Algorithms and Data Structures

Exercise Session 0

About Me

- Ahmet Ala
- 3rd year Computer Science BSc.

About the Course

- 7 Credits
- Introduction to Algorithms and Data Structures
- Sorting algorithms, searching algorithms, abstract data types, greedy approach, dynamic programming, graph algorithms...
- Correctness and Run-Time Analysis
- Continues with Algorithms and Probability(second semester Basisblock Course) and Algorithms, Probability and Computing (Major: Theoretical Computer Science)
- But the content appears everywhere in Theoretische Informatik, Computer Networks, Numerical Methods for Computer Science etc.

Exercise Session Logistics

- CHN D 29
- Monday, 09:15-11.00
- Website for announcements, slides, kahoots etc.
- https://n.ethz.ch/~ahmala/and/

Homeworks (theoretical part)

- Groups of 2 and 3 (changes in every three weeks)
- New exercise sheet every monday, submissions sent either to ahmala@ethz.ch or handed in at the beginning of exercise session
- Deadline for Exercise Sheet 1 is next week (Oct 2, 09:15)
- Only today no peer grading
- 11:00 - 12:00 can be used for peer grading

Point Distribution

- 12* {Weekly theoretical exercise sheet(3 points) + peer grading(1 points)} [In Groups]
- 5* {Code Expert programming tasks(biweekly) (4 points)} [Individual]
- min(0.25, 0.25 * n_points / (0.8 * max_points))
- 0.25 bonus for 80% of full possible points

Induction Proofs

- Base Case(usually n=1): Show, the statement holds for n=1
- Induction Hypothesis(n=k): Assume the statement holds for n=k
- Induction Step(usually n=k+1): Show, the statement also holds for n=k+1
- (usually) By the principle of mathematical induction, this is true for any positive integer n.

Example

Prove by mathematical induction that for any positive integer n,

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

Logarithms

- Quite important for Asymptotic Analysis

$$
\begin{aligned}
\log_a(bc) &= \log_a(b) + \log_a(c) \\
\log_a(b^c) &= c\log_a(b) \\
\log_a(1/b) &= -\log_a(b) \\
\log_a(1) &= 0 \\
\log_a(a) &= 1 \\
\log_a(a^r) &= r \\
\log_{1/a}(b) &= -\log_a(b) \\
\log_a(b)\log_b(c) &= \log_a(c) \\
\log_b(a) &= \frac{1}{\log_a(b)} \\
\log_{a^m}(a^n) &= \frac{n}{m}, \qquad m \neq 0
\end{aligned}
$$

$$\log_b a = \frac{\log_d a}{\log_d b}$$

De l’Hopital Rule

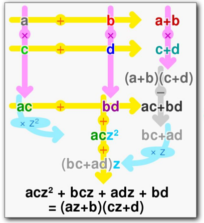
Let $f, g : \mathbb{R} \to \mathbb{R}$ be differentiable functions with $f(x) \to \infty$, $g(x) \to \infty$ for $x \to \infty$. If $\lim_{x\to\infty} \frac{f'(x)}{g'(x)}$ exists, then

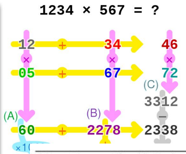
$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$$

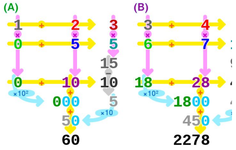
$$1 < log(log(n)) < log(n) < \sqrt{n} < n < n \cdot log(n) < n^2 < 2^n < n!$$

Pasture Break Succeeded

Karatsuba Algorithm(not relevant as well)







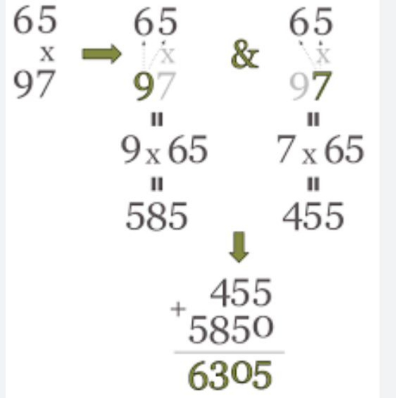
Algorithm 1 The ADK algorithm for long multiplication

INPUT: Degree n, and radix $b = 2^t$

INPUT: $\boldsymbol{x} = [x_0, \ldots, x_{n-1}], \boldsymbol{y} = [y_0, \ldots, y_{n-1}]$ where $x_i, y_i \in [0, b-1]$

OUTPUT: $\boldsymbol{z} = [z_0, \ldots, z_{2n-2}, 0]$, where $z_i \in [0, b^2 - 1]$ and $\boldsymbol{z} = \boldsymbol{xy}$

```
function ADKMUL(x, y)
    for i ← 0 to n − 1 do
        d_i ← x_i y_i
    end for
    s ← d_0
    z_0 ← s
    for k ← 1 to n − 1 do
        s ← s + d_k
        t ← s
        for i ← 1 + ⌊k/2⌋ to k do
            t ← t + (x_i − x_{k−i})(y_{k−i} − y_i)
        end for
        z_k ← t
    end for
    for k ← n to 2n − 2 do
        s ← s − d_{k−n}
        t ← s
        for i ← 1 + ⌊k/2⌋ to n − 1 do
            t ← t + (x_i − x_{k−i})(y_{k−i} − y_i)
        end for
        z_k ← t
    end for
    return z
end function
```



Groups