Horner Yöntemi
(Bottom up)

$$\underbrace{\text{Transform and Conquer}}$$

Çarpanlara ayırma, L'Hopital gibi yöntemler

bu Algoritmayı kullanır

$x^0 = 1$

$$a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 = ?$$

Ortak Çarpan → $x(a_3x^2 + a_2x + a_1) + a_0$

Tekrar → $x \cdot (x(a_3x + a_2) + a_1) + a_0$

Tekrar → $x(x(a_3(x) + a_2) + a_1) + a_0$

! En başta 0. Bir sabit sayı alıyoruz ve x ile çarpıyoruz. Sonra elimizdeki x'li ifadeye sabit sayı ekliyoruz ve yine x ile çarpıyoruz mantığı ile bir algoritma kuruyoruz

1-) $a_3(x)$

2-) $(a_3(x) + a_2)$

3-) $(a_3(x) + a_2)x$

4-) $(a_3x^2 + a_2x) + a_1$

5-) $(a_3x^2 + a_2x + a_1) \cdot x$

6-) $(a_3x^3 + a_2x^2 + a_1x + a_0$

→ Bu algoritmanın karmaşıklığı =

Temel işlemlerin sayımı ile tespit edelim.

Her $k \le n$ için 1 karşılaştırma, 1 çarpma, 2 toplama, 1 çıkarma

$+$

$1 \rightarrow k > n$ için karşılaştırma

$$\rightarrow \left(\sum_{k=1}^{n} 5 \right) + 1 \text{ maliyetimiz var}$$

$\rightarrow 5n + 1$

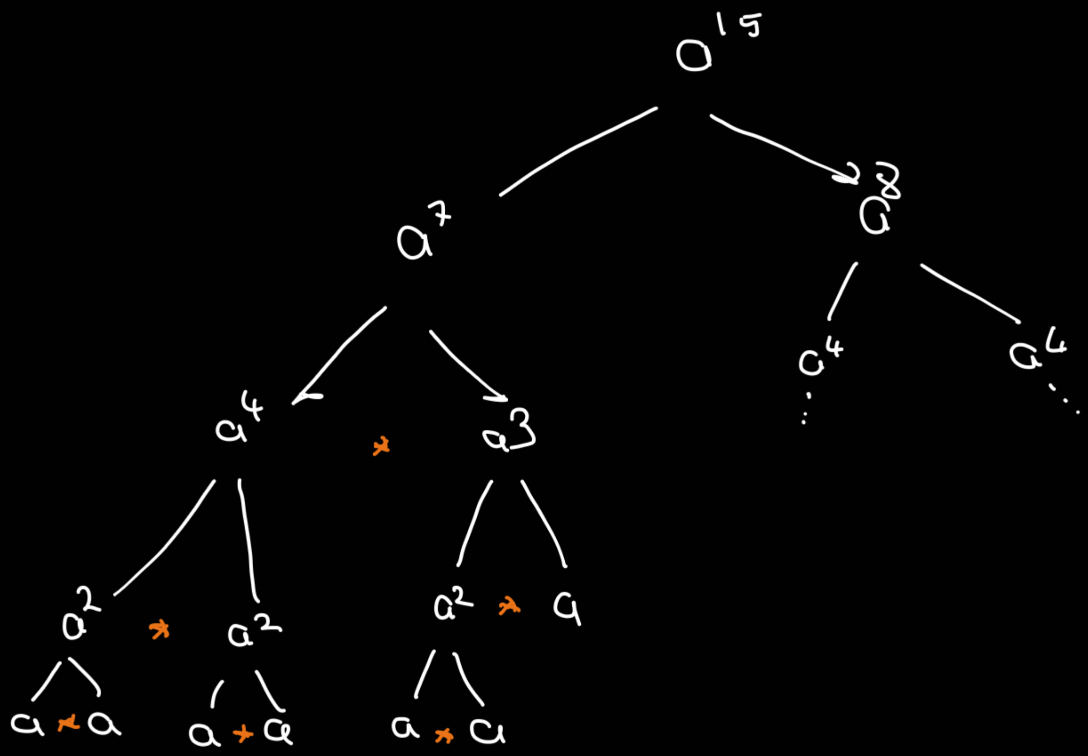
Horner Yöntemi;

Divide and Conquer Alg.

$$a^4 = a.a.a.a \longrightarrow \Theta(n)$$

$$a^n \begin{cases} 1 & n = 0 \\ a & n = 1 \\ a^{\lfloor n/2 \rfloor} . a^{\lceil n/2 \rceil} & n > 1 \end{cases} \longrightarrow a^5 = a^2 . a^3$$

$(a^x . a^{x+1}$



$$M(n) = M(\lfloor n/2 \rfloor) + M(\lceil n/2 \rceil) + 1 \quad \text{For } n > 1$$

Maliyet $\Theta(\log n)$

Binary Exponention

$p = 1$ başlangıç

$p = 2p + b_i$

left to right B.E.

→ a ve n pozitif açılımındaki $b_I, \dots b_0$ listesi kullanıcıdan alınacak ve a^n cevabı verilecek.

$a^{13} = \underset{3}{1}\;\underset{2}{1}\;\underset{1}{0}\;\underset{0}{1}_{(2)}$

$n = 13$
$\hookrightarrow I = 4$ → i değerimiz 3'e kadar sayacak
1-1'e kadar döngü çalışır.

Maliyet = $\log_2 n - 1$ adet karşılaştırma
↳ Her karşılaştırma için $(b-1)$ ile $(b-1) \cdot 2$ arasında çarpma yapılır.

$b = \log_2 n$

→ Eğer sınavda sorsal

$$(b-1) \leq M(n) \leq 2(b-1)$$ → isteriz

Sınavda Çıkabilir Dedi

Right to Left Binary Expon.

$$a^n = a^{b_l 2^l + \dots + b_i 2^i + \dots + b_0} = a^{b_l 2^l} + \dots + a^{b_i 2^i} + \dots \; a^{b_0}$$

$a^{13} = a^{1101}$

b_i	1	1	0	1
term	a^8	a^4	a^2	a
çarpım	$a^5 \cdot a^8 = a^{13}$	$a^4 \cdot a = a^5$		a

term $= a$

eğer $b_0 = 1$ ise
$p = a$

eğer $b \neq 1$ ise
$p = 1$

$i = 1$'den l'ya kadar
term = term . term

eğer $b_i = 1$ ise $p = p \cdot$ term

Right to Left ve Left to Right

algoritmalarının ikisinde de maliyet logaritmiktir.