

<https://www.youtube.com/shorts/SJyZv-B7xs0>

<https://www.youtube.com/shorts/J50xFMS1Htw>

<https://www.youtube.com/watch?v=8UFvXp5-ueQ>

[l-systems images - Search Images \(bing.com\)](#)