

تمرین شبیه سازی سری اول

علی قبله ۹۹۷۱۰۹۹۱۰

سوال ۱:

طبق صحبتی که سر کلاس شد، برای تولید برفدانه کخ نیاز به ۴ تابع داشتیم. ایده اصلی بسیار آسان است. کافیت خط اولیه را به سه قسمت تقسیم کنیم. دو قسمت روی همان خط اولیه و قسمت وسط ۶۰ درجه دوران دارد. برای تشکیل این برفدانه، روی هر خط ۵ نقطه خواهیم داشت که عبارتند از: نقاط ابتدایی و انتهایی — دو نقطه در ۳/۱ ام ها — یک نقطه زاویه دار به سمت بیرون خط

تنها کار باقی مانده وصل نقاط به صورت ۲ به ۲ پشت هم است. برای اینکار نیز از subplot و comprehension list استفاده کردیم و ۲ function با این کارایی را تعریف کردیم. استفاده از subplot برای جمع کردن تعداد زیادی خط در یک plot می باشد. طراحی به صورت مرحله به مرحله نشان داده شده است.

سوال ۲:

ایده دقیقاً همانند سوال اول است. تنها یک ضریب K به علت تغییر زاویه از ۶۰ به ۴۵ تغییر کرده است. البته باید توجه می کردیم که برای بدست آوردن این شکل نقطه وسطی یکبار به سمت بیرون و یکبار به سمت داخل دوران می کردند به عبارتی شکل یکبار ۴۵ درجه و یکبار ۱۳۵ درجه دوران در ادامه هم قرار می گرفت. برای بدست آوردن این یکی در میان، کافی بود متغیری رشد کننده را تعریف کنیم و هربار زوج یا فرد بودن آن را چک کنیم.

طراحی به صورت مرحله به مرحله نشان داده شده است.

سوال ۳ و ۵:

سوال واقعیت نیاز به ایده خاصی ندارد! تنها چالش وصل کردن وسط هر ضلع به همدیگر و حذف کردن مثلث وسطیست. برای اینکار به جای آنکه مثلث وسط را حذف کنیم داخل تمام مثلث های جدید به وجود آمده را به رنگ آبی در آوردیم تا فرکتال تشکیل شود. برای بخش رندوم نیز به همین صورت و طبق توضیح کتاب عمل کردیم.

سوال ۶:

به دست آوردن احتمالات و ضرایب با کمک اینترنت انجام شده است. طراحی به صورت نقطه محور می باشد و هرچه تعداد نقاط بیشتر شود، دقت و طراحی دقیق تر خواهد شد.

سوال ۴:

برای این سوال ابتدا نیاز است که مثلث پاسکال را بدست آوریم. اینکار صرفا نیاز به یک تابع بازگشتی با توجه به تابع پیشین است. پس از مشخص شدن خط بعدی مثلث کافیسست چک کنیم آیا عدد در آرایه فرد است یا خیر. در صورت فرد بودن کافیسست در نقطه ای در صفحه `matplotlib lib` خطی بسیار کوتاه بکشیم. حال با تعریف `subplot` می توانیم تعداد زیادی از این خطوط را در `plot` ذخیره کنیم و یکجا نشان دهیم. فاصله هر خط با خط دیگری وابسته به میزان دقت مثلث می باشد و قابل تغییر است. طول هر خط نیز به همین صورت! (بخش بازگشتی تابع را فرصت نشد در کد بنویسم وگرنه هربار با استفاده از `comprehension list` می توان هر خط مثلث پاسکال را چک و خطوط جدید به شکل افزود تا مثلث تشکیل شود)

****مهم**

لطفا در هنگام اجرای کد به توضیحات اولیه نوشته شده توسط دانشجو توجه کنید. ابتدای تمامی برنامه ها (به جز ۴ امی) تعداد حلقه های پیشنهادی یا رزولوشن پیشنهادی اعلام شده است چرا که زمان زیادی مصرف نشود و یا شکل به خوبی واضح نباشد.