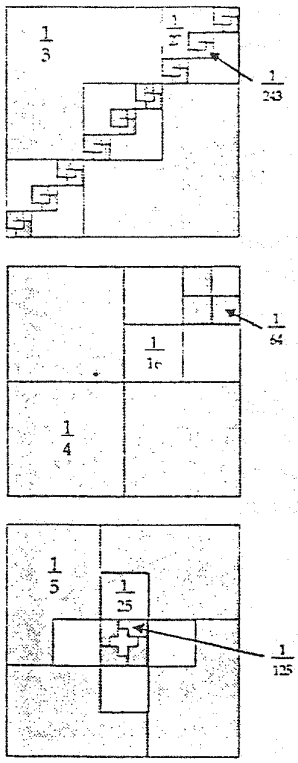

$$2\left(\frac{1}{3}+3\cdot\frac{1}{27}+9\cdot\frac{1}{243}+\cdots\right)=1$$

$$2\sum_{n=1}^{\infty}\frac{1}{3^n}=1$$

$$\sum_{n=1}^{\infty}\frac{1}{3^n}=\frac{1}{2}$$

$$3\sum_{n=1}^{\infty}\frac{1}{4^n}=1$$

$$\sum_{n=1}^{\infty}\frac{1}{4^n}=\frac{1}{3}$$

$$4\sum_{n=1}^{\infty}\frac{1}{5^n}=1$$

$$\sum_{n=1}^{\infty}\frac{1}{5^n}=\frac{1}{4}$$

KAYNAKÇA

[1] A. G. Aksoy, *Fibonacci Sayıları*, *Matematik Dünyası*, **5**, sayı 4, 13–16 (1995).

[2] Ş. Alpay, *Modern Matematiğin Sabahı*, *Bilim ve Ütopya*, Ağustos (1995).

[3] R. B. Nelsen, *Proofs Without Words: Exercises in Visual Thinking*, The Mathematical Association of America, Washington, 1993.

[4] Ö. Özlük, A. Şahin, C. Tezer, *Pisagor Teoreminin Çeşitli İspatları*, *Matematik Dünyası*, **1**, sayı 3, 6–9 (1991).

FAKTÖRİYEL TABAN İLE HATASIZ İŞLEMLER

Oya Tabağ *

İşte küçük ama ilginç bir deney. Alalım hesap makinemizi elimize ve 1/3 kesirini "1 ÷ 3" yazarak makineye girelim. Alışılmış 8 basamaklı (ilk sıfır dahil) ondalık sayımız görünecek: 0.3333333. Bu sayının 1/3 kesirinin tam değeri olmadığını biliyoruz, fakat yine de kabul edelim ki yeterince yakın. Şimdi ekrandaki sayıyı 4 ile çarpalım ve sonuçtan 1 çıkaralım.

$4(1/3) - 1 = 1/3$ olduğundan yanıtın yine 0.3333333 olduğuna şaşırmıyoruz. Bu işlemlere devam edelim. Yani ekrandaki görüntüyü değiştirmeden tekrar tekrar 4 ile çarpalım ve 1 çıkaralım. Hesap makinemle ben bu deneyi yaptığımda yedinci tekrardan sonra ekrandan aldığım değer 0.3333333 değildi. 24. tekrarda ise ekranda görünen değer -9382.1659 idi.

Evet, yanlışlık nerede? 1/3 kesirinin dijital olarak 0.3333333 şeklindeki kullanımında $3.\bar{3} \times 10^{-7}$'den çok daha küçük bir değer gözardı edilmişti. Ama işlem sayısı büyüdükçe bu değer de büyüyüp yanıtlarda çok büyük yanlışıklara yol açtı. Böyle bir olay daha önceden hesaplanmış değerlerin başka veya tekrar yapılan işlemlerde de kullanıldığı bilgisayar programlarında gerçekten can sıkıcı sonuçlara yol açabilir. Bu yazı, 1/3 gibi rasyonel sayıların dijital olarak tam karşılığının kullanılabilmesi için bir yol anlatıyor. Böylece tekrarlı işlemlerde ortaya çıkan hata payı ortadan kalkıyor. Bu yol ise *faktöriyel taban*a geçiştir.

Faktöriyel Taban

İlk kez 19. yüzyıl Alman matematikçilerinden Georg Cantor'un bulduğu faktöriyel tabanda

* Özel İzmir Amerikan Lisesi 2. sınıf öğrencisi

bütün rasyonel sayıların sonlu bir karşılığını bulmak mümkün [2]. Bunun nasıl olduğunun anlatılabilmesi için öncelikle faktöriyel tabanın ne olduğunun açıklanması gerekir.

Bu tabanın basamak değerleri olarak onluk tabandaki gibi 10'un ardışık üslerinin kullanımı yerine ardışık faktöriyeller kullanılır.

$$\cdots \quad \overline{(a+2)!} \quad \overline{(a+1)!} \quad \overset{N}{\overline{a!}} \quad \cdots \quad \overline{3!} \quad \overline{2!} \quad \overline{1!} \quad \bullet \quad \overline{[2!]^{-1}} \quad \overline{[3!]^{-1}} \quad \cdots \quad \overline{[a!]^{-1}} \quad \overline{[(a+1)!]^{-1}} \quad \overline{[(a+2)!]^{-1}} \quad \cdots$$

Örneğin, yukarıda görüldüğü gibi N değişkenini bir faktöriyel taban sayısının $a!$ basamağındaki değer olarak kabul edelim. N'nin buradaki gösterimi bu basamağın onluk tabanda $N \cdot a!$ değerini içerdiğini ifade eder.

Sayısal ve karşılaştırmalı bir örnek ise şu şekildedir: Onluk tabanda

$$321.1_{10} = (3 \cdot 10^2) + (2 \cdot 10^1) + (1 \cdot 10^0) + (1 \cdot 10^{-1})$$

iken faktöriyel tabanda

$$\begin{aligned} 321.1_F &= (3 \cdot 3!) + (2 \cdot 2!) + (1 \cdot 1!) + (1 \cdot (2!)^{-1}) \\ &= 23.5_{10} \end{aligned}$$

olur.

Bu sayıların ifadesinde en önemli nokta şudur: Bir faktöriyel taban tamsayısında, $a!$ basamağında yer alabilecek en büyük sayı a'dır. Örneğin 5000_F gibi bir ifadeyi faktöriyel tabanda yazmıyoruz. Bunun sebebi ise bu sayının onluk tabandaki değerinin $5 \cdot 4!$, yani $5!$ olmasıdır. Aynı değerin doğru gösterimi 10000_F şeklindedir.

Eğer ele aldığımız basamak, faktöriyel noktasının sağ tarafında ise, yani basamak değeri $1/a!$ ise, burada yer alabilecek en büyük sayı $a - 1$'dir. Örneğin 0.004_F gibi bir ifadeyi yazmıyoruz, çünkü $4(1/4!) = 1/3!$'dir. Bu sayının doğru gösterimi ise $0.01_F = 0.1\overline{6}_{10}$ şeklindedir.

Bir de basamak ayırımında bir zorluk vardır ki bu sadece $9!$'den büyük ya da $1/9!$'den küçük basamakları gerektiren sayılar için geçerlidir. Sebebi ise bu basamaklara gelebilecek sayıların 2 veya daha çok rakamlı olma olasılığıdır. Ben de bu sayıları diğer basamaklardan ayırabilmek için aşağıdaki örnekte görüldüğü gibi kesme işaretini kullanmayı uygun gördüm:

$$4'10'000000000_F = 4 \cdot 11! + 10 \cdot 10! = 43545600_{10}.$$

Burada gördüğümüz gibi onluk tabanda basamaklara gelebilecek rakam sayısı 10 tane ile sınırlı iken, faktöriyel tabanda sınırsızdır.

Faktöriyel tabandan onluk tabana çevirim yöntemi bu tabanın tarifinden anlaşılıyor. Şimdi onluk tabandan faktöriyel tabana çevirim yollarına bakalım. Bu, tamsayılar ve kesirler için farklıdır.

Bir onluk taban tamsayısını faktöriyel tabana çevirmek için, bu tamsayı 2'den başlayarak ardışık sayılarla, kalan bölenden küçük olana kadar bölünür. Elde edilen son bölüm ve her bölümde artan sayılar ters sırada yazılarak faktöriyel taban tamsayısı bulunur.

Örnek. $27_{10} = (?)_F$

$$\begin{array}{rrrr} 27 & \lfloor\, 2 & & \\ -\,26 & 13 & \lfloor\, 3 & \\ \boxed{1} & -\,12 & 4 & \lfloor\, 4 \\ & \boxed{1} & -\,4 & \boxed{1} \\ & & \boxed{0} & \end{array}$$

Dolayısıyla $27_{10} = 1011_F$. Sağlamasını yapalım: $(1 \cdot 4!) + (0 \cdot 3!) + (1 \cdot 2!) + (1 \cdot 1!) = 27_{10}$.

1'den küçük bir pozitif kesiri çevirmek için ise, bu kesir sırayla 2'den başlayarak ardışık sayılarla, çarpım kesirsiz tamsayı olana kadar, çarpılır. Bu işlem yapılırken her çarpımın tamsayı kısmı çarpımdan çıkarılır ve kesirler sadeleştirilir. Elde edilen tamsayılar noktadan sonra yazılarak faktöriyel taban sayısı elde edilir.

Örnek 1. $(1/5)_{10} = 0.2_{10} = (?)_F$

	tamsayı kısmı
$(1/5)\ 2 = 2/5$	0
$(2/5)3 = 6/5 = 1 + 1/5$	1
$(1/5)\ 4 = 4/5$	0
$(4/5)\ 5 = 4$	4

veya

	tamsayı kısmı
$(0.2)\ 2 = 2/5$	0
$(0.4)3 = 6/5 = 1 + 1/5$	1
$(0.2)\ 4 = 4/5$	0
$(0.8)\ 5 = 4$	4

Dolayısıyla $1/5_{10} = 0.2_{10} = 0.0104_F$. Sağlayalım: $(0/2) + (1/3!) + (0/4!) + (4/5!) = 1/6 + 4/120 = (1/5)_{10}$.

Örnek 2. $(1/6)_{10} = (?)_F$

	tamsayı kısmı
$(1/6)\ 2 = 2/6 = 1/3$	0
$(1/3)3 = 1$	1

Dolayısıyla $1/6_{10} = 0.01_F$. Bunu da sağlayalım: $(0/2!) + (1/3!) = (1/6)_{10}$.

Onluk tabandaki kesir sırayla ardışık sayılarla çarpıldığı için, bir noktada kesirin paydasının bütün bölenleri sadeleştirilmiş olacaktır ve böylece çevirme işlemi sona erecektir. İşte bu sebeple faktöriyel tabanda bütün rasyonel sayıların, noktadan sonra sonlu sayıda basamakla ifade edilebilen birer tam dijital karşılığı vardır.

Aritmetik İşlemleri

Şimdi sıra aritmetik işlemlerine yani toplama, çıkarma, çarpma ve bölmeye geldi. Benim geliştirdiğim toplama ve çıkarma işlemleri ilkokul öğrencilerine öğretilen onluk taban toplama ve çıkarmasına benziyor. Buna karşılık çarpma ve bölme işlemlerinin, onluk tabandaki karşılıkları ile kıyaslandığında, çok daha karmaşık ve sıkıcı oldukları görülüyor.

Toplamada bir basamağa konması gereken sayı, buraya gelebilecek en büyük değerden de büyük ise bu sayının fazlası sol taraftaki basamaklara nakledilir. Bu da bu sayının sol taraftaki basamak değeri ile bölünmesi ile olur. Bu sol taraftaki basamak noktanın solundaysa geçerlidir; eğer sol taraftaki basamak noktanın sağındaysa, o zaman sayımız sağ taraftaki basamak değerine bölünür.

Örneğin 2! basamağında toplama sonucu elde edilen bir 5 sayısını 3! basamağına geçirmek için 3'e bölmek gerekir. Bölüm 1, 3! basamağına geçirilir; kalan 2 ise 2! basamağına yazılır. Benzer şekilde 1/3! basamağında toplama sonucu elde edilen bir 4 sayısını 1/2! basamağına geçirmek için gene 3'e bölmek gerekir. Bölüm 1, 1/2! basamağına geçirilir; kalan 1 ise 1/3! basamağına yazılır.

Örnek.

$$\begin{array}{r} 21.02_F \\ +\ 20.12_F \\ \hline 120.01_F \end{array}$$

Sağlama: $(5+2/6))_{10} + (4+5/6)_{10} = (10+1/6)_{10}$.

Çıkarmada ise sol basamaklardan ödünç alınan 1 değerleri alınan basamağın değeri ile çaprılarak sağ basamağa nakledilir. Örneğin 2! basamağından alınan bir 1 sayısı, 1! basamağına 2 olarak geçer.

Örnek.

$$\begin{array}{r} 120.11_F \\ -\ 11.02_F \\ \hline 101.02_F \end{array}$$

Sağlama: $(10+4/6)_{10} - (3+2/6)_{10} = (7+2/6)_{10}$.

1 sayısının toplama ve çıkarmada kullanılan transfer dizisi, yani soldan sağa aktarılırken çarpılması gereken sayılar dizisi, aşağıda görülüdüğü gibidir:

$$\ldots \underset{4!}{\underline{\ }} \quad \underset{3!}{\underline{\ }} \quad \underset{2!}{\underline{\ }} \quad \underset{1!}{\underline{\ }} \quad \underset{[2!]^{-1}}{\overset{\cdot}{\underline{\ }}} \quad \underset{[3!]^{-1}}{\underline{\ }} \quad \underset{[4!]^{-1}}{\underline{\ }} \quad \underset{[5!]^{-1}}{\underline{\ }} \ldots$$
$$\begin{array}{ccccccccccc} & 4 & & 3 & & 2 & & 2 & & 3 & \\ & \curvearrowright & & \curvearrowright & & \curvearrowright & & \curvearrowright & & \curvearrowright & \end{array} \qquad \begin{array}{ccc} & 4 & \\ & \curvearrowright & \end{array} \qquad \begin{array}{ccc} & 5 & \\ & \curvearrowright & \end{array}$$

Çarpma ve bölmede ise oluşan kısmi çarpım ve bölümlerin faktöriyel tabanda kullanılabilmesi için iki yol vardır. Bunlar bölerek ve çarparak dağıtım algoritmalarıdır. Eğer bir basamağa gelmesi gereken sayı gereğinden büyük ise bu sayı bölünerek sol taraftaki basamaklara faktöriyel taban tanımına uygun olacak şekilde dağıtılır. Eğer bir basamağa gelmesi gereken sayı bir kesirse bu sayı çarpılarak tamsayı olacak şekilde sol taraftaki basamaklara dağıtılır. Faktöriyel tabanda basamak değerleri sınırsızdır ve bu nedenle çarpım tablosu sonsuzdur. Bu yüzden çarpma işleminde çarpan sayının rakamlarının her biri tek başına onluk tabana çevirilir.

Örnek.

$$\begin{array}{r} 301.12_F \\ \times\ 21.1_F \\ \hline 0.01_F \\ 0.012_F \\ 0.1_F \\ 0._F \\ 111._F \\ \hline 301.12_F \\ \hline 1.02_F \\ 10._F \\ 20._F \\ 0._F \\ +\ 3000._F \\ \hline 4201.002_F \end{array}$$

$(19 + 5/6)_{10} \times (5 + 1/2)_{10} = (96 + 2/24)_{10}$ de işlemin sağlamasıdır.

Çarpan sayının en sağından başlayarak bütün basamaklarındaki değerler onluk tabana çevriliyor. Çarpılan sayının rakamlarının her biri bu onluk taban değerleri ile çarpılıyor ve dağıtım algoritmalarından uyan bir tanesiyle faktöriyel tabanda toplamaya uygun halde olması için faktöriyel tabana çevriliyor.

Bölme işleminde ise bölenin bir bütün olarak onluk tabandaki değeri kullanılır. Bölünecek sayının soldan sağa doğru bütün basamakları tek tek bölünüp dağıtım algoritmalarından uyan biriyle bu tabanın tanımına uygun şekle getirilir. Daha sonra bütün değerler alt alta yazılıp toplanır.

Örnek.

$$\begin{array}{rl} 1.02_F & \big| \quad 1.01_F \\ \hline & 0.120251_F \\ + & 0.012415_F \\ \hline & 1.003206_F \end{array}$$

Sağlama: $(1+2/6)_{10} \div (1+1/6)_{10} = (1+1/7)_{10}$.

Bilgisayar Hesapları

Artık aritmetik algoritmalar geliştirildiği için bunları bilgisayar programlarına çevirebiliriz. Böylece faktöriyel tabanın avantajlarını gösterebiliriz. Benim yazdığım bilgisayar programları QuickBasic dilinde.

Makalenin başında tarif ettiğim deney, Peter Wayner'ın 1988'de *Byte* dergisine yazdığı bir makaleden alınmıştı [2]. İşte faktöriyel tabanın avantajlarının gösterilmesinde yine bu deneyden yararlanıyoruz.

$f(x) = x(n + 1) - 1$ fonksiyonunda x değişkeni için $1/n$ değerini girdiğimizde sonucun hep $1/n$ olması gerekiyor. Yani fonksiyona verilen ve alınan değerler aynı olduğu için, bu işlemi sonuçların tekrar veri olarak kullanıldığı bir "döngü" fonksiyonuna dönüştürebiliriz. Makalenin başında n değerini 3 olarak kullandık ve işlem döngüsü çoğaldıkça aldığımız sonuçlar da 1/3'ten uzaklaştı. Fakat faktöriyel tabanı kullanarak aynı fonksiyonu yine n için 3 sayısını kullanarak çalıştırdığımızda bilgisayardan aldığımız sonuç tablosu aşağıda görülüyor.

İşlem sayısı	x	Faktöriyel taban ile	Normal işlem
1	1/3	0.3333333	0.3333333730697632
5	1/3	0.3333333	0.333343505859375
10	1/3	0.3333333	0.34375
15	1/3	0.3333333	11
20	1/3	0.3333333	10923
25	1/3	0.3333333	11184811
30	1/3	0.3333333	11453246123
35	1/3	0.3333333	11728124029611
40	1/3	0.3333333	$1.2009599006321132 \times 10^{16}$
45	1/3	0.3333333	$1.229782938247304 \times 10^{19}$
50	1/3	0.3333333	$1.259297728765239 \times 10^{22}$

Kısaca yukarıda tarifi görülen faktöriyel tabanın, sonuçta istenen netliğin hızdan daha önemli olduğu bilgisayar programlarında kullanılması düşünülebilir.

Tabii ki hem bilgisayarları hem de hesap makinelerini rasyonel sayıların p/q (p ve q tamsayı) şeklindeki kullanımı için programlayabiliriz, fakat dijital formda oldukları için faktöriyel taban kullanıldığında rasyonel sayıların büyüklüğünün kıyaslanması daha kolaydır. Örnek vermek gerekirse, $(17/35)_{10}$ ve $(53/108)_{10}$ arasında yapılması istenen bir karşılaştırma, aynı değerlerin faktöriyel tabandaki görüntülerinde daha kolaydır: 0.023315_F ve 0.02335226_F.

Bundan sonra ise bu tabanla ilgili programların hızlandırılması ve böylece kullanımı daha kolay bir taban haline gelmesi konusunda çalışmalar yapılabilir.

Program isteme adresi: Mustafa Bey cad., 34/6, Alsancak, İzmir 35220.

KAYNAKÇA

[1] R. L. Francis, *Star Numbers and Constellations, Mathematics Teacher*, **86**, 88–89 (1993).

[2] P. Wayner, *Error-Free Fractions, Byte*, (1988).