CENG 114 BİLGİSAYAR BİLİMLERİ İÇİN AYRIK YAPILAR

Doç. Dr. Tufan TURACI
tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü

- Hafta 5

Ders İçereği

- **İkilik, Onluk, Onaltılık tabanda sayıların yazılımı**
- **Fonksiyonlar**
- **Rekürsif Fonksiyonlar ve Rekürsif Algoritmalar**

İkilik Tabandan Onluk Tabana Sayıların Çevrilmesi

$$(abcde)_2 = a.2^4 + b.2^3 + c.2^2 + d.2^1 + e.2^0 \ (a,b,c,d,e \in \{0,1\})$$

$$(10011)_2 = 1.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 19$$

Sözde Kodu veya Program Nasıl yazılabilir?

C kodu:

```c
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{ int s = 0, bd = 0, kat = 0, top = 0;
  printf("2 lik tabanda sayiyi giriniz: ");
  scanf("%d", &s);
    while(s > 0)
    {
        bd = s % 10;
        if (bd>=2) { printf("Sayi ikilik tabanda degildir... \n");
                     goto son;}
        s = s/10;
        top = top + bd * pow(2,kat);
        kat++;
    }
   printf("Sayinin 10 luk tabandaki karsiligi : %d\n",top);
   son:
   getch ();
   return 0;
}
```

```
2 lik tabanda sayiyi giriniz: 10011
Sayinin 10 luk tabandaki karsiligi : 19

--------------------------------
```

```
2 lik tabanda sayiyi giriniz: 00111
Sayinin 10 luk tabandaki karsiligi : 7

--------------------------------
```

```
2 lik tabanda sayiyi giriniz: 12001
Sayi ikilik tabanda degildir...

--------------------------------
```

```
2 lik tabanda sayiyi giriniz: 100532
Sayi ikilik tabanda degildir...

--------------------------------
```

İstenilen Tabandan Onluk Tabana Sayıların Çevrilmesi

$$(abcde)_t = a.t^4 + b.t^3 + c.t^2 + d.t^1 + e.t^0 (a,b,c,d,e \in \{0,1,2,\ldots,t-1\})$$

$$(12003)_5 = 1.5^4 + 2.5^3 + 0.5^2 + 0.5^1 + 3.5^0 = 878$$

$$(121)_3 = 1.3^2 + 2.3^1 + 1.3^0 = 16$$

Sözde Kodu veya Program Nasıl yazılabilir?

C kodu:

```c
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{ int s = 0, bd = 0, kat = 0, top = 0, t;
    printf("Sayinin hangi tabanda oldugunu giriniz: ");
    scanf("%d", &t);
    printf("%d tabanindaki sayiyi giriniz: ", t);
    scanf("%d", &s);
    while(s > 0)
    {
        bd = s % 10;
        if (bd>=t) { printf("Sayi %d tabanindan degildir... \n",t);
                     goto son;}
        s = s/10;
        top = top + bd * pow(t,kat);
        kat++;
    }
   printf("Sayinin 10 luk tabandaki karsiligi : %d\n",top);
   son:
   getch ();
   return 0;
}
```

```
Sayinin hangi tabanda oldugunu giriniz: 5
5 tabanindaki sayiyi giriniz: 12003
Sayinin 10 luk tabandaki karsiligi : 878

--------------------------------
```

```
Sayinin hangi tabanda oldugunu giriniz: 3
3 tabanindaki sayiyi giriniz: 121
Sayinin 10 luk tabandaki karsiligi : 16

--------------------------------
```

```
Sayinin hangi tabanda oldugunu giriniz: 5
5 tabanindaki sayiyi giriniz: 12053
Sayi 5 tabanindan degildir...

--------------------------------
```

```
Sayinin hangi tabanda oldugunu giriniz: 3
3 tabanindaki sayiyi giriniz: 1242
Sayi 3 tabanindan degildir...

--------------------------------
```

Onluk Tabandan İstenilen Bir Tabana Sayıların Çevrilmesi

$$133 = (\ ?\ )_5$$

Çözüm: Verilen sayı sürekli 5' e bölünür. (Bölüm 5 den küçük olana kadar devam eder.)

$$133 = 26.5 + 3$$
$$26 = 5.5 + 1$$
$$5 = 1.5 + 0$$

$$133 = (1013)_5$$

Çalışma Sorusu: Sözde Kodu veya Programı Nasıl yazılabilir?

Onaltılık Tabanda Sayıların Yazılımı

$$A \Rightarrow 10 \qquad D \Rightarrow 13$$
$$B \Rightarrow 11 \qquad E \Rightarrow 14$$
$$C \Rightarrow 12 \qquad F \Rightarrow 15$$

Örnek:

$$\begin{aligned}(1AB3)_{16} &= 1.16^3 + A.16^2 + B.16^1 + 3.16^0 \\ &= 4096 + 10.256 + 11.16 + 3 \\ &= 4096 + 2560 + 176 + 3 \\ &= 6835\end{aligned}$$

Örnek:

$$6835 = (\ ?\ )_{16}$$

$$6835 = 427.16 + 3$$

$$427 = 26.16 + 11$$

$$26 = 1.16 + 10$$

$$\Rightarrow (1AB3)_{16}$$

Aritmetik İşlemler

Örnek: $(124)_5 + (562)_7 = (\ ?\ )_3$

Örnek: $(1572)_8 - (662)_8 = ?$

Örnek: $(5A3B6)_{16} + (F25E4)_{16} + (1CB25)_{16} = ?$

Fonksiyonlar

- *A* ve *B* boştan farklı kümeler olsunlar. *A*'dan *B*'ye bir *f fonksiyonu* A'nın her elemanının B'nin tekil bir elemanına atanmasıdır.
- Eğer *B*'nin *tekil* bir *b* elemanı *f* fonksiyonu ile *A*'nın bir *a* elemanına atanıyor ise $f(a)=b$ şeklinde belirtiriz.
- Eğer *f* , *A* dan *B* ye bir fonksiyon ise,

$$f : A \rightarrow B$$

- Fonksiyonlar birçok farklı şekilde belirtilebilir. Bazen atamaları resim olarak belirtilebilir.
- Bazen de, bir fonksiyonu tanımlamak için bir formül kullanır.
- $f(x) = 3x + 5$,
- Ya da bir fonksiyonu belirtmek için bir bilgisayar programı kullanılabilir.

- Bir $f : A \rightarrow B$ fonksiyonu A dan B ye bir bağıntı olarak da tanımlanabilir.
- A dan B ye bir bağıntı $A \times B$ *nin bir alt kümesidir.*
- **A** nın bir **a** elemanı için bir ve sadece bir (a, b) ***sıralı çiftini*** içeren A'dan B'ye bir bağıntı, A'dan B'ye ***f*** fonksiyonu tanımlar.
- Bu fonksiyon $f(a) = b$ ataması ile tanımlanır. Burada *(a, b)*, ilk öğesi a olan, bağıntıda yer alan benzersiz sıralı çifttir.

Tanım – Değer Kümeleri

- Eğer *f* , *A* dan *B* ye bir fonksiyon ise *A ya f ’in* ***Tanım Kümesi*** ve *B* ye de ***Değer Kümesi*** denir.
- Eğer *f* (*a*) = *b* ise, *b ye a nın* ***görüntüsü*** ve *a* ya da *b* nin ***öngörüntüsü*** denir.
- *Bir f fonksiyonunun* ***Görüntü Kümesi*** *A* nın tüm elemanlarının görüntülerinin kümesidir.
- *f* nin görüntü kümesi değer kümesi ile aynı küme veya değer kümesinin bir altkümesi olabilir.
- Eğer *f* , *A* dan *B* ye bir fonksiyon ise *f* , *A* yı *B* ye ***eşler*** deriz.

Örnek:

- Programlama dillerinde Fonksiyonların tanım ve değer kümeleri sıklıkla kullanılır.
- Örneğin, Java deyimi

  int **floor**(float real){. . .}

- Pascal deyimi

  function *floor*(*x*: **real**): **integer**

- her ikisi de floor fonksiyonunun tanım kümesinin reel sayılar kümesi ve değer kümesinin tamsayılar kümesi olduğunu belirtir.

Bire-bir Fonksiyonlar

- Bazı fonksiyonlar iki farklı tanım kümesi elemanına asla aynı değeri atamazlar. Bu türden fonksiyonlara **bire-bir** fonksiyon denir.
- Bir f fonksiyonu bire-bir dir, (injective), eğer ve ancak eğer $f(a) = f(b)$, f in tanım kümesindeki tüm a ve b ler için a = b yi gerektiriyor ise.

- Bir *f* fonksiyonu bire-bir dir eğer ve ancak eğer

$$a \neq b \text{ iken } f(a) \neq f(b) \text{ ise.}$$

- ***f*** in bire-bir olduğunu bu şekilde ifade etmek bire-bir tanımındaki gerektirmenin contrapositive 'ini alarak sağlanır.
- *f* 'in bire-bir olduğunu niceleyiciler ile de ifade edebiliriz:

$$\forall a \forall b (f(a) \neq f(b) \rightarrow a \neq b)$$

- veya denk olarak

$$\forall a \forall b (a \neq b \rightarrow f(a) \neq f(b))$$

Artanlık & Azalanlık

- Tanım ve değer kümesi reel sayılar kümesinin bir alt kümesi olan bir *f* fonksiyonu, *x* ve *y* , *f* in tanım kümesine ait ve x<y olmak üzere
  - ***artandır*** *eğer* $f(x) \leq f(y)$ *, ve*
  - ***kesin artandır*** *eğer* $f(x) < f(y)$

- Benzer şekilde, *x* ve *y* , *f* in tanım kümesine ait ve x<y olmak üzere
  - ***azalandır*** *eğer* $f(x) \geq f(y)$ *, ve*
  - ***kesin azalandır*** *eğer* $f(x) > f(y)$

Artanlık & Azalanlık

- Bir *f* fonksiyonu
- **artandır** eğer
$$\forall x \forall y (x < y \rightarrow f(x) \leq f(y)),$$
- **kesin artandır** eğer
$$\forall x \forall y (x < y \rightarrow f(x) < f(y)),$$
- **azalandır** eğer
$$\forall x \forall y (x < y \rightarrow f(x) \geq f(y)),$$ ve
- **kesin azalandır** eğer
$$\forall x \forall y (x < y \rightarrow f(x) > f(y))$$

Artanlık & Azalanlık

- Bu tanımlardan şunu görüyoruz ki :
- kesin artan ya da kesin azalan bir fonksiyon bire-bir fonksiyondur.
- Bununla birlikte, artan, ancak kesin artmayan veya azalan, ancak kesin azalmayan bir fonksiyon bire-bir değildir.

Örten (Onto) Fonksiyonlar

- *A* dan *B* ye tanımlı bir *f* fonksiyonu *örten dir (üzerine, surjective)* ,
- *eğer ve ancak eğer*
- her $b \in B$ elemanı için
  öyle bir $a \in A$ vardır ki $f(a) = b$

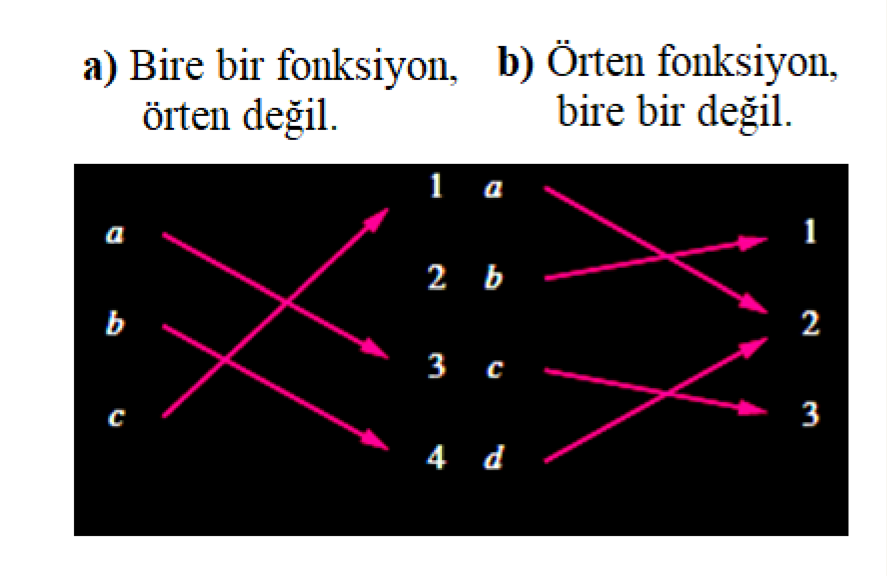
Örten (Onto) Fonksiyonlar

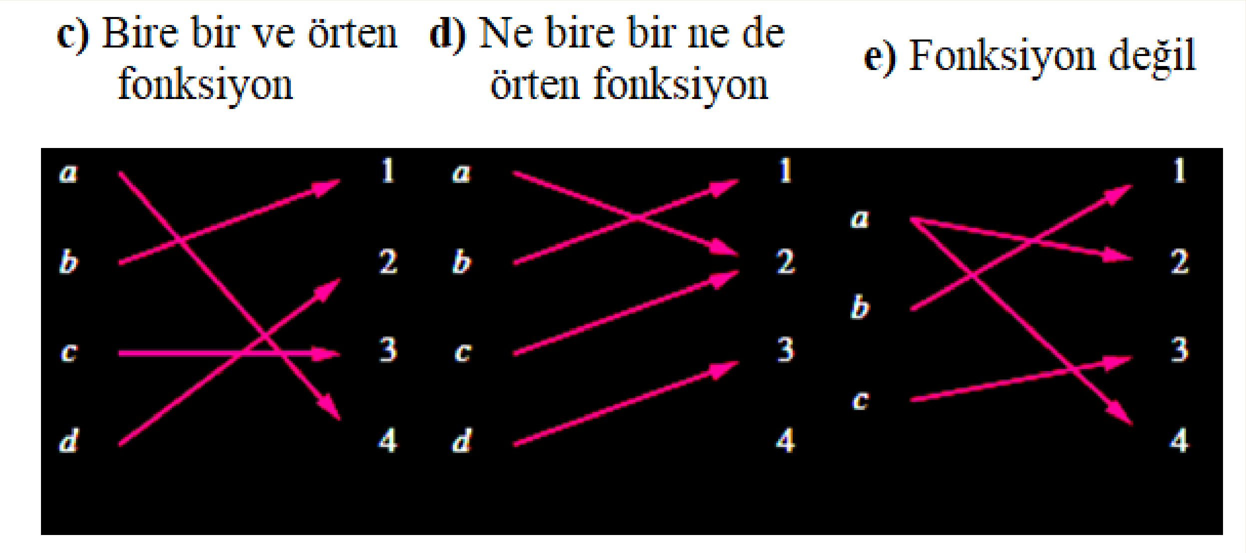
- Bir *f* fonksiyonu örtendir, eğer $\forall y \exists x (f(x) = y)$ ise
- Burada *x*'in tanım kümesi fonksiyonun tanım kümesi ve
- *y*'nin tanım kümesi ise fonksiyonun değer kümesidir.

Bire-bir örten (Bijection) fonksiyonlar

- *f* fonksiyonu bir *bire-bir denklik* dir, veya *bijection*, eğer hem bire-bir hem de örten ise.

Örnekler:





Ters Fonksiyon

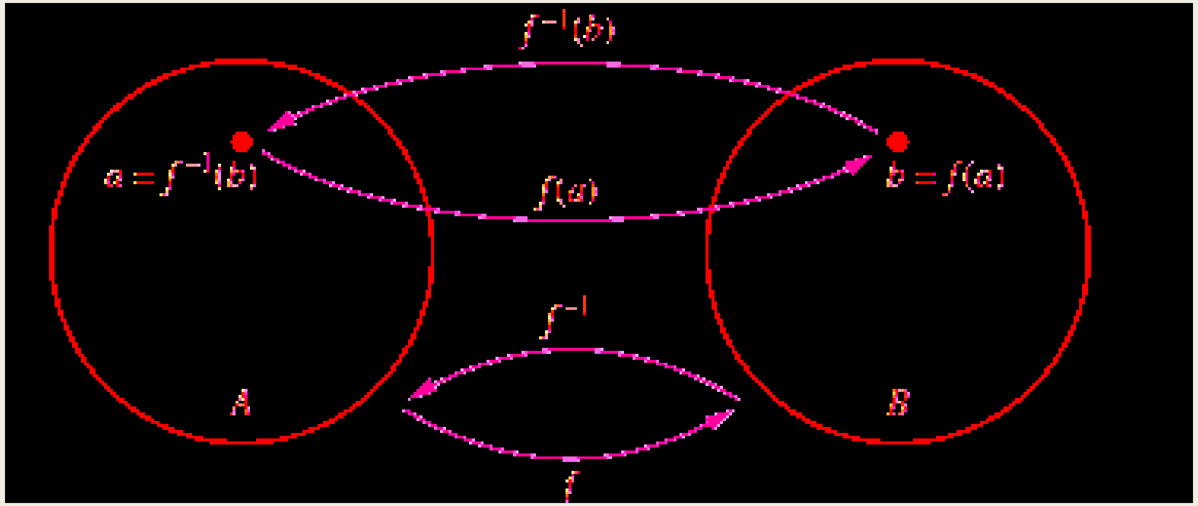
- f, A kümesinden B kümesine bire-bir denklik olsun.
  f in ters fonksiyonu B nin bir b elemanını $f(a) = b$ olacak şekilde A nın benzersiz bir a elemanına atayan bir fonksiyondur.
  f in ters fonksiyonunu f^{-1} ile belirtiriz.
  Dolayısıyla $f(a) = b$ olduğunda $f^{-1}(b) = a$ olur.

- Not: f^{-1} fonksiyonu $1/f$, ile karıştırılmamalıdır.
  f^{-1} ters fonksiyon.
  Diğeri ise tanım kümesindeki her bir x i $1/f(x)$ 'e atar.
  Hatta ikincisi $f(x)$ sadece sıfırdan farklı olduğunda anlamlıdır.

Ters Fonksiyon

- Eğer bir *f* fonksiyonu bire-bir denklik değil ise *f* fonksiyonunun tersini *tanımlayamayız*.

- *f* fonksiyonu bire-bir denklik olmadığında hem bire-bir hem de örten değildir.



Örnek

- f, $\mathbb{R}$ den $\mathbb{R}$ ye $f(x) = x^2$ şeklinde tanımlı bir fonksiyon olsun. f tersinir midir?

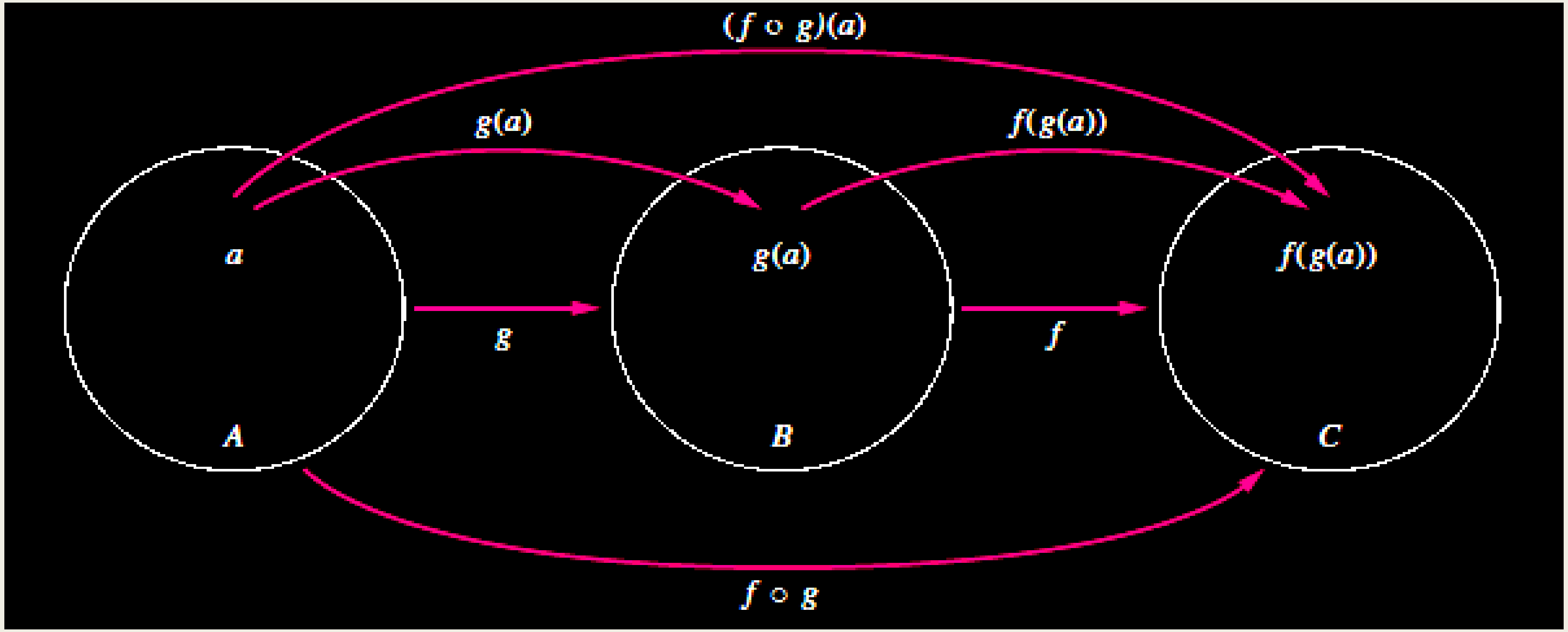
Çözüm: $f(-2) = f(2) = 4$, olduğundan f bire-bir değil. Eğer bir ters fonksiyon tanımlansaydı, 4 'e iki eleman ataması olurdu. Dolayısıyla, f tersinir değildir.

- Tersinir bir fonksiyon elde etmek için bazen fonksiyonun tanım kümesini veya değer kümesi, bazen de her ikisini birden sınırlayabiliriz.

- Yukarıdaki örnekteki $f(x) = x^2$ fonksiyonunu, negatif olmayan gerçel sayılar kümesinden negatif olmayan gerçel sayılar kümesine tanımlı bir fonksiyona kısıtlarsak, f tersinir olur.

Bileşke Fonksiyon

- g, A kümesinden B kümesine ve
  f, B kümesinden C kümesine tanımlı fonksiyonlar olsun
  $f \circ g$ ile temsil edilen f and g fonksiyonlarının bileşkesi,
  şöyle tanımlanır: $(f \circ g)(a) = f(g(a))$



Bazı Kurallar

$$\left(f^{-1} \circ f\right)(a) = f^{-1}(f(a)) = f^{-1}(b) = a$$
$$\left(f \circ f^{-1}\right)(b) = f\left(f^{-1}(b)\right) = f(a) = b$$

- *f* , A dan B ye bire-bir denklik ise f ‘in tersi de B den A ya bire-bir denkliktir.

Bazı Fonksiyonlar

Floor fonksiyonu , x gerçel sayısına x değerinden küçük veya x değerine eşit olan en büyük tamsayıyı atar. Floor fonksiyonunun x deki değeri $\lfloor x \rfloor$ ile gösterilir. ***Ceiling fonksiyonu*** , x gerçel sayısına x'den büyük veya x'e eşit en küçük tamsayıyı atar. Ceiling fonksiyonunun x deki değeri $\lceil x \rceil$ ile gösterilir.

Örnek:

- Bir bilgisayar diskinde depolanan veya bir veri ağı üzerinden iletilen veriler genellikle bir byte dizisi olarak temsil edilir. Her byte 8 bitten oluşur. 100 bit veriyi kodlamak için kaç byte gerekir?

- *Çözüm:* Gerekli byte sayısını belirlemek için, 100, bir byte taki bit sayısı olan 8'e bölündüğünde en az bölüm kadar olan en küçük tamsayıyı belirleriz. Sonuç olarak,

$$\lceil 100/8 \rceil = \lceil 12.5 \rceil = 13 \text{ byte gerekir.}$$

Floor & Ceiling Fonksiyonları

TABLE 1 Useful Properties of the Floor and Ceiling Functions.
(n is an integer)

(1a)	$\lfloor x \rfloor = n$ if and only if $n \leq x < n + 1$
(1b)	$\lceil x \rceil = n$ if and only if $n - 1 < x \leq n$
(1c)	$\lfloor x \rfloor = n$ if and only if $x - 1 < n \leq x$
(1d)	$\lceil x \rceil = n$ if and only if $x \leq n < x + 1$
(2)	$x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
(3a)	$\lfloor -x \rfloor = -\lceil x \rceil$
(3b)	$\lceil -x \rceil = -\lfloor x \rfloor$
(4a)	$\lfloor x + n \rfloor = \lfloor x \rfloor + n$
(4b)	$\lceil x + n \rceil = \lceil x \rceil + n$

Rekürsif Fonksiyonlar ve Rekürsif Algoritmalar

Rekürsif Fonksiyonlar ve Algoritmalar

Bir değişkenin terimlerinde doğrudan bir fonk. tanımlamak genellikle tercih edilebilir. Bu yöntemi kullanmak tekrarlamalık (rekürsiflik) olarak adlandırılır.

Tümevarım ilkesinden biliyoruz ki;

(i) Bir fonk. verilen bir a başlangıç değeri (genellikle a=0 veya a=1) için tanımlanmış ve;

(ii) a'dan büyük bir k değeri için tanımlandığında (k+1) içinde tanımlanıyorsa bu fonk, a'dan büyük bütün tamsayılar için tanımlanmış olur. Bu şekilde tanımlanmış bir fonk.na rekürsif fonk. denir.

Ön m

$f(1) = 1$

$f(k) = 2 \cdot f(k-1)$ fonksiyona rekürsif fonk. diyoruz. (Kapalı formda yazım diyoruz.)

$k=2$ için $f(2) = 2 \cdot f(1) = 2 \cdot 1 = 2 = 2^1$

$k=3$ için $f(3) = 2 \cdot f(2) = 2 \cdot 2 = 4 = 2^2$

$k=4$ için $f(4) = 2 \cdot f(3) = 2 \cdot 4 = 8 = 2^3$

$\vdots$

$k=n$ için $f(n) = 2 \cdot f(n-1) = 2^{n-1}$

$f(n) = 2^{n-1} \Rightarrow$ Rekürsif fonk.'nun açık formda yazımı denir.

(*) Açık formdaki ifadenin kapalı formdaki ifadeyi sağlaması gereklidir.

$n=1$ için $f(1) = 2^{n-1} = 2^{1-1} = 2^0 = 1$ $\qquad f(1)=1$

$n=k$ için $f(k) \overset{?}{=} 2 \cdot \underline{f(k-1)}$ doğru old. kabul edilir

$$2^{k-1} \overset{?}{=} \underbrace{2 \cdot 2^{k-2}}_{} = 2^{k-1}$$

Bunlar aynı fonk.lar olur.

(*) Uyarı: Rekürsif fonksiyonlar pozitif tamsayılar için tanımlanmış fonk. için kullanılır.

(*) Uyarı: Rekürsif fonk.lar kapalı ve açık formda ifade edilebilir.

Örn

$$f(0) = 2$$

$$f(k+1) = \frac{f(k)!}{(k+1)!}$$

ifadesi rekürsif bir fonk. mudur?

$k=0$ için $f(1) = \dfrac{f(0)!}{1!} = \dfrac{2!}{1!} = 2$

$k=1$ için $f(2) = \dfrac{f(1)!}{2!} = \dfrac{2!}{2!} = 1$

$k=2$ için $f(3) = \dfrac{f(2)!}{3!} = \dfrac{1!}{3!} = \dfrac{1}{6}$

$k=3$ için $f(4) = \dfrac{f(3)!}{4!} = \dfrac{\left(\frac{1}{6}\right)!}{4!} \Rightarrow ?$

Örneğimiz rekürsif fonk. değildir.

Rekürsif Algoritmeler

$f(n) = a^n$ fonk.nu rekürsif olarak yazınız.

1.yol (iteratif)

```
print ("a ve n değerlerini giriniz");
 read (a,n);
  carp=1;
for (i=1; i<=n; i++)
  carp=carp*a;
```

2. yol

```
function us(n)
begin
    If n=0 then us = 1;
                break;
    us(n) = a * us(n-1);
end;
```

$(\ldots)^n$

$3^5 = ?$

$us(5) = 3 * us(4)$

$3 * us(3)$

$3 * us(2)$

$3 * us(1)$

$3 * us(0)$

1

$= 243$

$f(n) = n!$ rekürsif olarak yazalım.

```
function fakt(n)
begin
    if n=0 then fakt = 1;
                break;
    fakt(n) = n * fakt(n-1);
end function
```

$4! = ?$

$fakt(4) = \boxed{4} * fakt(3)$

$3 * fakt(2)$

$2 * fakt(1)$

$1 * fakt(0)$

1

$= 24$

Ör) Fibonacci sayıları rekürsif olarak kapalı formda yazınız.

$1, 1, 2, 3, 5, 8, 13, 21, 34, \_ \ldots$

$$\boxed{\begin{aligned} f(0) &= 1 \\ f(1) &= 1 \\ f(n) &= f(n-1) + f(n-2) \end{aligned}} \Rightarrow$$

Fibonacci fonk. denir.

$$f(1) = 1$$
$$f(k) = f(k-1) + k$$

kapalı formda verilen rekürsif fonk. çözünüz.

$k=2$ için $\quad f(2) = f(1) + 2 = 1 + 2$

$k=3$ için $\quad f(3) = f(2) + 3 = 1 + 2 + 3$

$k=4$ için $\quad f(4) = f(3) + 4 = 1 + 2 + 3 + 4$

$\vdots$

$k=n$ için $\quad f(n) = f(n-1) + n = 1 + 2 + \dots + n-1 + n$

$$\boxed{f(n) = \frac{n.(n+1)}{2}}$$

açık forma.

Açık form, kapalı forma sağlamalıdır!!

$n=1$ için $f(1) = \frac{1.2}{2} = 1$ $\quad f(1) = 1$ ✓

⋮

$n=k$ için $f(k) \overset{?}{=} \underbrace{f(k-1)}_{\text{doğ. old. kabul edilir}} + k$

$\frac{k.(k+1)}{2}$

$= \frac{(k-1).(k-1+1)}{2} + k$

$= \frac{(k-1).k}{2} + k = \frac{k^2 - k + 2k}{2}$

$= \frac{k^2+k}{2} = \frac{k.(k+1)}{2}$

eşit

(c) $f(1) = 2$

$f(k) = 2k \cdot f(k-1)$ rekürsif fonksiyonu çözünüz.

$k=2$ için $f(2) = 2.2.f(1) = 2\cdot 2\cdot 2$

$k=3$ için $f(3) = 2\cdot 3\cdot f(2) = 2\cdot 3\cdot 2\cdot 2\cdot 2 = 3!\cdot 2^3$

$k=4$ için $f(4) = 2.4\cdot f(3) = 2\cdot 4\cdot 2\cdot 3\cdot 2\cdot 2\cdot 2 = 4!\cdot 2^4$

$k=5$ için $f(5) = 2.5\cdot f(4) = 5\cdot 2\cdot 4\cdot 2\cdot 3\cdot 2\cdot 2\cdot 2 = 5!\cdot 2^5$

$\vdots$

$k=n$ için $\boxed{f(n) = n!\cdot 2^n} \Rightarrow$ açık form?

Bu ifadenin kapalı formu sağlaması gerekir.

$n=1$ için $\quad f(1) = 1! \cdot 2^1 = 1 \cdot 2 = 2 \quad f(1)=2$

⋮

$n=k$ için $\quad f(k) = 2k \cdot f(k-1)$

$$k! \cdot 2^k \stackrel{?}{=} 2 \cdot k \cdot (k-1)! \cdot 2^{k-1}$$

$5 \cdot 4! = 5!$

$k!$

$2^1 \cdot 2^{k-1} = 2^k$

eşit

$= k! \cdot 2^k$

Örnek:

$$f(n) = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$$

fonk.nu rekürsif olarak kapalı formda yazınız ve doğruluğunu kanıtlayınız.

$n=1$ için $f(1) = \frac{1 \cdot 2 \cdot 3}{6} = 1$

$n=2$ için $f(2) = \frac{\cancel{2} \cdot \cancel{3} \cdot 5}{\cancel{6}} = 5 = 1+4$

$n=3$ için $f(3) = \frac{3 \cdot 4 \cdot 7}{6} = 14 = 5+9$

$n=4$ için $f(4) = \frac{4 \cdot 5 \cdot 9}{6} = 30 = 14+16$

⋮

$n=k$ için $f(k) = f(k-1) + k^2$

Böylece:

Kapalı form $\Rightarrow$ $\boxed{\begin{array}{l} f(1)=1 \quad ? \\ f(k)=f(k-1)+k^2 \end{array}}$ K

$n=1$ için $\quad f(1)=\dfrac{1.2.3}{6}=1 \qquad f(1)=1$ ✓

$\vdots$

$n=k$ için $\quad f(k)=f(k-1)+k^2$

$\Leftarrow$

$$\frac{k(k+1).(2k+1)}{6} \stackrel{?}{=} \frac{(k-1).k.(2k-1)}{6}+k^2$$

$$= \frac{(k^2-k).(2k-1)+6k^2}{6}$$

$$= \frac{2k^3-k^2-2k^2+k+6k^2}{6} = \frac{2k^3+3k^2+k}{6}$$

$2k \quad 1$

$k \quad 1$

$$= \frac{k.(2k^2+3k+1)}{6}$$

$$= \frac{k.(k+1).(2k+1)}{6}$$

c(n)

Catalan sayılarının bir dizisi

$$cat(n) = \frac{(2n)!}{(n+1)! \cdot n!}$$ şeklinde tanımlanır

ilk 10 terimi:

$\boxed{cat(0) = 1}$ $(*)$

$1, 1, \boxed{2}, 5, 14, 42, 132, 429, 1430, 4862, 16796$

cat(3) ...

cat(2)

cat(0) cat(1)

Bu sayı dizisini rekürsif olarak kapalı formülü:

$$\boxed{\begin{aligned} cat(0) &= 1 \\ cat(n+1) &= \frac{2\cdot(2n+1)}{n+2} \cdot cat(n) \end{aligned}}$$ $(*)$

şeklinde tanımlanır.

Açık formun, kapalı formu sağlaması gerekir.

$n=0$ için $cat(0) = \frac{(2\cdot 0)!}{(0+1)!\cdot 0!} = \frac{1}{1} = 1$ $\quad cat(0) = 1$ ✓

⋮

$n=k$ için $cat(k+1) \stackrel{?}{=} \frac{2\cdot(2k+1)}{k+2} \cdot cat(k)$

✓

$$\frac{(2k+2)!}{(k+2)!\cdot(k+1)!} \stackrel{?}{=} \underbrace{\frac{2\cdot(2k+1)}{k+2} \cdot \frac{(2k)!}{(k+1)!\cdot k!}}$$

paydaşı paydayı $k+1$ ile çarpalım.

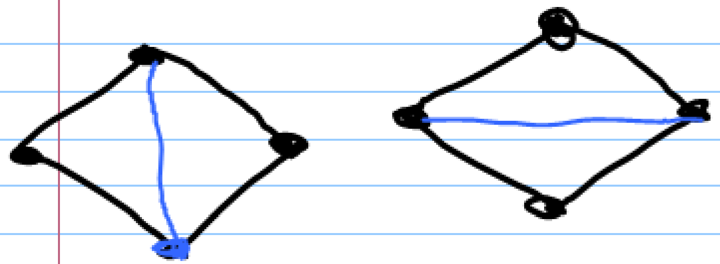
$$= \frac{\overbrace{2.(k+1)}^{(2k+2)}.(2k+1)(2k)!}{\boxed{(k+2)(k+1).k!}.(k+1)!}$$

$$= \frac{(2k+2)!}{(k+2)!.(k+1)!}$$

Çözüm tamamlandı.

Kullanım Alanı: $(n+2)$ kenarlı konveks bir çokgen kullandığın da $(n-1)$ doğru kullanarak, çokgeni üçgenlere ayırma sayısı $Cat(n)$ 'dir.

4 kenarlı çokgende $n+2=4$ $\boxed{n=2}$ $n-1 \Rightarrow 2-1=1$ doğru ile birbirini üçgenlere ayırmanın sayısı $Cat(2)=2$ 'dir.

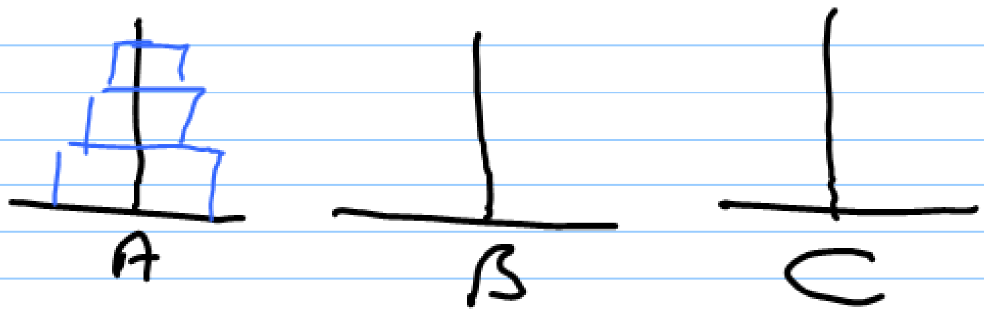


2 farklı şekilde üçgenlere ayrılır.

5gen/ $n+2=5$ $\boxed{n=3}$ $n-1 \Rightarrow 3-1=2$ doğru ile $Cat(3)=5$ farklı şekilde üçgenlere ayrılır.

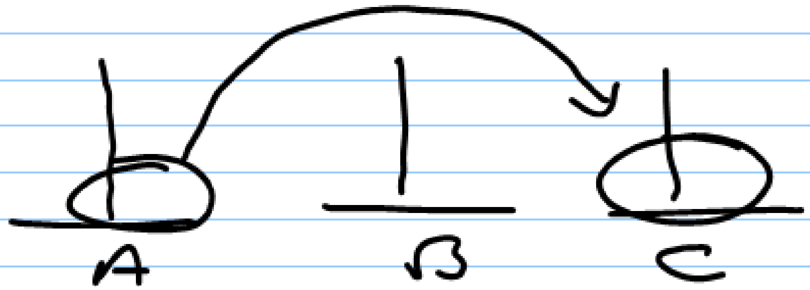
Örnek: Hanoi Kulesi Problemleri

n farklı disk olsun



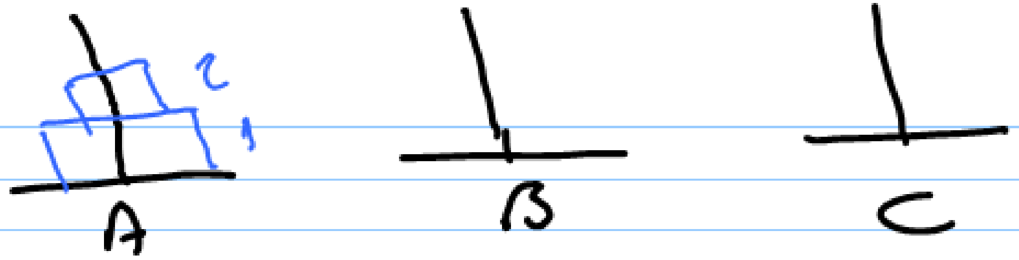
Kural: A'daki diskleri her hamlede 1 tane diski taşımak ve bir çubuktan kendinden küçük bir diske koymadan C çubuğuna en az kaç hamlede geçiririz.

$n=1$



1 hamle

$n=2$



$2 \to B$

$1 \to C$

$2 \to C$

3 hamle

$n=3$

$3 \to C$

$2 \to B$

$3 \to B$

$1 \to C$

$3 \to A$

$2 \to C$

$3 \to C$

7 hamle

$n=4$ 15 hamle

$n=5$ 31 "

$n=1 \quad H(1)=1$

$n=2 \quad H(2)=3=2.1+1$

$n=3 \quad H(3)=7=2.3+1$

$n=4 \quad H(n)=15=2.7+1$

$$\boxed{H(n)=2^n-1}$$

açık

?

$$\boxed{\begin{array}{l} H(1)=1 \\ H(k)=2.H(k-1)+1 \end{array}}$$

kapalı

Çalışma Sorusu: Hanoi probleminin çözümünün sözde kodu veya programı?

Kaynaklar

- *Discrete Mathematics and Its Applications*, Kennet H. Rosen

(Ayrık Matematik ve Uygulamaları, Kennet H. Rosen (Türkçe çeviri), Palme yayıncılık)

- *Discrete Mathematics: Elementary and Beyond*, L. Lovász, J. Pelikán, K. Vesztergombi, 2003.

- *Introduction to Algorithms*, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, 2009.

- *Introduction To Design And Analysis Of Algorithms*, A. Levitin, 2008.