Problem: compute $x^k$ where $x$ is real number and $k$ is an integer

Algorithm 1
1. Compute $n = 2^k$
   1. for $i ← 1$ to $k$
      1. $n ← 2^n$
   2. Compute $x^n = x \cdot x \cdot \ldots \cdot x$
      2. for $i ← 1$ to $n$
         2. $x ← x \cdot x$
   Cost: $k + 2^k$ multiplications

Algorithm 2
1. for $i ← 1$ to $k$
   2. $x ← x \cdot x$
Cost: $k$ multiplications

Problem: Given coefficients $a_0, a_1, \ldots, a_n$ and $x$, evaluate the polynomial $p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$.

An obvious way is to compute each term and add it to the sum of the others that are already computed.

Naive-poly($a$, $n$, $x$)
1. $P ← a[0]$
   2. for $i ← 1$ to $n$
      2. $xpower ← xpower \cdot x$
      2. $p ← p + a[i] \cdot xpower$
   3. return $p$

Running time: $2n$ multiplication and $n$ additions.

Horner’s algorithm
To observe that $p(x)$ can be rewritten as follows.
$p(x) = (\ldots (a_0 x + a_1) x + a_2) x + \ldots + a_i) x + a_{i+1} x + \ldots + a_n x + a_n$

Horner-poly($a$, $n$, $x$)
1. $P ← a[n]$
2. for $i ← n-1$ to 0
   2. $p ← p \cdot x + a[i]$
4. return $p$

Running time: $n$ multiplications and $n$ additions.

Monomial

Evaluating Polynomial Functions

Vector and Matrix Multiplication

Definitions
Let $V = (v_1, v_2, \ldots, v_n)$ and $W = (w_1, w_2, \ldots, w_n)$ be two $n$-vectors, then the dot product of $V$ and $W$ is defined as $V \cdot W = \sum_i v_i \cdot w_i$.

Let $A$ be an $m \times n$ matrix and $V$ be an $n$-vector, then the product $W = AV$ is defined as $w_i = \sum_j a_{ij} v_j$.

Let $A$ be an $m \times n$ matrix, $B$ an $n \times q$ matrix, then the product $C = AB$ is an $m \times q$ matrix and its elements are defined as $c_{ij} = \sum_k a_{ik} b_{kj}$.

i.e., is the dot product of the $i$th row of $A$ and $j$th column of $B$. 
Up to 1969, $n^3$ looked like the complexity of matrix multiplication.


Matrix Multiplication

Time complexity

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

If $A$ and $B$ are both $n \times n$ square matrix, so is $C$. Each element $c_{ij}$ is obtained as a dot product of $i$th row of $A$ and $j$th column of $B$, thus by $n$ multiplications and $n-1$ additions. As there are $n^2$ elements in $C$, the total cost is

$$n^2 (n + n-1) = 2n^3$$

arithmetic operations.

Can we do better?
Time complexity
- Multiplications: \( T_m(2^k) = 7 \cdot T_m(2^{k-1}) = 7^k \cdot T_m(2^0) = 7^k \)

\[ 7^k = 2^{\log_2 7^k} = n^{\log_2 7} \]

\[ \log_2 7 \approx 2.81 \]

\[ n^{2.81} < n^3 \]

We arrived at a better performance.

- V. Pan (1978) IBM
- ...
- V. Strassen (1986): \( n^{2.5} \)
- D. Coppersmith et al (1987): \( n^{2.376} \)
- \( c_1 n^2 \leq \text{complexity}(n) \leq c_2 n^{2.36} \)
- We do not the complexity yet. Probably because the problem is a complete information problem.