پروژه مبانی احتمال

به نام خدا

دانشگاه علم و صنعت ایران دانشکده ریاضی و علوم کامپیوتر درس: مبانی احتمال

مدرس درس: دکتر فرنوش

نیم‌سال تحصیلی: نیم‌سال دوم تحصیلی

دستورالعمل پروژه پایانی درس مبانی احتمال

عنوان پروژه: طراحی و پیاده‌سازی هسته شبیه‌ساز، تحلیل‌گر آماری و اثبات‌گر قضایای حد و تقریب احتمالی

زبان برنامه‌نویسی مجاز: Python 3.x

ساختار تیم‌ها: حداکثر ۲ نفر

نکته: نمره و تاریخ ارائه پروژه متعاقباً اطلاع‌رسانی خواهد شد.

محدودیت پروژه

استفاده از هرگونه کتابخانه آماده آماری، احتمالی و پردازش داده (نظیر `random` ، `scipy` ، `numpy` ، `pandas` و موارد مشابه) جهت تولید اعداد تصادفی، محاسبه فرمول توزیع‌ها یا آزمون‌های فرضی ممنوع است. تنها کتابخانه‌های مجاز در این پروژه، کتابخانه‌های داخلی و استاندارد `time` (صرفاً جهت دریافت زمان سیستم) و `math` (جهت محاسبات ریاضی پایه‌ای مانند لگاریتم و توان) می‌باشند. تمام منطق ریاضی و شبیه‌سازی باید از صفر توسط اعضای تیم کدنویسی شود.

نیازمندی‌ها و فازهای اجرایی پروژه

فاز ۰: طراحی و پیاده‌سازی موتورهای تولید عدد تصادفی یکنواخت $U(0,1)$

برنامه باید مجهز به دو الگوریتم مجزا برای تولید اعداد شبه‌تصادفی پیوسته در بازه $[0,1)$ باشد. هر دو موتور باید در زمان مقداردهی اولیه (Constructor)، زمان فعلی سیستم را به میکروثانیه دریافت کرده و از آن به عنوان هسته اولیه (Seed) استفاده کنند:

۱. **موتور همنهشتی خطی (Linear Congruential Generator - LCG):** پیاده‌سازی بر اساس فرمول زیر و با پارامترهای استاندارد مشخص شده:

$$X_{n+1} = (48271 \cdot X_n) \pmod{2^{31} - 1}$$

خروجی هر مرحله به صورت $U_{n+1} = \frac{X_{n+1}}{2^{31}-1}$ نگاشت می‌شود.

۲. **موتور Xorshift:** پیاده‌سازی الگوریتم ۳۲ بیتی با استفاده از عملیات‌های بیتی (Bitwise Operators) شامل شیفت و XOR متوالی روی بیت‌های هسته اولیه و نگاشت نهایی به بازه $[0,1)$.

- **بخش مقایسه‌ای فاز صفر:** برنامه در بخش اجرایی خود باید زمان لازم برای تولید ۱,۰۰۰,۰۰۰ عدد تصادفی توسط هر دو موتور را محاسبه، مقایسه و موتور بهینه را معرفی کند.

فاز ۱: پیاده‌سازی متغیرهای تصادفی و توزیع‌های خاص

یک ساختار شیءگرا (OOP) طراحی کنید که شامل کلاس‌های مجزا برای توزیع‌های زیر باشد. هر کلاس باید متدی به نام `generate_sample()` داشته باشد که یک نمونه تصادفی را **صرفاً** با تکیه بر خروجی یکی از موتورهای رندوم فاز قبل تولید کند:

- **توزیع‌های گسسته (خروجی عدد صحیح):**
  - **توزیع برنولی:** با پارامتر احتمال موفقیت p.
  - **توزیع دوجمله‌ای:** با پارامترهای n و p (شبیه‌سازی از طریق مجموع n آزمایش مستقل برنولی).
  - **توزیع هندسی نوع اول:** تعداد کل آزمایش‌ها تا ثبت اولین موفقیت با پارامتر p.
  - **توزیع پواسون:** با پارامتر نرخ وقوع λ (پیاده‌سازی بر اساس الگوریتم ضرب پیشامدهای یکنواخت کِنوت).
- **توزیع‌های پیوسته (خروجی عدد اعشاری):**
  - **توزیع نمایی:** با پارامتر نرخ λ (پیاده‌سازی با روش تبدیل معکوس روابط انباشته).
  - **توزیع نرمال:** با پارامترهای میانگین μ و انحراف معیار σ (پیاده‌سازی با روش تبدیل دوبعدی باکس-مولر).

فاز ۲: تحلیل آماری و ارزیابی خطای تجربی

برنامه باید پتانسیل دریافت نام توزیع، پارامترهای ورودی و عدد M (تعداد دفعات شبیه‌سازی، به عنوان مثال $M = 100,000$) را از کاربر داشته باشد. سپس با فراخوانی متوالی متد تولید نمونه، شاخص‌های زیر را بدون توابع آماده محاسبه کند:

۱. **میانگین تجربی (Empirical Mean):**

$$\bar{X} = \frac{1}{M}\sum_{i=1}^{M} X_i$$

۲. **Variance تجربی (Empirical Variance):** محاسبه واریانس نمونه‌ای با اعمال اصلاح نااریبی بسل:

$$S^2 = \frac{1}{M-1}\sum_{i=1}^{M}(X_i - \bar{X})^2$$

۳. **گزارش خطا:** مقایسه شاخص‌های تجربی حاصل از شبیه‌سازی با مقادیر تئوریک (روابط و فرمول‌های کتابی امید ریاضی و واریانس هر توزیع) و محاسبه درصد خطای مطلق شبیه‌سازی بر اساس فرمول زیر:

$$\text{Error } \% = \left|\frac{\text{Theoretical} - \text{Empirical}}{\text{Theoretical}}\right| \times 100$$

فاز ۳: اثبات تجربی قضایا و خواص احتمالی

برنامه باید به عنوان فاز نهایی، دو مفهوم زیر را بر روی داده‌های انبوه تولیدشده ارزیابی و اثبات کند:

۱. **بررسی خاصیت بی‌حافظگی (Memoryless Property):** برای توزیع هندسی و توزیع نمایی، با فیلتر کردن داده‌های تولید شده در کد و محاسبه احتمالات شرطی، صحت رابطه زیر را با گزارش میزان تفاوت دو طرف تایید کنید:

$$P(X > s + t \mid X > s) \approx P(X > t)$$

۲. **سنجش قضایای تقریب حدی:**

- **تقریب دوجمله‌ای توسط توزیع نرمال:** دریافت یک توزیع دوجمله‌ای با n بزرگ ($n \geq 50$). محاسبه تجربی احتمال وقوع متغیر در یک بازه مشخص مانند $P(a \leq X \leq b)$ و مقایسه آن با احتمال متناظر تئوریک در توزیع نرمال با میانگین $\mu = np$ و واریانس $\sigma^2 = np(1-p)$. اعمال **تصحیح پیوستگی (Continuity Correction)** با تغییر بازه پیوسته به $[a - 0.5, b + 0.5]$ الزامی است.
- **تقریب پواسون توسط توزیع نرمال:** دریافت توزیع پواسون با λ بزرگ ($\lambda \geq 30$)، محاسبه احتمال یک بازه مشخص روی داده‌های شبیه‌سازی‌شده و مقایسه آن با تقریب توزیع نرمال متناظر ($\mu = \lambda$ و $\sigma^2 = \lambda$).

نیازمندی‌های رابط کاربری و نحوه اجرا

- برنامه باید به صورت یک اسکریپت مستقل پایتون طراحی شود که قابلیت اجرا در ترمینال (کنسول) را داشته باشد.
- برنامه باید در زمان اجرا به صورت تعاملی از طریق دستور `input()` پارامترهای لازم (نوع توزیع، پارامترهای ریاضی توزیع، تعداد شبیه‌سازی‌ها و بازه‌های مورد نظر برای قضایای تقریب) را از کاربر دریافت کند و بر اساس ورودی‌های کاربر محاسبات را انجام دهد.

قوانین تحویل پروژه

- کدهای نوشته شده باید به صورت کاملاً ساختاریافته و شیءگرا تحویل داده شوند.

- خروجی برنامه در ترمینال باید به صورت شکیل، تفکیک‌شده و شامل مقایسه دقیق آمارهای تجربی، تئوریک و درصدهای خطای متناظر باشد.
- تک تک اعضای تیم باید به بخش‌های مختلف کد تسلط کامل داشته باشند و فرآیند بارم‌بندی بر اساس جلسه‌ی ارائه پروژه و ارزیابی منطق کدها صورت خواهد گرفت.