Recursive (Özyinelemeli) Fonksiyonlar:

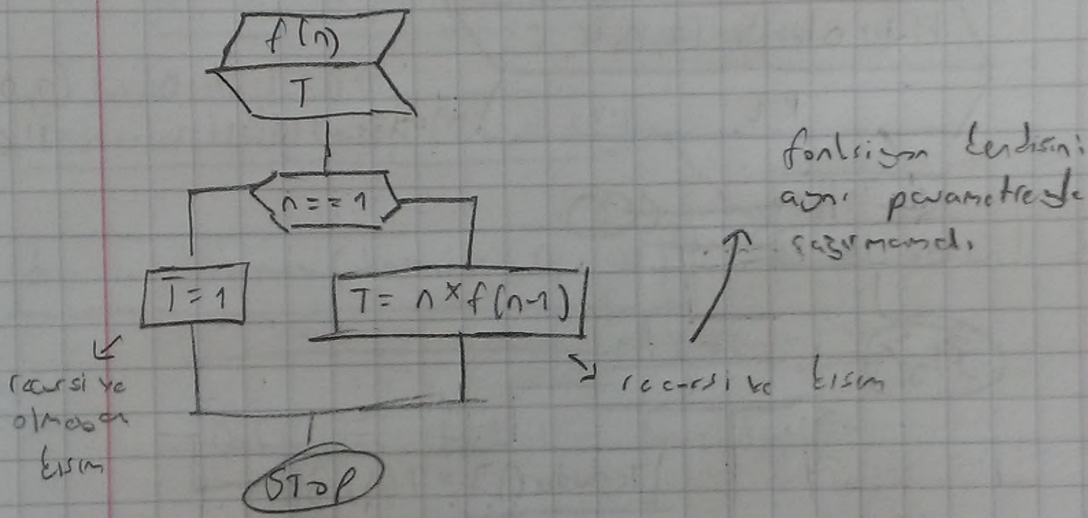
a → a (direkt)

a → b → a (dolaylı)

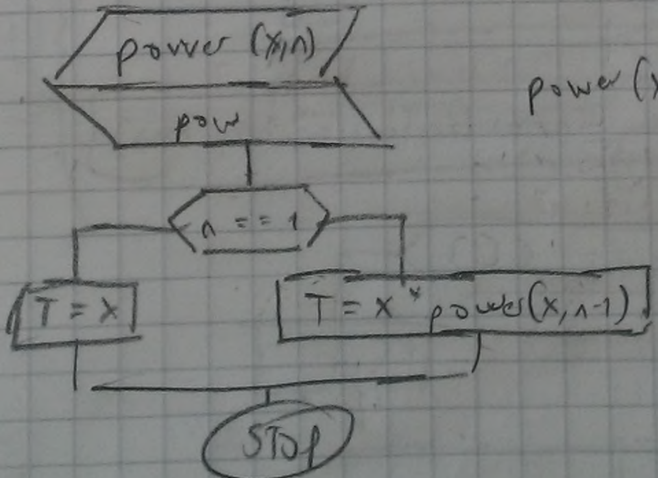
faktöriyel:

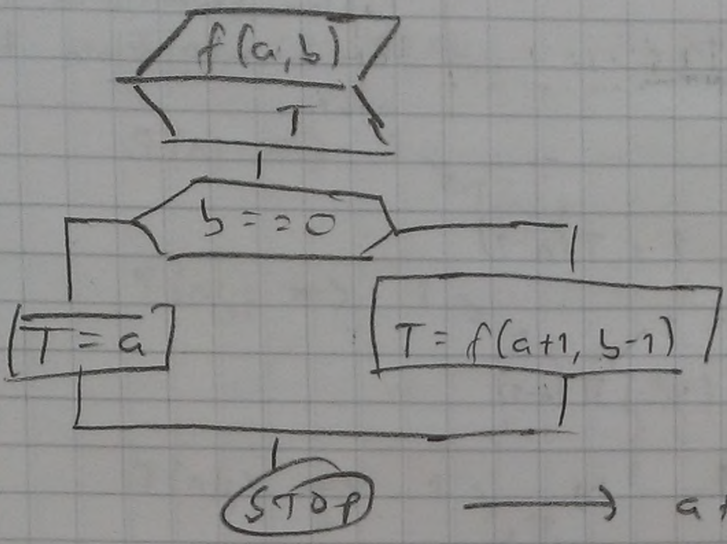
$n = n * n-1$

$\rightarrow f(n) = n * f(n-1)$



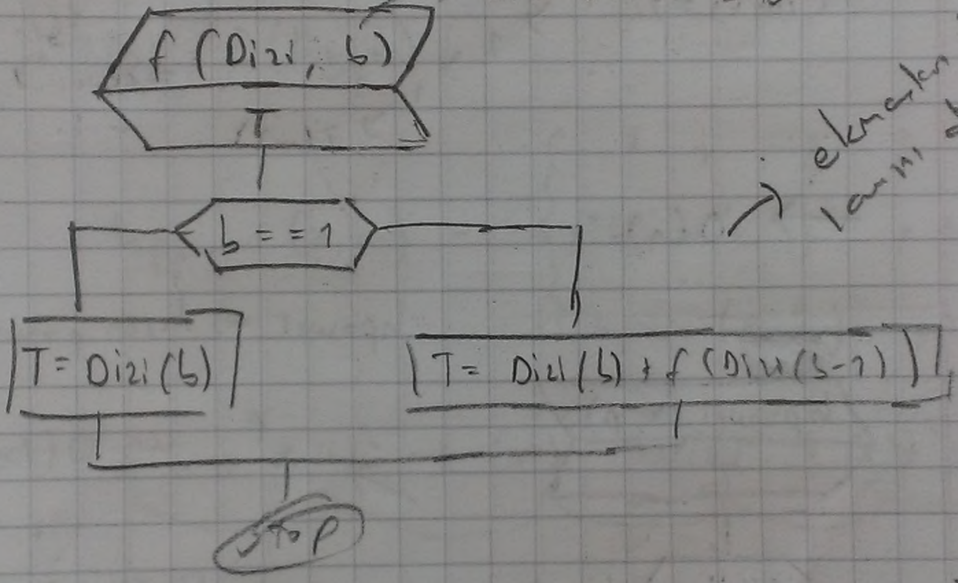
$power(x, n) = x * x^{n-1}$



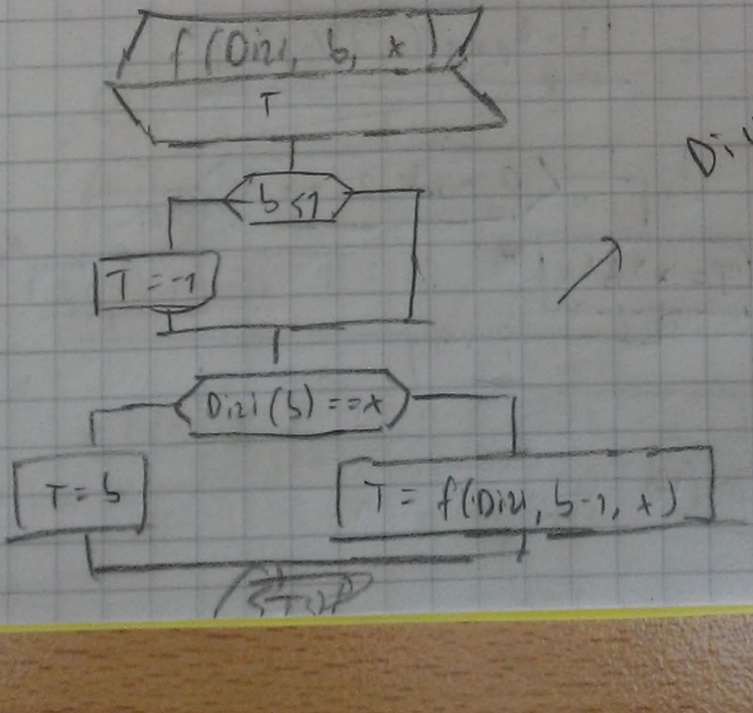


a+b döndürür

dizi eleman sayısı



elemanları toplamını döndürür



Dizinin indisini döndürür

Recursive Tree .

Toplam karmaşıklık ağacın tüm elemanlarının toplamı olur.

fact(x) :

tree ← $T(n) = 1 + T(n-1)$

$T(n) \Rightarrow 1 \searrow T(n-1)$

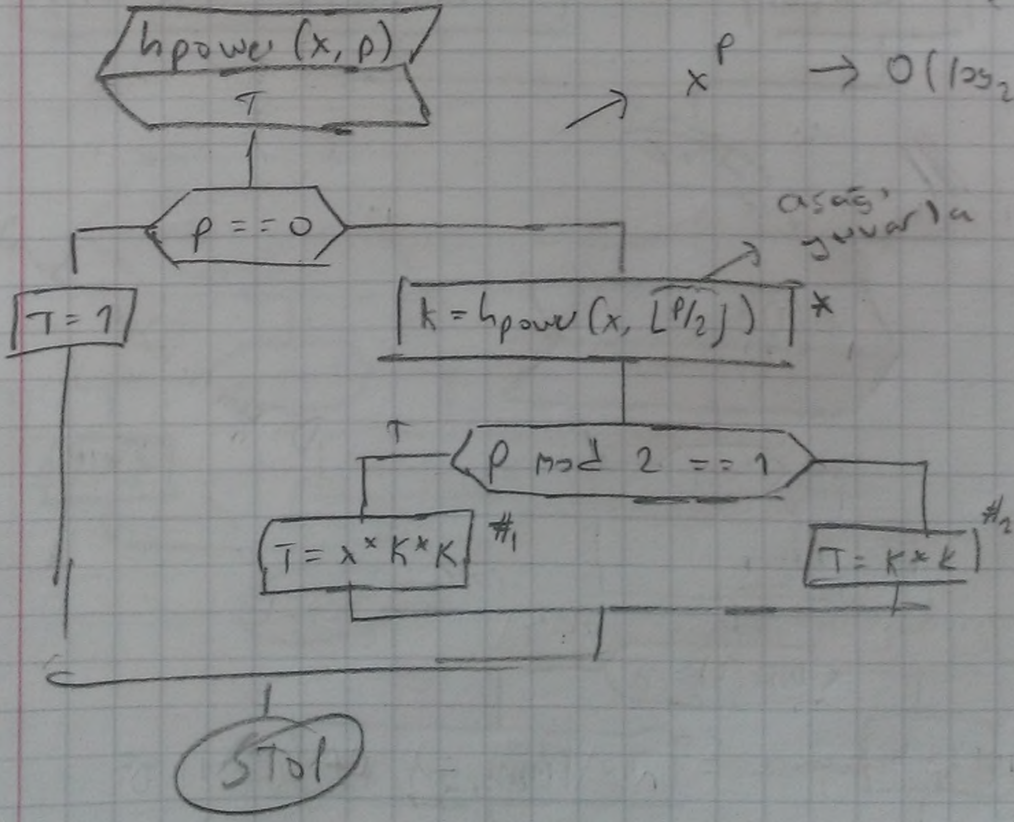
$T(n-1) \Rightarrow 1 \searrow 1 \searrow T(n-2)$

$T(n-2) \Rightarrow 1 \searrow 1 \searrow 1 \searrow T(n-3)$ $\Rightarrow$ $1 \searrow 1 \searrow \cdots \searrow 1$ $h = n$

$\longrightarrow O(n)$

hpower → high power

$x^p \rightarrow O(\log_2 n)$



$$T(n) = 1 + T(n/2)$$

$$T(n) \Rightarrow 1 \searrow T(n/2)$$

$$T(n/2) \Rightarrow 1 \searrow 1 \searrow T(n/4)$$

$$T(n/4) \Rightarrow 1 \searrow 1 \searrow 1 \searrow T\left(\frac{n}{8}\right)$$

$h = \log_2 n$

$$T(n) = O(\log_2 n)$$

$$hpower(x,p) = \begin{cases} 1, & p == 0 \\ x * hpower(x, p/2)^2, & p \to \text{tek} \\ hpower(x, p/2)^2, & p \to \text{çift} \end{cases}$$

$$x^{32} \to (x^{16})^2 \to (x^8)^2 \to (x^4)^2 \to (x^2)^2 \to (x^1)^2$$

$\log_2 32 = 5$ işlem

$$\left.\begin{array}{l} (*) \to k = hpower(x, \lfloor p/2 \rfloor)^2 \\ (\#_1) \to T = X * K \\ (\#_2) \to T = K \end{array}\right\}$$

olsa idi

$T(n) = 1 + 2T(n/2)$

olurdu

$h = \log_2 n$

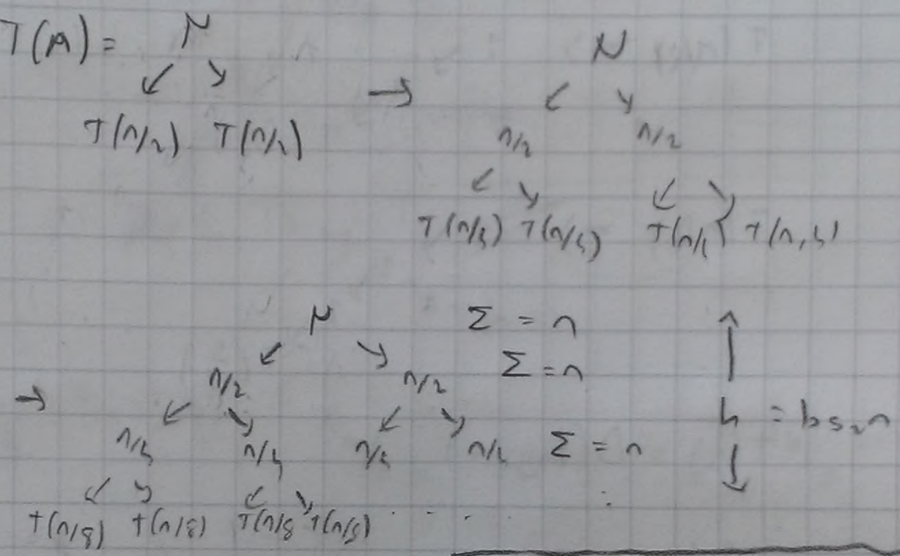
$$T(n) = \sum_{i=0}^{\log_2 n} 2^i = 2 \cdot 2^{\log_2 n} - 1$$

$$= O(n)$$

Ex/ $T(n) = 2T(n/2) + n$ bir ağaç olsun:

↳ quick sort ve merge sort ağacı

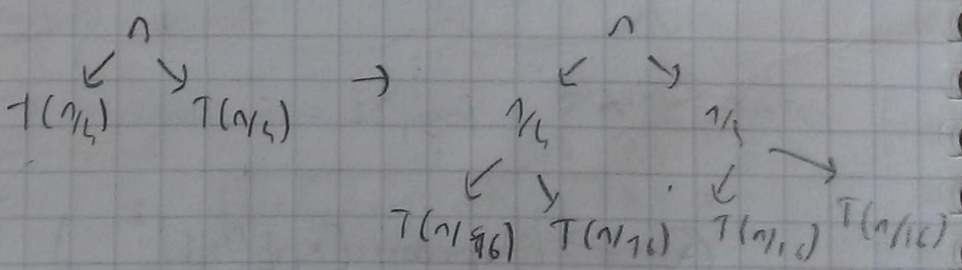
$T(n) =$



$T(n) = n \cdot \log_2 n$

$$\sum_{i=0}^{n} A^i = \begin{cases} \dfrac{A^{n+1}-1}{A-1} & (A>1) \\ \dfrac{1-A^{n+1}}{1-A} & (A<1) \end{cases}$$

Ex/ $T(n) = 2T(n/4) + n$



$h = \log_4 n$

$T(n) = n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} \dots$

$= n\left(1 + \frac{1}{2} + \frac{1}{4} \dots\right) = n \cdot \sum_{i=0}^{\log_4 n} \left(\frac{1}{2}\right)^i$

$= n \cdot \dfrac{1 - \left(\frac{1}{2}\right)^{\log_4 n + 1}}{1 - \frac{1}{2}} = 2n - \sqrt{n}$

$O(n) \longleftarrow$

c) $T(n) = 8T(n/2) + n$

n

$T(n/2) \quad T(n/2) \quad \dots \rightarrow$ 8 tane $T(n/2)$

n

$n/2 \quad n/2 \quad \dots \rightarrow$ 8 tane $n/2$

$T(n/4) \quad T(n/4) \quad \dots\dots \rightarrow$ 64 tane $T(n/4)$

$\rightarrow$ 64 tane $\frac{n}{4}$

$h = \log_2 n$

$T(n) = n + 8 \cdot \frac{n}{2} + 64 \cdot \frac{n}{4} \dots$

$= n + 4n + 8n + 16n$

$= n(1 + 4 + 8 + 16 \dots)$

$$T(n) = n \cdot \sum_{n=0}^{\log_2 n} (4)^{\log_2 n + 1}$$

$$= n \cdot \frac{4^{\log_2 n + 1} - 1}{4 - 1} = n \cdot \frac{4 \cdot n^{\log_2 4} - 1}{3}$$

$$= \frac{4}{3} n^3 - \frac{n}{3}$$

$\rightarrow O(n^3)$

Ex

$$T(n) = 8T(n/2) + n^2$$

n^2

$T(n/2)$ $T(n/2)$ . . . 8 tane $T(n/2)$

n^2

$(n/2)^2$ $(n/2)^2$. → 8 tane $(n/2)^2$

$T((n/4)^2)$ $T((n/4)^2)$ . . → 64 tane $(n/4)^2$

→ 64 · 8 tane $(n/8)^2$

. . .

$h = \log_2 n$

$T(n) = n^2 + 2n^2 + 4n^2 \dots = n^2(1+2+4 \dots)$

$$= n^2 \cdot \sum_{i=0}^{\log_2 n} 2^i = n^2 \cdot \frac{2^{\log_2 n + 1} - 1}{2-1} = \frac{2 \cdot 2^{\log_2 n} - 1}{1} n^2$$

$$= (n-1) \cdot n^2 \rightarrow O(n^3)$$

Ex

$$T(n) = \overbrace{8\,T(n/2)}^{\text{recursive kısım}} + \overbrace{1}^{\text{recursive olmayan}}$$

$\left(\frac{n}{2}\right) \rightarrow 1$

$\downarrow$

$\left(\frac{n}{2}\right)^0$

1

↙ ↓

$T(n/2)$ $T(n/2)$ ... $\rightarrow$ 8 tane $T(n/2)$

1

↙ ↓

1 1 ... $\rightarrow$ 8 tane 1

1

↙ ↓ ...

1 1

↙ ↘

$T(n/4)$ $T(n/4)$ .. $\rightarrow$ 64 tane $T(n/4)$

1

1 ↙ ↓ 1 ...

1 ↙ ↓ 1 ......

$\rightarrow$ 64 tane 1

$h = \log_2 n$

$T(n) = 1 + 8 + 64 + \ldots$

$$= \sum_{i=0}^{\log_2 n} (8)^i = \frac{8^{\log_2 n + 1} - 1}{8-1}$$

$$= \frac{8 \cdot n^{\log_2 8} - 1}{7} = \frac{8n^3 - 1}{7} \rightarrow O(n^3)$$

Ex

$T(n) = 4T(n/3) + 1$

$$
\begin{array}{ccc}
 & 1 & \\
\swarrow & \downarrow & \\
T(n/3) & T(n/3) & \cdots
\end{array}
\quad \rightarrow \quad 4 \text{ tane } T(n/3)
$$

$$
\begin{array}{ccc}
 & 1 & \\
\swarrow & \downarrow & \\
1 & 1 & \cdots
\end{array}
\quad \rightarrow \quad 4 \text{ tane } 1
$$

$$\rightarrow \quad 16 \text{ tane } 1$$

$h = \log_3 n$

$$T(n) = \sum_{i=0}^{\log_3 n} 4^i = \frac{4^{\log_3 n + 1} - 1}{4 - 1}$$

$$= \frac{4 \cdot n^{\log_3 4} - 1}{3}$$

$O(n^{\log_3 4})$

Ex/

$$T(n) = 9\,T(n/3) + 1$$

1

$T(n/3)$ . . . . $\rightarrow$ 9 tane $T(n/3)$

1

1 1 . . . $\rightarrow$ 9 tane 1

$\rightarrow$ 81 tane 1

$h = \log_3 n$

$$T(n) = \sum_{i=0}^{\log_3 n} 9^i = \frac{9^{\log_3 n+1} - 1}{9-1} = \frac{9 \cdot n^{\log_3 9} - 1}{8}$$

$$= \frac{9n^2 - 1}{8} \rightarrow O(n^2)$$

Ex/

$$T(n) = 4\,T(n/2) + n$$

n

$T(n/2)$ $T(n/2)$ . . . $\rightarrow$ $4 \times T(n/2)$

n

$n/2$ $n/2$ . . .

$T(n/4)$ $T(n/4)$ . . . $\rightarrow$ $16 \times T(n/4)$

$h = \log_3 n$

$$T(n) = n + 4\cdot\frac{n}{3} + 16\cdot\frac{n}{9}\cdots$$

$$= n\left(1 + \frac{4}{3} + \frac{16}{9}\cdots\right)$$

$$= n\cdot\sum_{i=0}^{\log_3 n}\left(\frac{4}{3}\right)^i = n\cdot\frac{\left(\frac{4}{3}\right)^{\log_3 n+1}-1}{\frac{4}{3}-1}$$

$$= n\cdot\frac{\frac{4}{3}\cdot\frac{4^{\log_3 n}}{3^{\log_3 n}}-1}{1/3}$$

$$= 3n\left(\frac{4\cdot 4^{\log_3 n}}{3n}-1\right)$$

$$= 4\cdot 4^{\log_3 n} - 3n \rightarrow O(n^{\log_3 4})$$

$2.\ T(n-2)+1$

Ex: $T(n) = 2\,T(n-1)+1$

$$\begin{tikzcd}[column sep=small] & 1 \arrow[dl] \arrow[dr] & \\ T(n-1) & & T(n-1) \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & & & 1 \arrow[dll] \arrow[drr] & & & \\ & 1 \arrow[dl] \arrow[dr] & & & & 1 \arrow[dl] \arrow[dr] & \\ T(n-2) & & T(n-2) & & T(n-2) & & T(n-2) \end{tikzcd}$$

$h = n$

$T(n) = 1 + 2 + 4 \ldots$

$$= \sum_{i=0}^{n} 2^i = \frac{2^{n+1}-1}{2-1} = 2.2^n - 1$$

$\rightarrow O(2^n)$

	n^2	2^n
1	1	2
10	100	1024
20	400	1M
30	900	1B ...

$2^n - n^2$ Karşılaştırması

$T(n) = T(n-1) + T(n-2) + 1 \rightarrow$ Fibonacci

$$\begin{array}{ccc} & 1 & \\ \swarrow & & \searrow \\ T(n-1) & & T(n-2) \end{array}$$

$$\begin{array}{cccc} & & 1 & \\ & \swarrow & & \searrow \\ & 1 & & 1 \\ \swarrow & \downarrow & \swarrow & \searrow \\ T(n-2) & T(n-3) & T(n-3) & T(n-4) \ldots \end{array}$$

→ Ağacın kolları farklı uzunlukta olduğunda

$T(n) = 2T(n-1) + 1$ varsayalım $\rightarrow O(n^2)$

↳ bundan daha küçük olmalı

Gerçekte $\rightarrow O(k^n)$, $k = \frac{1+\sqrt{5}}{2}$ (altın oran)

~~ALTERNATİF YOL~~

$$F = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

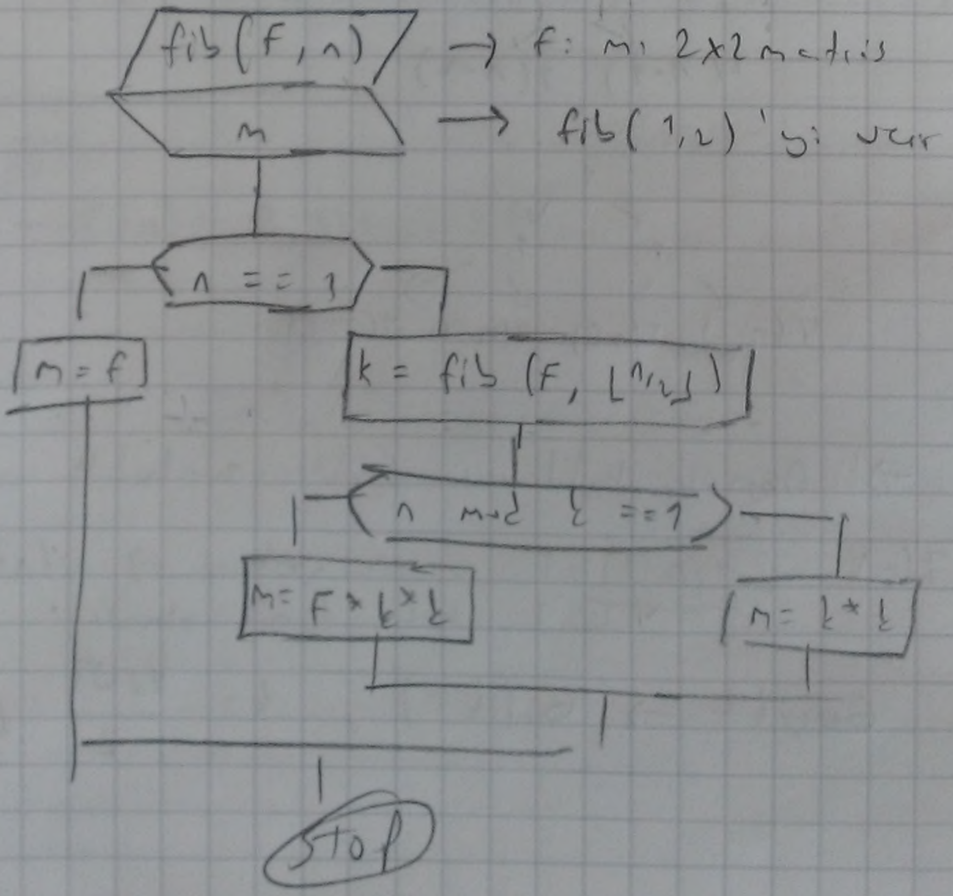
$$F^2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

$$F^3 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}$$

$$F^4 = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$$

$$F^5 = \begin{bmatrix} 8 & 5 \\ 5 & 3 \end{bmatrix}$$

$$F^n = \begin{bmatrix} fib(n+1) & fib(n) \\ fib(n) & fib(n-1) \end{bmatrix}$$ → power serisi ($O(\log_2 n)$)



$$T(n) = T(n/2) + c$$ → önemsiz işlem sayısı (n'den bağımsız)

$$= O(\log_2 n).$$

Sepet Algoritması :

S(N×2)

	Sepet ID	Ürün ID
1 →	3	42
2 →	3	15
:	3	42
:	:	:
	5	6
N →	5	15

A)

Sepet ID → sepet(N)

$$
\begin{array}{c|ccccccccc|}
 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
 & & & 1 & & & & & & \\
 & & & 0 & & & & & & \\
\text{ürün ID} & & & 1 & & & & & & \\
 & & & 0 & & & & & & \\
 & & & 0 & & & & & & \\
 & & & 1 & & & & & & \\
 & & & 0 & & & & & &
\end{array}
$$

↓ ürün(N)

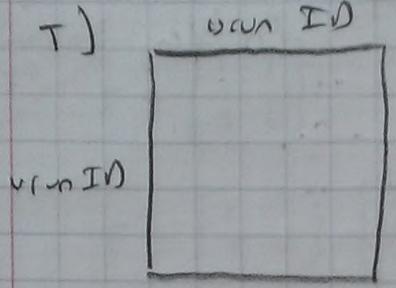
i=1 | 1

N

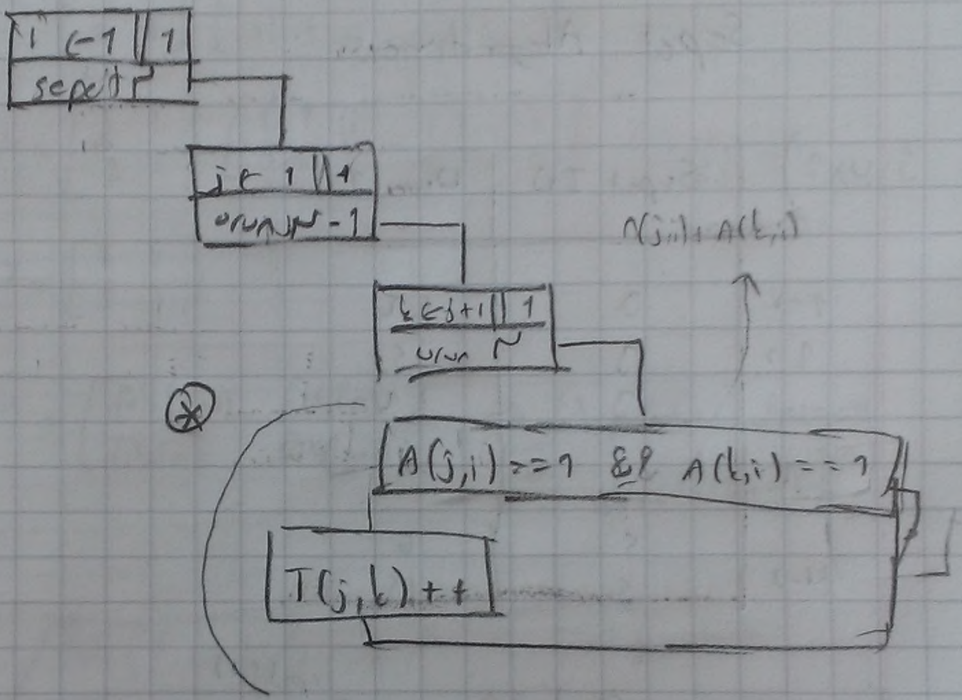
$\boxed{A(S(i,2), S(i,1)) = 1}$

S'dan A'ya geçiş?

T)



→ A'dan T'ye geçiş?

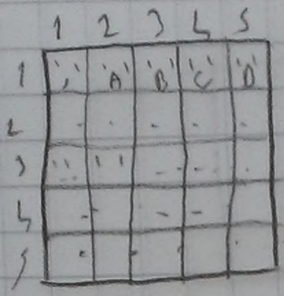


⊛ → $T(j,k) \mathrel{+}= A(j,i) * A(k,i)$

(iyileştirme)

Daha da iyisi Apriori algoritması

Şifreleme Alg.

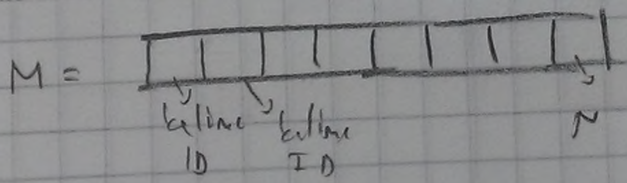


A B - C D

12 13 32 14 15 ...

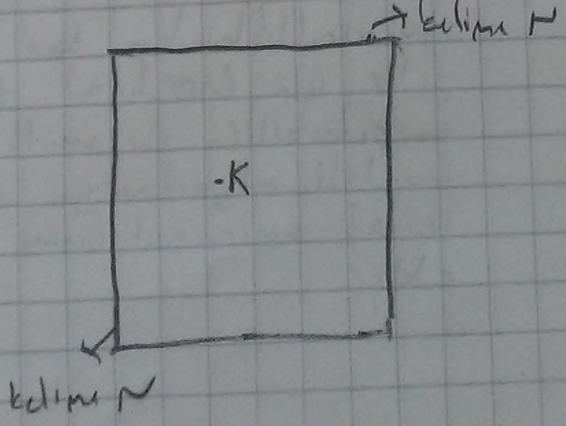
Şifrelenmemiş bir mesajda, "A" harfi genelde en çok kullanılan harf olması mantığıyla mesaj tahmin olarak çözülebilir. Bunun sebebi bir harfin numarasının hep aynı olmasıdır.

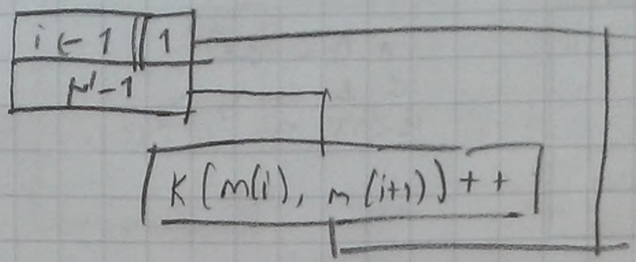
Bunun önüne geçmek için bir harfin bir dilde kullanılma sıklığı kadar farklı numarası olur. Böylece çözülecek metnin tüm sayıları benzer frekansta olur, kırılması zorlaşır.

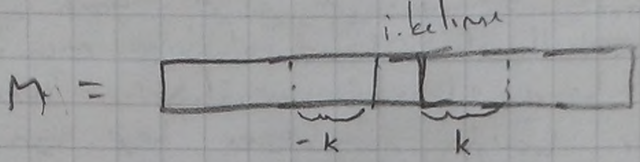


→ metin (bu kelime için ID var)

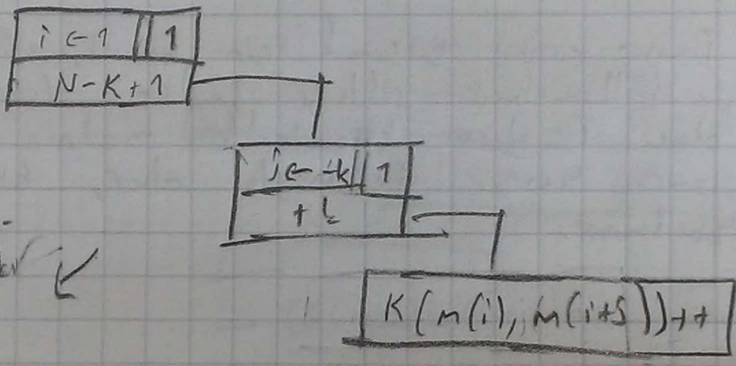
Arka arkaya en sık tekrar eden kelime ikilisi?







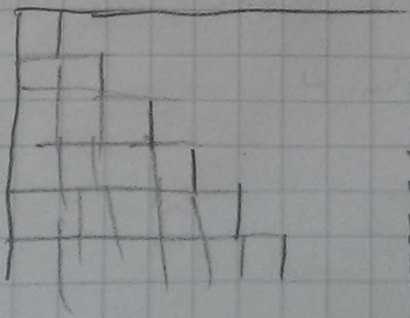
→ ortada i'nin olduğu kelime grupları



kelime grup-larının içindeki anlamsal yakınlık

HARİTA → şehirler arası uzaklık

↓



bu uzaklıklardan her bir şehir için koordinat tahmini yapılabilir. Bu mantıkla yukarıdaki algoritmaya uygularak anlamsal uzay elde edilir.