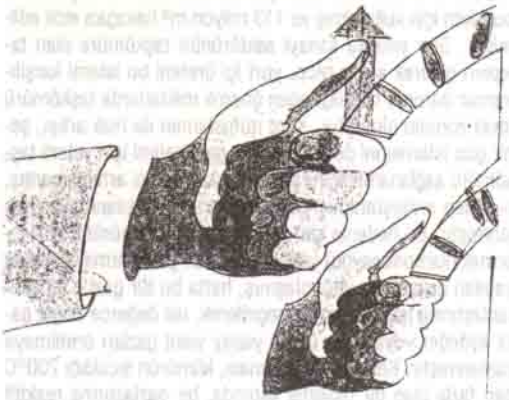
Emrehan HALICI

GELİŞİGÜZEL (RANDOM) SAYILAR



Bu sayımızda, bazı okuyucularımızdan gelen isteğe uyarak "Gelişigüzel Sayılar" konusunu inceleyeceğiz.

Gelişigüzel sayılar, bilgisayarlar için en önemli konulardan biridir. Çevremizde gelişen olayların birçoğu önceden kestirilemezken, aynı belirsizliği bilgisayarda gerçekleştirerek, gelişigüzel sayılar ve karakterler üretmek hiç de sanıldığı kadar kolay değildir. Gelişigüzel sayılar, istatistik hesaplamalarda birinci derecede kullanılmalarının yanısıra, şifreli kodlar üreterek, uydu sinyallerinin transferi, diplomatik ve askeri haberleşmelerde gizliliğin sağlanması ve bilgisayar kontrollü banka hesap transferlerine yetki dışı girişimlerin önlenilmesi gibi konularda da kullanılmaktadır. İstatistikçiler, verilerin doğruluğunu gelişigüzel sayılar kullanarak test ederler. Tıp araştırmacıları, denemelerinin doğruluğunu saptamada, elektronik mühendisleri tasarladıkları devrelerin her durumda istenilen şekilde çalışmasının denetiminde, ekonomistler ise pazar araştırmalarında ve piyasa tahminlerinde gelişigüzel sayılara dayanan modeller kullanırlar.

Gelişigüzellik ile ilgili olarak üniversitelerde yeni bilim dalları ortaya çıkıyor. Fizikçiler ve matematikçiler olayları gelişigüzel yapabilmenin yanşı içindeler. Kaliforniya Üniversitesi'nden Manuel Blum, bu konuda şunları söylüyor: "Gelişigüzellik başlıbaşına bir değer. Tıpkı bankadaki para gibi."

Gelişigüzellik deyince, önce konunun tanımını yapmak gerekiyor. Neyin gelişigüzel olup, neyin olmadığını birbirinden ayırmak gerçekten çok zor. Bir olayın gelişigüzel olması, çıkabilecek sonuçların hangi sırada gerçekleşeceğinin önceden bilinememesi demektir. Fakat olay belli sayıda tekrar edildiğinde, çıkacak sonuçların dağılımı önceden kestirilmekte, hatta böyle olması beklenmektedir. Eğer bu beklenti gerçekleşmezse, o olayın gelişigüzelliğinden şüphe edilmesi gerekmektedir. Örneğin yazı-tura atma olayını ele alalım. Eğer bu olay gelişigüzel ise "yazı" ve "tura"nın hangi sıralarla geleceğini önceden kestirmek mümkün değildir. Ama bu işlem 1000 kez tekra edilecekse, yaklaşık 500 kez yazı, 500 kez de tura geleceği beklenir. Bu sayıların çok ötesinde sayılar gerçekleşmişse, örneğin 300 yazı, 700 tura gelmişse, yazı-tura atma olayı gelişigüzel gerçekleşmemiş demektir. Stanford Üniversitesi istatistikçilerinden Persi Diaconis bu konuyla ilgili bir toplantıda, herhangi bir parayı alarak arka arkaya atmış ve devamlı tura elde etmiştir. Gelişigüzellik ve olasılık hesaplarını altüst eden bu olay şöyle açıklanmıştır: Diaconis aynı zamanda amatör bir sihirbazdır. Parmaklarını uzun yıllar bu şekilde eğiterek, paranın istediği yüzünün gelmesini sağlayabilmektedir. Dolayısıyla bir olayın gelişigüzel olması, çevre faktörleri ve dış etkenlerle yakından ilişkilidir. Hilesiz olan bir para atma olayı gelişigüzelliğe iyi bir örnektir; ama paranın dönüş hızı, ilk uygulanan kuvvet, yükselme noktası gibi parametreler tam olarak bilinirse, paranın hangi yüzünün geleceği de bilinebilir. Böylece, para atma olayında gelişigüzelliğin tamamen dış etkenlerden kaynaklandığı, bunların kesinlikle belirlenemeyeceği ve ölçülemeyeceği anlaşılmaktadır. Bu düşünceden esinlenilerek, gelişigüzel sayılar üretmede fiziksel olaylardan yararlanılmaya başlanmıştır. Bilgisayar bilimcisi David Gifford tasarladığı gelişigüzel sayı üreticisinde, bilgisayara gelen elektrik akımındaki küçük değişikliklerden yararlanmaktadır. Bu değişiklikleri önceden kestirmek mümkün olmadığı için, üretilen sayıların da gelişigüzel olacağını kabul etmektedir. "Gelişigüzel sayı üreticisi" derken, bilgisayardan bağımsız bir cihazdan değil de, bilgisayara verilmesi gereken bir dizi komut veya bu işlemi gerçekleştirecek algoritmadan bahsedilmektedir.

Bilgisayarlarda ilk gelişigüzel üretici, John von Neumann ve Stanislaw Ulam tarafından kullanılmıştır. O zamandan günümüze kadar Monte-Carlo metodları başlığı altında, birçok modellemede gelişigüzel sayılardan yararlanılmaktadır. Kişisel bilgisayarlarda en yaygın olarak kullanılan gelişigüzel sayı üretme metodu LCG (Linear Congruential Generator)'dır. 1948 yılında Derrick Lehmer tarafından geliştirilen metod oldukça basittir. Birçok diğer metod gibi LCG de ilk seçilen bir sayıyla başlar. Bu sayıya tohum (seed) denir. Tohum ikinci bir sayıyla çarpılır (çarpan), buna bir üçüncü sayı eklenir (artım) ve sonuç dördüncü bir sayıya bölünür (bölen). Elde edilen kalan, gelişigüzel serinin ilk elemanıdır. Metod ikinci elemanı bulmaya başlarken bu sayıyı tohum olarak kullanır ve işlem, üçüncüyü bulurken, ikincinin tohum olarak kullanılması şeklinde devam edip gider. LCG ve benzer metodların iki büyük dezavantajı vardır:

1. Belli bir süre sonra seri kendisini tekrar etmeye başlar,

2. Programa girilen ilk sayı (tohum) aynı olursa, üretilen seri de tamamen aynı olur.

Bu iki sebepten dolayı, bu şekilde üretilen sayılara sözde gelişigüzel (pseudo random) sayılar denmektedir.

Yukarıda bahsedilen dezavantajlar, şifreli kodlar üreten ve çözen kriptoloji biliminde bir avantaj olarak karşımıza çıkar. Modern bilgi kodlama işlemlerinde şifrelenmiş mesajların tamamen gelişigüzel bir görünüme sahip olması istenir. Daha önceleri kullanılan basit kodlama sistemlerinde, her harf belli sayıda ileriye ya da geriye kaydırılarak yeni harfler elde edilir ve mesajlar bu sistemle yazılırdı. Bu şekilde şifrelenmiş mesajlar ne kadar anlamsız ve karışık görünürse görünsün, her harfe sabit bir işaret karşılık geldiğinden, bilgisayarlar yardımıyla hassas analizler sonucu (dilin özelliklerin-

den de yararlanarak) kolayca çözülebilmektedir. Örnek olarak, her harfi iki ileriye kaydırarak yeni bir alfabe elde edelim.

Normal : A, B, C, Ç, D, E, F G, Ğ, H, I, İ, J, K, L, M, N, O, Ö, P, R, S, Ş, T, U, Ü, V, Y, Z,

Kodlu : C, Ç, D, E, F, G, Ğ, H, I, İ, J, K,L, M, N, O, Ö, P, R, S, Ş, T, U, Ü, V, Y, Z, A, B.

"BİLİM VE TEKNİK" bu kodlama sistemiyle "ÇKNKO ZG ÜGMÖKM" şeklinde yazılabilir.

Oysa mesajların kodlanmasında gelişigüzel sayılardan yararlanmak, çok daha güvenilir sonuçlar elde edilmesini sağlamaktadır. Kodlanacak her karaktere hangi karakterin karşılık geleceği, gelişigüzel sayılardan oluşmuş bir seriye göre bulunmaktadır. Alıcı, mesajı aldığında aynı gelişigüzel seriyi kullanarak, ters işlem yapacak ve şifreyi çözecektir. Aynı gelişigüzel serinin kullanılması için, hem yollayıcı hem de alıcıda aynı gelişigüzel sayı üreticisinin bulunması gerekir. Tek yapılacak iş, seriyi başlatan aynı tohumun ilk sayı olarak, hem yollayıcı hem de alıcı tarafından kullanılmasıdır. Aynı tohumun aynı seriyi üreteceği dezavantajı kriptolojistlerin böyle bir metodu kullanmasını mümkün kılmaktadır. Örnek olarak "RAMDOM SAYILAR" mesajının kodlanacağını varsayalım. Standart olarak bu harflerin ASCII kod karşılıkları aşağıda verilmiştir.

Mesaj	:	**R**	**A**	**N**	**D**	**O**	**M**		**S**	**A**	**Y**	**I**	**L**	**A**	**R**
ASCII kod	:	**82**	**65**	**78**	**68**	**79**	**77**	**32**	**83**	**65**	**89**	**73**	**76**	**65**	**82**

(Boşluğun ASCII karşılığı 32'dir.)

Bu kodların ne kadar arttırılıp ne kadar eksiltileceğini belirlemek için kullanılacak gelişigüzel seri ise şu olsun

Seri	:	**6**	**4**	**7**	**-8**	**6**	**-5**	**2**	**4**	**9**	**1**	**7**	**10**	**-2**

Bu iki sayı serisi toplandığında şifrelenmiş mesaj kodları ortaya çıkar:

Şifre kod	:	:	**82**	**71**	**82**	**75**	**71**	**83**	**27**	**85**	**69**	**98**	**74**	**83**	**75**	**80**

Alıcı, bu sayılardan gelişigüzel serideki sayıları çıkararak orijinal mesajı elde eder.

Aşağıda, aynı tip bilgisayara (dolayısıyla aynı gelişigüzel sayı üreticiye) sahip olan yollayıcı ve alıcıda bu işi gerçekleştirecek program örnekleri verilmiştir:

```
 10 REM 15 KARAKTERE KADAR UZUNLUKTA***
 20 REM MESAJI ŞİFRELEYEN PROGRAM******
 30 DIM A (15), B (15)
 40 RANDOMIZE
 50 REM TOHUM SORULDUĞUNDA 1987 GİRİNİZ
 60 INPUT "MESAJI GİRİNİZ"; MS
 70 FOR I=1 TO LEN (M$)
 80 A(I)=ASC (MIDS (M$, I,1))
 90 B(I)= A(I)+INT (RND*21)-10
100 NEXT I
110 FOR I=1 TO LEN (MS)
120 PRINT B (I);
130 NEXT I
```

```
 10 REM 15 KARAKTERE KADAR UZUNLUKTA***
 20 REM MESAJI ÇÖZEN PROGRAM**********
 30 DIM A(15), B(15)
 40 RANDOMIZE
 50 REM TOHUM SORULDUĞUNDA 1987 GİRİNİZ
 60 INPUT "ŞİFRELİ MESAJ UZUNLUĞU"; L
 70 FOR I=1 TO L
 80 INPUT "KODLARI SIRASIYLA GİRİNİZ"; A(I)
 90 B(I)=A(I)-INT (RND*21)+10
100 NEXT I
110 FOR I=1 TO L
120 PRINT CHRS (B (I));
130 NEXTI
```

Her iki programda kullanılırken gelişigüzel sayı üretmek için tohum (random number seed) sorulduğunda aynı sayı girilmelidir. Yukarıda örnek olarak 1987 kullanılmıştır.

Gelişigüzel üretildiği sanılan sayıların bazılarının aslında gelişigüzel olmayıp, belli bir düzene uyduğunu, bilgisayarcı George Marsalgia aşağıdaki örnekle açıklamıştır:

a) Üretilen sayılar üç boyutlu bir grafiğe (bir kübün tanımladığı) işlenmiştir. Herşey gelişigüzel gözüküyor.

b) Bilgisayar kübü döndürdüğünde belli paralel çizgiler beliriyor.

c) Küp daha da döndürülünce, noktaların tam bir uyum içinde olduğu ortaya çıkıyor.