6. تولید اعداد کتره‌ای[۲۱]

در تمام فصل‌های گذشته با پدیده‌هایی آشنا شدید که رفتار کاتوره‌ای داشتند. همچنین در شبیه سازی آنها نیز مولد اعداد کتره‌ای نقش اساسی را در کُد و شبیه‌سازی پدیده داشته‌اند. احتمالا شما برای انجام تمرین‌های خود از مولدی که نرم افزار مورد استفاده در اختیارتان قرار می‌داده استفاده کرده‌اید. ولی فراموش نکنید که کامپیوتر شما یک ماشین کاملا منطقی است و تصادف و احتمال در آن جایی ندارد. در حقیقت این جای خوشبختی است که این ماشین همیشه برای یک سری فرآیندهای منطقی پاسخی قاطع و تکرار پذیر دارد. ولی چگونه می‌توان با چنین ماشینی اعداد کتره‌ای ساخت؟ جواب ساده است: **نمی‌توان.** پس این مولدها چه چیزی به ما تحویل می‌دهند و ما چگونه بر خروجی برنامه‌های خود اعتماد کنیم وقتی می‌دانیم که این مولدها ما را گول زده‌اند و یک رشته اعداد تکرار پذیر که با یک دیگر رابطه منطقی دقیقی نیز دارند را بجای اعداد کتره‌ای جا زده اند؟ برای پاسخ این سوال در ابتدا تعریفی از یک رشته‌ی کاتوره‌ای خوب باید داشته باشیم.

6.3. تابع توزیع احتمال[۲۲]

یک مولد اعداد کتره‌ای، رشته‌ای از اعداد را به ما تحویل می‌دهد و ادعا می‌کند که این رشته مجموعه مناسبی است برای آنکه آنها را کتره‌ای بدانیم. ولی این رشته چه خصوصیتی باید داشته باشد. اولین خاصیتی که انتظار می‌رود تبعیت از تابع توزیع مورد نظر است. به طور مثال اگر شما تابعی در زبان برنامه نویسی خود دارید که ادعا می‌کند اعداد کتره‌ای با توزیع یکنواخت بین صفر و یک تولید می‌کند، ساده‌ترین انتظاری که می‌توان داشت این است که در رشته‌ای که این مولد به ما تحویل می‌دهد احتمال ظهور عددی بیش از دیگری نباشد.

6.1.	
تمرین	– در زبان برنامه نویسی که استفاده می‌کنید تابع مولد اعداد کتره‌ای که اعداد صحیح بین 0 تا 9 تولید می‌کند را صدا بزنید. این تابع را در حلقه‌ای به طول N قرار دهید و منحنی فرآوانی اعداد خروجی را رسم کنید. اگر این مولد سالم باشد انتظار دارید هر یک از اعداد $N/10$ بار ظاهر شود. آیا این گونه است؟ – نشان دهید که انحراف نسبی از عدد بالا با جذر عکس N به سمت صفر می‌رود. یعنی $$\frac{\sigma}{N} \sim \frac{1}{\sqrt{N}}$$ – آیا شباهتی میان این تمرین و تمرین ول نشست مشاهده می‌کنید؟

[21] Random Number Generator
[22] Probability Distribution Function

در ادامه خواهیم دید که می‌توان مولدهایی با تابع توزیع غیریکنواخت نیز داشت که امتیازی برای استفاده از مولد به حساب می‌آید. نکته مهم این است که مولد باید بخوبی تابع توزیعی که ادعا می‌کند از آن تبعیت می‌کند را تولید کند.

6.4. همبستگی[23]

وقتی یک تاس را به بالا پرت می‌کنید انتظار دارید که در صورت سلامت تاس، احتمال نشستن هر یک از اعداد 1 تا 6 برابر باشد. به عبارت دیگر تابع توزیع ظهور اعداد باید یکنواخت باشد. ولی این تنها انتظاری نیست که از یک تاس سالم می‌رود. نکته دیگری نیز وجود دارد که به همین اندازه مهم است. انتظار می‌رود که مقداری که تاس در هر پرتاب نشان می‌دهد از مقدارهایی که قبلا نشان داده است مستقل باشد[24]. یعنی همبستگی بین اعداد وجود نداشته باشد و تاریخچه تاثیری بر آینده نداشته باشد.

6.2. تمرین	تمرین قبل را تکرار کنید با این تفاوت که این دفعه از رشته اعداد تصادفی فقط اعدادی را بردارید که عدد قبلی آنها 4 بوده باشد. مثلا اگر اعداد تولید شده، 1 4 6 7 8 9 4 4 8 3 5 7 1 4 9 باشد، تابع توزیع اعداد قرمز را نمایش دهید.آیا تابع توزیع این اعداد یکنواخت است؟

6.5. تولید اعداد شبه کتره ای[25] با توزیع یک نواخت

دوباره به سوال اول این بخش بر می‌گردیم. چگونه یک ماشین کاملا منطقی مانند کامپیوتر می‌تواند اعداد کتره‌ای تولید کند؟ به هر حال نتیجه‌ی دو تمرین قبل ظاهرا در تایید کیفیت کتره‌ای بودن اعداد این ماشین‌هاست. در حقیقت رشته تولید شده توسط کامپیوتر اصلا کتره‌ای نیست بلکه یک رشته ی کاملا منطقی است ولی الگوریتم تولید این رشته توانایی تولید رشته‌ای را دارد که در ظاهر کتره‌ای به نظر بیاید. برای همین به اینها مولد اعداد **شبه کتره‌ای** می‌گویند.
آلگوریتم‌های تولید اعداد شبه کتره‌ای انگشت شمار هستند. یکی از معروف‌ترین ومتداول‌ترین این آلگوریتم‌ها روش استفاده از هم‌نهشتی است. این روش به دلیل عدم نیاز به حافظه زیاد و نیز به دلیل سرعت بالا در تولید اعداد، بسیار پر کاربرد است.

---

[23] Correlation

[24] بعضی مواقع در زندگی روزمره با بی توجهی به این اصل، انتظار نامعقولی داریم. مثلا گفته می‌شود که مدتی است که اصلا 6 نیامده، پس این دفعه حتما می‌آید. این یعنی انتظاری وجود دارد که اگر مثلا در 30 پرتاب قبل 6 ظاهر نشده در پرتاب 31ام احتمال ظهور 6 کمی بیشتر از بقیه اعداد باشد.

[25] Pseudorandom

در این روش رشته اعداد با یک مقدار اولیه که به بذر[۲۶] معروف است شروع می‌شود. با یک معادله بازگشتی، دیگر عددهای رشته قدم به قدم تولید می‌شود. این برنامه بازگشتی به شکل

$$x_{n+1} = (a\, x_n + c) \mod m \qquad (1)$$

است. در اینجا a ضریب، c جابجایی و m عامل هم‌نهشتی است و هرسه اعداد مثبتی هستند و به پارامترهای الگوریتم معروفند. بدیهی است که این رابطه ساده نمی‌تواند یک رشته‌ی کتره‌ای را تولید کند. اولین اشکالی که به این الگوریتم وارد است خاصیت تناوبی آن است. از آنجا که پارامترهای مدل ثابت هستند، یک رابطه منطقی میان x_n و x_{n+1} وجود دارد. در نتیجه اگر در رشته‌ی اعداد، عددی تکرار شود دنباله نیز تکرار خواهد شد.

از آنجا که x ها باقیمانده‌ی تقسیم بر m هستند، پس همواره $x_i < m$ پس در بهترین حالت دوره تناوب این رشته نی‌متواند از m بزرگتر باشد. البته بسته به انتخاب مجموعه‌ی پارامترها، این تناوب می‌تواند بسیار کوچک‌تر نیز باشد. بنابراین کیفیت رشته اعداد تولیدی به شدت به انتخاب پارامترهای مدل بستگی دارد. داشتن دوره تناوب تنها اشکال این الگوریتم نیست. بلکه همبستگی‌های دیگری نیز در رشته‌های تولیدی می‌تواند کیفیت ظاهر کاتوره‌ای رشته را خدشه دار کند. روش همنهشتی برای تولید اعداد کتره‌ای به خوبی مطالعه شده است و مجموعه پارامترهایی برای تولید رشته پیشنهاد می‌شوند که نقاط ضعف را بهتر بپوشانند[۲۷].

به طور خاص می‌توان نشان داد که اگر $m = 2^k$ باشد، در صورتی که c عددی اول باشد و $(a-1)$ مضربی از 4 باشد، دوره تناوب این رشته به مقدار بیشینه خود، یعنی m می‌رسد. بیشتر توابع کتره‌ساز در نرم افزارهای آشنا از چنین مجموعه پارامترهایی استفاده می‌کنند. در بیشتر این نرم افزار ها $k = 32$ است و پارامترهای دیگر به گونه ای انتخاب شده‌اند که در شرط بالا صدق کرده و دوره تناوب مقدار بیشنه خود را داشته باشد. به طور مثال در Borland C++، $\{a, c, m\} = \{\mathbf{22695477},\ \mathbf{1},\ \mathbf{2^{32}}\}$ و در MS Visual C++، $\{a, c, m\} = \{\mathbf{214013},\ \mathbf{2531011},\ \mathbf{2^{32}}\}$ است.

خروجی این مولدها عددهای صحیحی از 0 تا $2^{32} - 1$ هستند. این عددها کمتر به همین شکل مورد استفاده قرار می‌گیرند. به طور مثال در بیشتر مواقع ما به خروجی بین 0 تا 1 نیازمندیم. برای این منظور توابع واسطی در کُدها وجود دارند که خروجی را متناسب با نیاز ما در اختیار ما قرار دهند. برای این مثال در صورتی که خروجی صحیح بر $2^{32} - 1$ تقسیم شود، حاصل همان چیزی است که ما به دنبالش هستیم. ولی باید توجه کرد که بدلیل این که این اعداد کاملا کاتوره ای نیستند در کار کردن با آنها باید احتیاط کرد. مثال زیر موضوع را روشن تر خواهد کرد.

به منظور انجام تمرین‌های بالا به مولدی احتیاج دارید که اعداد صحیح بین 0 تا 9 را تولید کند. همچنین فرض کنید که خروجی خام شبه کتره‌ساز را در اختیار دارید که عددی بین 0 تا $2^{32} - 1$ است. یک راه ساده این است که از اعداد تولید شده یک رقم را انتخاب کنیم. مثلا عدد یکان را بر داریم. حال اگر تمرین (6.2) را با این مولد انجام دهید نتیجه جالبی خواهید گرفت. تمام عددهایی که بعد از 4 ظاهر می‌شوند فرد هستند. اصلا جای تعجب ندارد. کافی است به معادله بالا و مجموعه پارامترهای معرفی شده نگاه کنید تا به دلیل آن پی ببرید. m عددی زوج و a, c عددهای فرد هستند. پس خروجی این مولدها به تناوب فرد و زوج خواهند بود. ولی من بعید می‌دانم که شما چنین نتیجه ای از تمرین (6.2) گرفته باشید. زیرا برنامه نویسان به خوبی با این مشکل آشنا هستند و در الگوریتم خود عدد دیگری را برای گزارش به جای عدد یکان انتخاب کرده اند.

---

[26] seed

[27] برای آشنایی با مجموعه های مناسب پارامترها به مرجع معرفی شده در **"بیشتر بدانیم"** در انتهای این فصل مراجعه کنید.

مثال ساده‌ی فوق به خوبی نشان می‌دهد اگر چه مولدهای اعداد شبه کاتوره‌ای می‌توانند در بسیاری مواقع بسیار مفید باشند ولی استفاده‌ی درست از آنها نیازمند اطلاعات کاملی از رفتار آنها است. وجود همبستگی منطقی بین اعداد رشته و همچنین دوره تناوب محدود می‌تواند درنتایج بعضی از شبیه سازی‌ها، بخصوص شبیه سازی‌هایی که نیاز به صدا کردن مکرر مولد دارند تاثیر جدی بگذارد. تمرین (6.1) به ما نشان داد که تابع توزیع این اعداد یکنواخت است و در کل فضای یک بعدی احتمال ظهور اعداد برابر است. ولی اگر از مجموعه های d تایی از اعداد متوالی در رشته برای مشخص کردن نقاط کاتوره ای در فضای d بعدی استفاده شود، قضیه ی مارساگلیا[۲۸] نشان می‌دهد که به دلیل همبستگی اعداد، این نقاط بر روی تعداد محدودی صفحه ی $d-1$ بعدی می‌نشینند.

6.6. بذر و کاتوره‌گر[۲۹]

آلگوریتم همنهشتی یک مزیت خیلی مهم از نظر محاسباتی دارد و آن هزینه پایین محاسبات است. آخرین عدد تولید شده را می‌گیرد و عدد بعدی را تحویل می‌دهد. ولی این داستان باید سر آغازی داشته باشد. اولین عددی که برای شروع دنباله به آلگوریتم باید تحویل داد را به اصطلاح بذر می‌نامند. ساختار منطقی آلگوریتم الزام می‌کند که با مشخص شدن بذر کل رشته مشخص می‌شود. اگر بذر در یک برنامه تعیین شود و مقدار ثابتی داشته باشد، اجرای مجدد برنامه دقیقا تکرار اجرای قبل خواهد بود. ولی در صورت تغییر بذر رشته کاتوره‌ای و در نتیجه خروجی برنامه تغییر خواهد کرد. یکی از راه‌های افزایش عامل تصادف در دنباله‌ی تولیدی، استفاده از بذر کاتوره‌ای است. همیشه عواملی وجود دارد که نقش تصادف در آنها بسیار بالا است. هرچند به دلیل محدود بودن این گزینه‌ها امکان تولید رشته کاتوره‌ای با آنها وجود ندارد ولی می‌توان به مقدار محدود از آنها سود جست. یکی از این مناسبت‌ها انتخاب بذر است. در بیشتر برنامه‌ها تابعی به عنوان **کاتوره‌گر** وجود دارد که چنین نقشی دارد. این تابع از **زمان** به عنوان عنصر تصادف استفاده می‌کند. اگر زمان اکنون (ساعت + دقیقه + ثانیه) را به صورت یک عدد صحیح نشان دهیم می‌توانیم ادعا کنیم که این عدد کاملا ساختار تصادفی دارد. لحظه‌ای که کاربر کلید اجرا را فشار می‌دهد به خیلی عوامل انسانی و محیطی بستگی دارد و در نتیجه وقتی برنامه به خطی می‌رسد که کاتوره‌گر را صدا می کند این عدد قابل پیش بینی نیست. استفاده از کاتوره‌گر نمی‌تواند کیفیت دنباله شبه تصادفی را بالا ببرد، ولی این امکان را به ما می‌دهد که قطعه‌ی مورد استفاده از دنباله را به طور تصادفی انتخاب کنیم.

نکته 1:

در شبیه‌سازی‌هایی که از آلگوریتم تصادفی استفاده می‌شود به طور معمول هرچه آمار بالاتر باشد، نتایج دقیق‌تر خواهد بود. برای همین گاهی بعد از اتمام اجرا در صورتی که شبیه ساز متوجه ضعف آماری نتایج شود به فکر تکرار برنامه برای بالا بردن آمار می‌افتد. اگر برنامه از کاتوره‌گر استفاده نکرده باشد و با بذر

[28] Marsaglia’s Theorem
[29] Randomizer

نکته 2:

یکی از مشکلات برنامه‌هایی که در آنها آلگوریتم‌های تصادفی وجود دارد این است که در صورت وجود مشکل یا باگ (bug) در برنامه، این مشکل نیز خود را به طور تصادفی نشان می‌دهد. یعنی در اجراهای متفاوت در زمان‌های متفاوتی ظاهر می‌شود. این باعث می‌شود که دیباگ یا عیب یابی کردن برنامه بسیار مشکل شود. برای همین توصیه می‌کنم که در مرحله دیباگ کردن برنامه، از بذر ثابت استفاده کرده و کاتوره‌گر را خاموش کنید. به این ترتیب با تکرار برنامه مشکل در زمان ثابتی آشکار می‌شود. این به شما این امکان را می‌دهد که به داخل برنامه رفته و در قدم ما قبل از خطا به وارسی برنامه و متغیرها بپردازید تا عامل خطا را بیابید. این کار را می‌توانید با چند بذر متفاوت نیز تکرار کنید. بعد از اطمینان از صحت برنامه و برای اجرای اصلی و ثبت نتایج فراموش نکنید که کاتوره‌گر را روشن کنید.

نکته 3:

بعضی مواقع شبیه سازان ترجیح می‌دهند که برای کاهش همبستگی دنباله‌ی اعداد، کاتوره‌گر را به دفعات در جای جای برنامه صدا کنند. مطمئنا این کار در کاهش همبستگی دنباله موثر است ولی باید توجه کرد که کثرت این کار می‌تواند نتایج مخربی داشته باشد. همانطور که گفته شد، کاتوره‌گرها معمولا از زمان با دقت ثانیه به عنوان بذر استفاده می‌کنند. اگر فاصله دو بار صدا زدن کاتوره‌گر کمتر از ثانیه باشد، این کار نه تنها کمکی به کاهش همبستگی نمی‌کند، بلکه تاثیر کاملا معکوس دارد و منجر به تکرار مجدد رشته می‌شود. برای امتحان می‌توانید کاتوره‌گر و دستور چاپ یک عدد تصادفی را در درون یک حلقه گذاشته وبرنامه را اجرا کنید.

مستقل از عدم کتره‌ای بودن واقعی اعداد مشکل مهم دیگر تناوب محدود رشته است. این نکته باعث می‌شود که امکان اجرای برنامه‌های طولانی از شبیه‌ساز گرفته شود. برای مثال اگر قصد بدست آوردن متوسط یک کمیت آماری را داشته باشید، به خوبی می‌دانیم که با افزایش آمار دقت اندازه‌گیری (محاسبه) افزایش میابد. ولی این جمله تا جایی درست است که نمونه برداری‌های جدید از قبلی‌ها مستقل باشد. مطمئنا دوبرابر کردن نمونه‌ها با تکرار آنها و بدون ورود آمار جدید هیچ ارزشی ندارد.
اگر در تمرین ول نشست در بخش (3.2) طول سیستم برابر با m، دوره تناوب مولد اعداد شبه کاتوره‌ای انتخاب شود، در هر m قدم تمام خانه های شبکه یک ذره دریافت می‌کنند (البته نه به ترتیب). بنابر این ناهمواری این ول نشست صفر خواهد شد. در این گونه مواقع باید راهی برای افزایش دوره تناوب پیدا کرد. در زیر یک روش ساده برای این کار معرفی می‌شود.

آلگوریتم بُر زدن دنباله‌ی اعداد کتره‌ای

1. دو تابع مولدR1 و R2 که از مجموعه پارامترهای متفاوتی استفاده می‌کنند ایجاد کنید (یکی از این‌ها می‌تواند همان مولد استاندارد نرم افزار شما باشد)
2. آرایه ای به طول مثلا 200 بسازید و آنرا با صدا کردن R1 مقدار دهی کنید.
3. با استفاده از R2 عددی کاتوره‌ای $1 \leq p \leq 200$ را تولید کنید.
4. عنصر pام آرایه را به عنوان خروجی مولد جدید خود گزارش کنید و دوباره با صدا کردن R1 این عنصر را مقدار جدیدی بدهید.

تمام خروجی‌های الگوریتم ترکیبی بالا بوسیله‌ی R1 تولید شده‌اند و در نتیجه این آلگوریتم تابع توزیع یکنواخت R1 را مخدوش نمی‌کند. تنها کاری که این آلگوریتم می‌کند استفاده از R2 برای بُر زدن دنباله تولید شده توسط R1 است. این کار باعث می شود که دوره تناوب رشته خروجی بسیار بزرگتر شود. اگر دوره‌های تناوب R1 و R2 نسبت به هم اول باشند، دوره تناوب خروجی آلگوریتم بالا برابر با حاصل ضرب دوره‌های تناوب دو مولد می‌باشد. این کار حتی در شبیه‌سازی‌های کوتاه‌تر که نگرانی برای مشاهده تکرار دنباله وجود ندارد نیز کار بسیار خوبی است، زیرا باعث می‌شود که همبستگی اعداد متوالی در دنباله مخدوش شود.

6.7. تولید اعداد کتره ای با توزیع غیر یکنواخت

در بسیاری از مواقع ما تمایل داریم که مولد ما اعداد کاتوره‌ای با توزیع دلخواه ما تولید کند. مثلا با بسیاری از پدیده‌های تصادفی در فیزیک آشنا هستیم که عامل تصادف توزیع طبیعی (گوسی) دارد. در چنین مواقعی شبیه‌سازی این پدیده‌ها نیازمند مولدی است که رشته اعداد کاتوره‌ای آن از توزیع گوسی تبعیت کند. در ابتدا با مثالی نشان می‌دهیم که می‌توان با استفاده از مولد کاتوره‌ای با توزیع یکنواخت، مولدی دیگر را آماده کرد.

6.7.1. قضیه حد مرکزی[۳۰]

این قضیه یکی از قضایای مهم در آمار است و کاربردهای فرآوانی در مسایل مختلف دارد. به زبان ساده این قضیه می‌گوید هر کمیت تصادفی که در حقیقت مجموع تعداد زیادی کمیت تصادفی مستقل باشد، از یک تابع توزیع احتمال گوسی پیروی می

[30] Central Limit Theorem (CLT)

کند[۳۱]. شکل ساده شده‌ی این قضیه می‌تواند به منظور تولید اعداد با توزیع گوسی استفاده شود. فرض کنید که مولدی با توزیع دلخواه $P(x)$ برای تولید دنباله اعداد کتره‌ای $\{x_1, x_2, \ldots, x_i, \ldots\}$ با مقدار میانگین

$$< x > = \int x\, P(x)\, \mathrm{d}x, \tag{2}$$

و مقدار مجذور افت و خیز (انحراف از معیار)

$$\sigma_x^2 = \int x^2\, P(x)\, \mathrm{d}x - < x >^2, \tag{3}$$

در اختیار داریم. متغیر y را اینگونه تعریف می‌کنیم:

$$y = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{4}$$

بنا به قضیه حد مرکزی y متغیر کاتوره‌ای است که تابع توزیع آن برای مقادر $N \gg 1$ به تابع توزیع طبیعی

$$P(y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\, e^{-\frac{(y-y_0)^2}{2\sigma_y^2}} \tag{5}$$

میل می‌کند که در آن

$$< y > = y_0 = < x > \tag{6}$$

و

$$\sigma_y = \frac{\sigma_x}{\sqrt{N}} \tag{7}$$

است.

.6.3 تمرین	برای تحقیق قضیه‌ی حد مرکزی با استفاده ازمولد اعداد کاتوره‌ای در نرم افزار مورد استفاده‌ی خود، تابع توزیع اعدادی که از جمع N تولید می‌شوند را بدست آورید. این کار را برای مقادیر $N = \{5, 10, 100, 1000\}$ انجام دهید. چه شباهتی میان این تمرین و تمرین ول‌گشت و ول‌نشست می‌بینید؟

[31] این قضیه به شکل دقیق بعضی شرایط را برای کمیتهای کاتوره ای که در جمع وارد میشوند قایل است که در اینجا از ذکر این جزییات میگذریم. خواننده ی علاقمند میتواند به مراجع انتهای این بخش مراجعه کند.

6.7.2. تغییر تابع توزیع با استفاده از تابع تبدیل

فرض کنید که مولدی با تابع توزیع $p(x)$ در اختیار داریم. در نتیجه احتمال اینکه این مولد عددی در بازه $(x, x+\mathrm{d}x)$ را تحویل دهد، $p(x)\mathrm{d}x$ است. موضوع این بخش این است که به دنبال تابعی هستیم که با اعمال آن بر روی x متغیری مانند y تولید کند ($y = f(x)$) که از تابع توزیع دلخواه $g(y)$ تبعیت کند. در این صورت احتمال داشتن عددی در بازه‌ی $(y, y+\mathrm{d}y)$ باید برابر با $g(y)\mathrm{d}y$ باشد. در نتیجه داریم:

$$p(x)\mathrm{d}x = g(y)\mathrm{d}y. \tag{8}$$

بدون اینکه چیزی از کلیت بحث کاسته شود و فقط به دلیل اینکه در عمل برای تولید اعداد معمولا مولدِ یکنواختِ استانداردِ نرم افزارهای خود را در اختیار داریم، فرض می‌کنیم که

$$p(x) = p_u(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{else where} \end{cases} \tag{9}$$

از طرفین رابطه (8) انتگرال می‌گیریم

$$\int_{-\infty}^{x} p_u(x)\mathrm{d}x = \int_{-\infty}^{y=f(x)} g(y)\mathrm{d}y. \tag{10}$$

سمت چپ این تابع به راحتی با استفاده از شکل تابع $p_u(x)$ قابل محاسبه و برابر با x است. فرض می‌کنیم انتگرال سمت راست را قادر به محاسبه هستیم و حاصل تابع تجمعی $G(y)$ است. درنتیجه داریم

$$x = G(y). \tag{11}$$

اگر این تابع معکوس پذیر باشد می‌توانید به راحتی y را برحسب x بنویسیم،

$$y = f(x) = G^{-1}(x). \tag{12}$$

چند مثال می‌تواند به روشن شدن توانایی این روش در تولید اعداد کاتوره‌ای با توزیع دلخواه کمک کند. در ابتدا با مثال بسیار ساده‌ای شروع می‌کنیم. فرض کنید

$$g(y) = \begin{cases} A & a \leq y < b \\ 0 & \text{else} \end{cases} \tag{13}$$

مقدار $A = \frac{1}{b-a}$ با فرض یکه بودن تابع توزیع تثبیت می‌شود. با استفاده از رابطه (10) به سادگی داریم:

$$x = G(y) = \frac{y-a}{b-a}. \quad (14)$$

که به رابطه نه چندان غیر قابل پیش بینی

$$y = a + (b-a)x \quad (15)$$

می‌رسیم.

مثال کمی پیچیده‌تر می‌تواند تابع توزیع نمایی باشد. فرض کنید که مولد قرار است اعدادی مثبت و با تابع توزیع زیر را بدهد.

$$g(y) = \begin{cases} a\, e^{-\frac{y}{a}} & x \geq 0 \\ 0 & \text{else} \end{cases} \quad (16)$$

در این حالت نیز به سادگی با محاسبه انتگرال داریم:

$$x = G(y) = 1 - e^{-\frac{y}{a}} \quad (17)$$

و معکوس آن به رابطه

$$y = -a\ln(1-x) \quad (18)$$

منجر می‌شود. در نتیجه این رابطه‌ی ساده می‌تواند از مولد یکنواخت کامپیوتر ما مولد کاتوره‌ای بسازد که خروجی آن تابع توزیع نمایی داشته باشد. توجه به این نکته که $(1-x)$ همان تابع توزیع x را دارد می‌تواند کمک کند تا رابطه بالا باز هم ساده تر شود

$$y = -a\ln x \quad (19)$$

یکی از مورد توجه‌ترین توابع توزیع در فیزیک، تابع توزیع نرمال یا گوسی است. ولی در مورد این تابع کار به سادگی مثال‌های بالا نیست. پاسخ برای انتگرال محدود تابع گوسی وجود ندارد. در این حالت می‌توان از ترفند آشنای استفاده از فضای دو بعدی استفاده کرد. تابع توزیع گوسی

$$g(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{y^2}{2\sigma^2}} \quad (20)$$

را در نظر می‌گیریم. دو متغیر y_1 و y_2 را که هر دو تابع توزیع بالا را دارند را در نظر می‌گیریم . در این حالت احتمال داشتن جفت (y_1, y_2) از رابطه زیر بدست می‌آید.

$$g(y_1, y_2) = g(y_1)g(y_2) = \frac{1}{2\pi\sigma^2} e^{-\frac{y_1^2 + y_2^2}{2\sigma^2}} \tag{21}$$

اگر متغیرهای (y_1, y_2) مختصات دکارتی یک نقطه در نظر بگیریم با تغییر متغیر به مختصات قطبی داریم:

$$g(y_1, y_2)dy_1 dy_2 = \frac{1}{2\pi\sigma^2} e^{-\frac{\rho^2}{2\sigma^2}} \rho \, \mathrm{d}\rho \, \mathrm{d}\theta \tag{22}$$

از رابطه بالا به راحتی می‌توان تابع توزیع ρ و θ را خواند.

$$g_\rho(\rho) = \frac{1}{\sigma^2} e^{-\frac{\rho^2}{2\sigma^2}} \rho \tag{23}$$

$$g_\theta(\theta) = \frac{1}{2\pi} \tag{24}$$

فاکتور ρ در رابطه اول کمک می‌کند که تابع نمایی انتگرال پذیر شود. در مورد θ کار حتی ساده‌تر نیز است. پس به این روش به راحتی قادر هستیم جفت اعداد (ρ, θ) را با توابع توزیع بالا تولید کنیم. با تغییر متغیر از مختصات قطبی به دکارتی به ازای هر جفت مختصات قطبی یک جفت مختصات دکارتی به دست می‌آید. نکته جالب توجه این است که هر دو مولفه‌ی مختصات دکارتی از تابع توزیع گوسی تبعیت می‌کنند. به این ترتیب به ازای هر دو بار صدا کردن مولد یکنواخت یک جفت عدد کاتوره ای با توزیع گوسی داریم. یعنی یک عدد به ازای هر با ر صدا کردن.

6.4.	با روشی که دربالا توصیف شد مولدی با توزیع گوسی بسازید و با رسم فرآوانی خروجی‌های آن نشان دهید که مولدتان خوب کار می‌کند.
تمرین	

بیشتر بدانیم:

برای آشنایی با الگوریتم های تولید اعداد کاتوره ای و نیز مجموعه پارامترهای مناسب برای استفاده دراین آلگوریتم ها میتوانید به کتاب Number Theory for Computing نوشته ی Song Yan مراجعه کنید. برای اطلاع بیشتر از قضیه حد مرکزی نیز متوانید به کتاب Statistical Physics of Particle نوشته یی Meharn Kardar مراجعه کنید.