

بسم الله الرحمن الرحيم

سری اول تمرینات درس شبیه سازی

حسین محمدی

۹۶۱۰۱۰۳۵

هشدار: این برنامه از کتابخانه turtle و ساختار تابع بازگشتی استفاده می کند و زمان به نسبت زیادی برای اجرا نیاز دارد. (تضمین می شود که stack overflow برای دقت های مانیتور رخ ندهد، اما کتابخانه turtle ساختاری دارد که در این ساختار برای جابه جا کردن قلم از یک طرف به طرف دیگر صفحه نمایش، زمان می برد.)

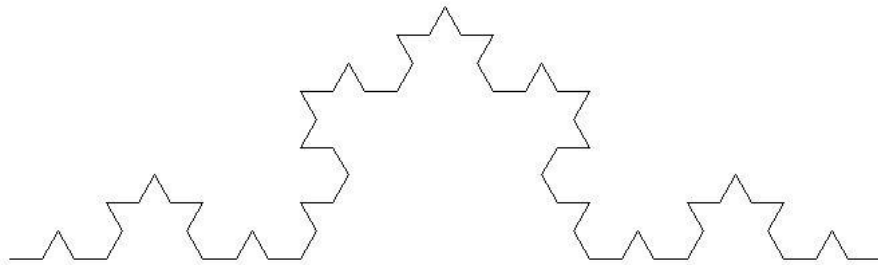
توجه: کد برای اجرا نیاز به دوبار ران کردن دارد.

برای تولید برف دانه ی کخ الگوریتم های متفاوتی هست که بنده در اینجا از الگوریتم بازگشتی و از کتابخانه turtle استفاده کرده ام، استفاده از این کتابخانه این مزیت را دارد که نیازی به دوران و ماتریس دوران ندارد و فقط با دادن زاویه ی مورد نظر، نشانگر بر روی صفحه می چرخد. برخی توابع در کد متعلق به این کتابخانه هستند، و اما الگوریتم به این صورت است:

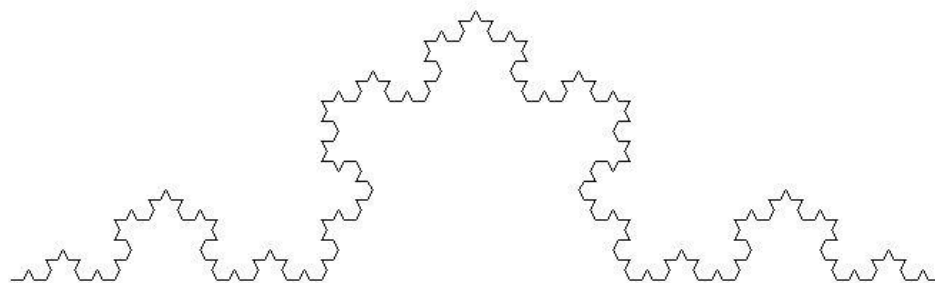
با متغیر `lv`، مراحل را که می خواهیم برف دانه کخ پیشروی کند مشخص می کنیم و متغیر `a`، طول مرحله صفرم است بر حسب پیکسل، که پاره خط اشغال می کند. حال فرض کنید ما کد را با مقادیری ران کنیم، در شرطی که در بدنه ی تابع هست بررسی می شود تا `lv` به صفر برسد و سپس رسم صورت می گیرد، در حقیقت این شرط بررسی می کند که الان `cursor` باید کجای شکل باشد و چند درجه بچرخد یا چقدر پیشروی کند تا شکل صحیح رسم شود. می توان با کمک نمودار درختی شیوه ی کار کد را توضیح داد. (فایل نمودار درختی را در همین فایل زیپ ضمیمه کرده ام)

توجه: کد با کتابخانه `Mathplotlib` هم قابل ترسیم بود، اما این کتابخانه توانایی کار کردن با پیکسل ها را در اختیار ما قرار نمی دهد و نقاط هم با کمک `scatter` از این کتابخانه، اندازه ی بزرگی دارند.

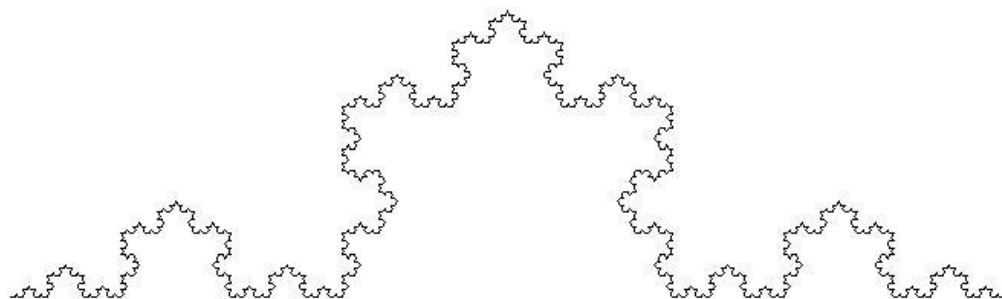
این هم چند خروجی از تابع با مقادیر مختلف `lv`، `a`:



با طول کل ۶۵۰ پیکسل و ۳ مرحله اجرای تابع (مرحله ۳ برف دانه کخ)



با طول کل ۶۵۰ پیکسل و ۴ مرحله اجرای تابع (مرحله ۴ برف دانه کنخ)



با طول کل ۶۵۰ پیکسل و ۵ مرحله اجرای تابع (مرحله ۵ برف دانه کنخ)