COT 5407: Introduction to Algorithms

Giri Narasimhan

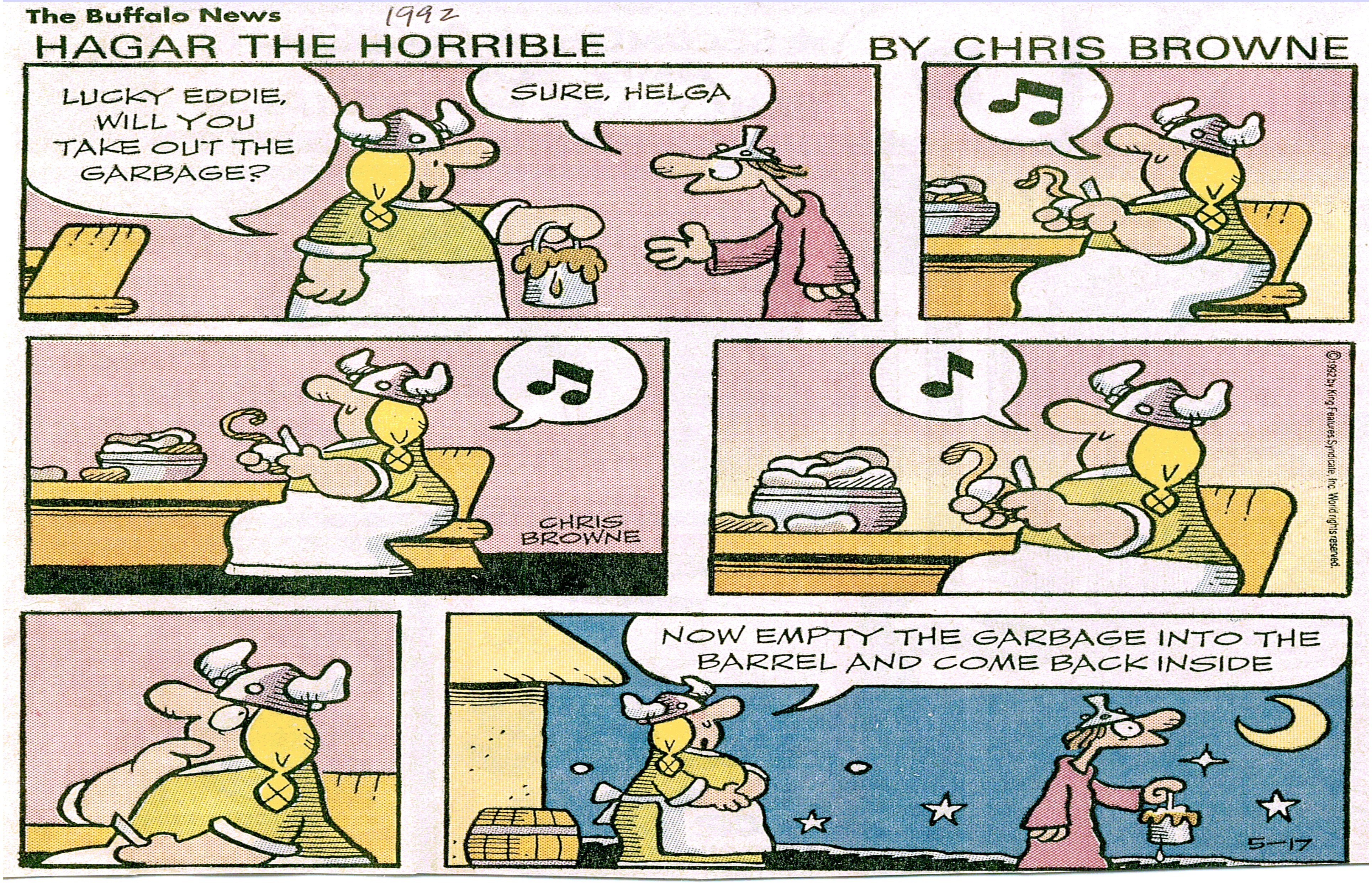
ECS 254A; Phone: x3748

giri@cis.fiu.edu

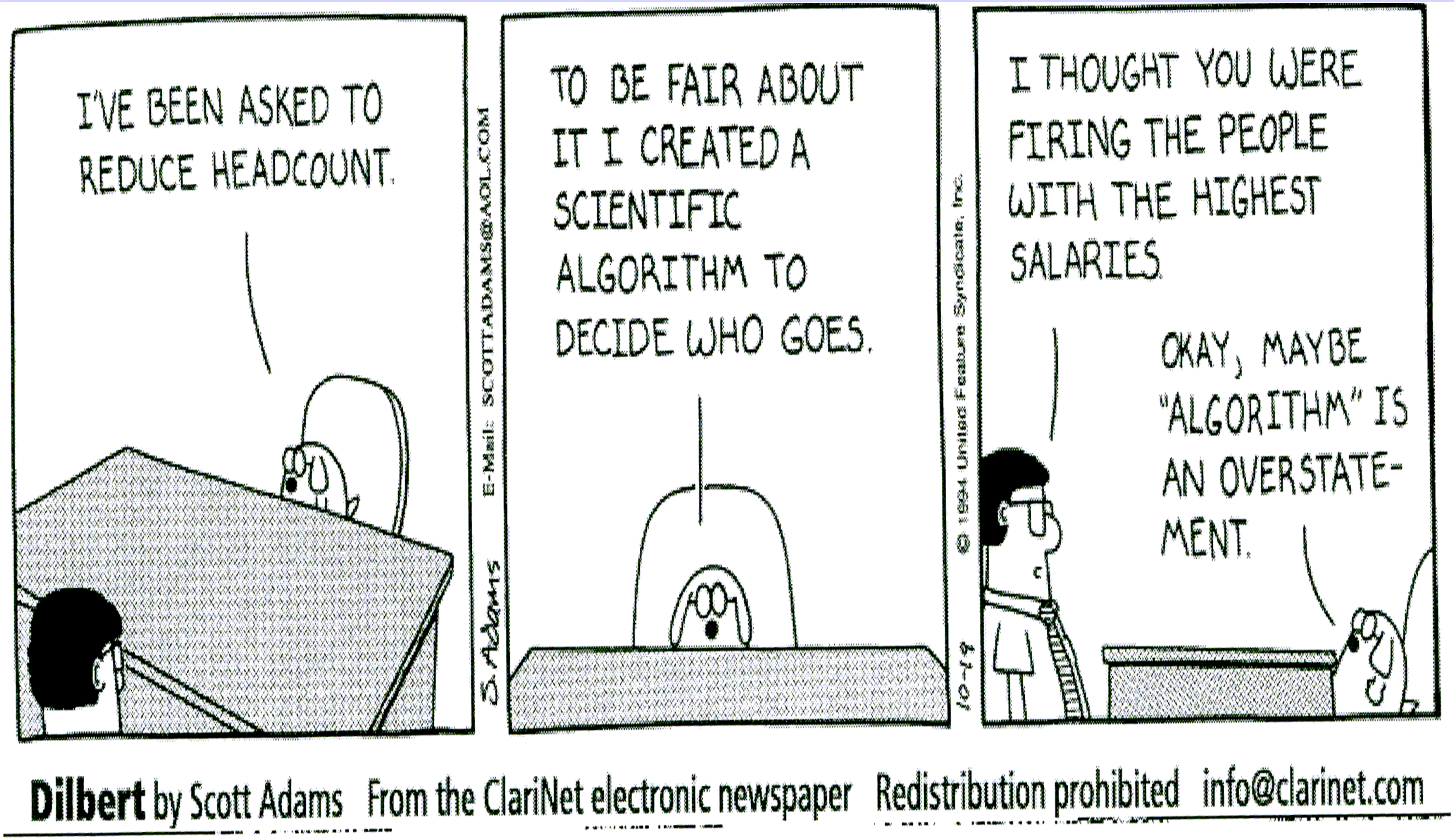
http://www.cis.fiu.edu/~giri/teach/5407F08.html

https://online.cis.fiu.edu/portal/course/view.php?id=285

Algorithms are “recipes”!

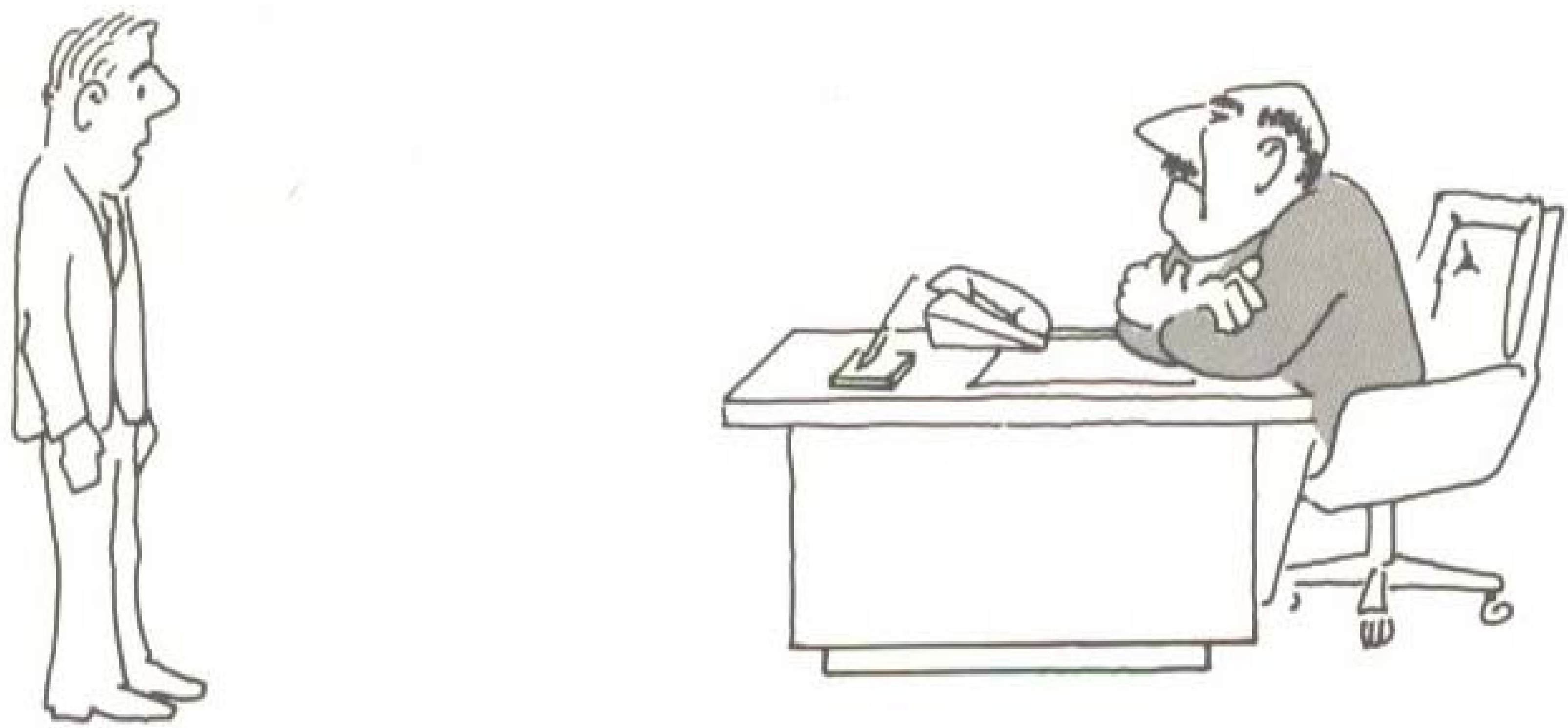


Algorithms can be simple



Why should I care about Algorithms?

Cartoon from *Intractability* by Garey and Johnson



"I can't find an efficient algorithm, I guess I'm just too dumb."

More questions you should ask

- Who should know about Algorithms?
- Is there a future in this field?
- Would I ever need it if I want to be a software engineer or work with databases?

Why are theoretical results useful?

"I can't find an efficient algorithm, because no such algorithm is possible!"

Cartoon from *Intractability* by Garey and Johnson

Why are theoretical results useful?

"I can't find an efficient algorithm, but neither can all these famous people."

Cartoon from *Intractability* by Garey and Johnson

Evaluation

• Exams (2)	50%
• Quizzes	10%
• Homework Assignments	30%
• Semester Project	5%
• Class Participation	5%

History of Algorithms

The great thinkers of our field:

- **Euclid, 300 BC**
- **Bhaskara, 6th century**
- **Al Khwarizmi, 9th century**
- **Fibonacci, 13th century**
- **Babbage, 19th century**
- **Turing, 20th century**
- **von Neumann, Knuth, Karp, Tarjan, …**

Al Khwarizmi's algorithm

- **43 X 17**
  - **43 17**
  - **21 34**
  - **10 68 (ignore)**
  - **5 136**
  - **2 272 (ignore)**
  - **1 544**

731

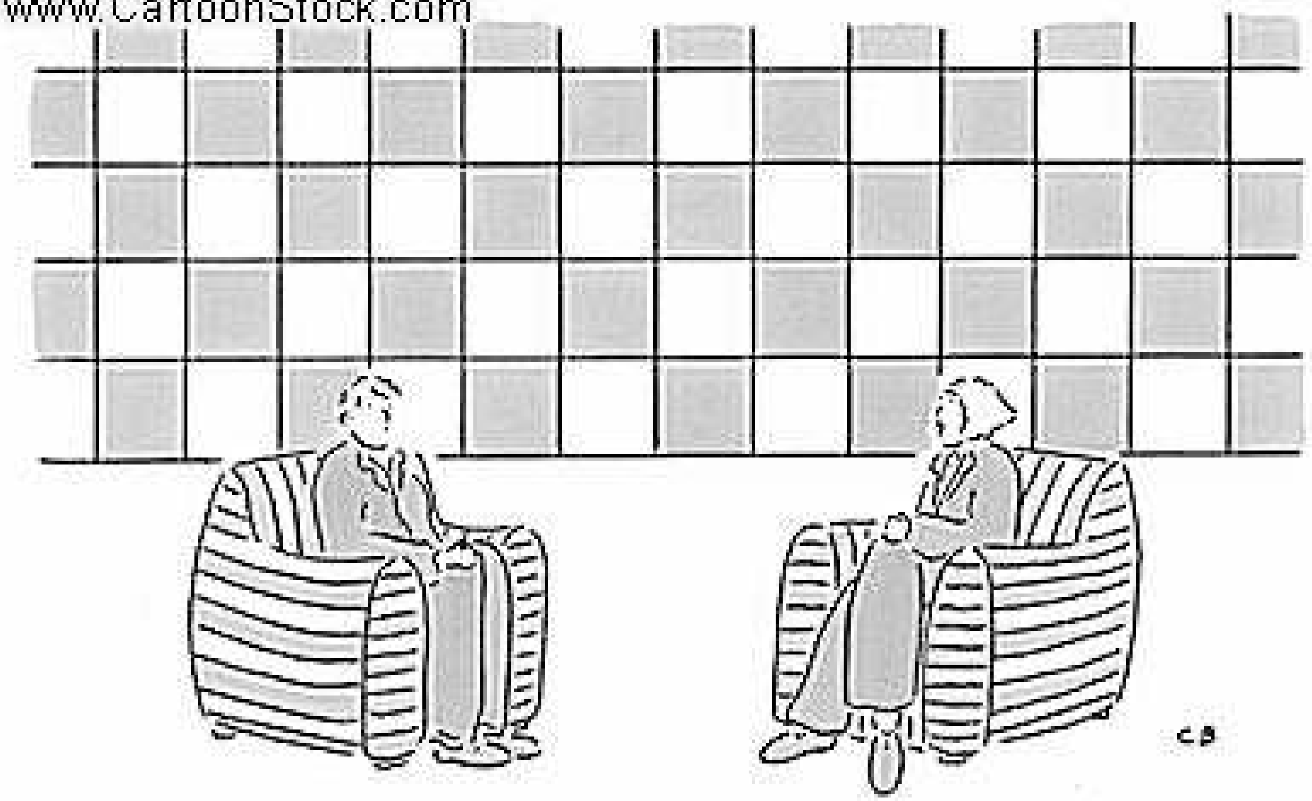
Euclid’s Algorithm

- GCD(12,8) = 4; GCD(49,35) = 7;
- GCD(210,588) = ??
- GCD(a,b) = ??
- **Observation**: [a and b are integers and a ≥ b]
  - **GCD(a,b) = GCD(a-b,b)**
- **Euclid's Rule**: [a and b are integers and a ≥ b]
  - **GCD(a,b) = GCD(a mod b, b)**
- **Euclid's GCD Algorithm:**
  - **GCD(a,b)**

    If (b = 0) then return a;

    return GCD(a mod b, b)

If you like Algorithms, nothing to worry about!



"Calculus is my new Versace. I get a buzz from algorithms. What's going on with me, Raymond? I'm scared."

Search

- You are asked to guess a number X that is known to be an integer lying in the range A through B. How many guesses do you need in the worst case?
  - Use **binary search**; Number of guesses = $\log_2(B-A)$
- You are asked to guess a positive integer X. How many guesses do you need in the worst case?
  - NOTE: No upper bound is known for the number.
  - Algorithm:
    - figure out B (by using **Doubling Search**)
    - perform binary search in the range B/2 through B.
  - Number of guesses = $\log_2 B + \log_2(B - B/2)$
  - Since X is between B/2 and B, we have: $\log_2(B/2) < \log_2 X$,
  - Number of guesses $< 2\log_2 X - 1$

Polynomial Evaluation

- Given a polynomial
  - $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n$

  compute the value of the polynomial for a given value of x.
- How many additions and multiplications are needed?
  - Simple solution:
    - Number of additions = n
    - Number of multiplications = 1 + 2 + … + n = n(n+1)/2
  - Reusing previous computations: n additions and 2n multiplications!
  - Improved solution using **Horner's rule**:
    - $p(x) = a_0 + x(a_1 + x(a_2 + \ldots x(a_{n-1} + x\, a_n))\ldots))$
    - Number of additions = n
    - Number of multiplications = n