- Design and Analysis of Algorithms
- Algorithm: precise instructions on how to perform a task
- Design: solve hard problem using basic operations
  - abstraction
  - reduction
    - divide and conquer
    - dynamic programming
    - greedy
- Analysis
  recall: Huffman tree

- correctness: why can this tree be used for encoding/decoding?
- optimality: why merge two least frequent characters?
  (it produces the shortest encoding)
- time/space complexity: how much time/memroy is needed to run the alg with n characters?
  (- robustness - fairness - privacy ...)
- **Asymptotic Analysis**

- **Def.** \( f(n) = \Theta(g(n)) \) if \( \exists \) constants \( C_1, C_2 > 0 \) such that
  \[
  \forall n > 0 \quad C_1 g(n) \leq f(n) \leq C_2 g(n)
  \]

- **Property**: 1. Small values of \( n \) does not matter
  \[
  f(n) = (0, 0, 0) \quad g(n) = n^2
  \]
  \( \quad \) still \( f(n) = O(g(n)) \)

  2. Drop the insignificant term
  \[
  3n^3 + 50n^2 + 100n + 250 = O(n^3)
  \]
  \[
  1 \leq \log n < \sqrt{n} < n < n^2 < n^3 < 2^n < 3^n < \ldots
  \]

- **Why asymptotic?**

  - Lazy
  \[
  \log n + \log(n-1) + \ldots + \log(1) = \Theta(n \log n)
  \]

  \[
  \geq \frac{n}{2} \log \frac{n}{2} = \frac{n}{2} \log n - \frac{n}{2}
  \]

- **Robust**

- **Easy to compare**

- **Euclid's Algorithm**: Greatest Common Divisor (gcd)

- **Given**: \( a, b \) nonnegative integers

- **Goal**: Find the (largest \( c \) such that \( c \) is a divisor of both \( a, b \))

  \[
  (\gcd(0, 0) = 0)
  \]

- \( \gcd(100, 30) \)

  \[
  \downarrow
  \]

- \( \gcd(30, 10) \)

  \[
  \downarrow
  \]

- \( \gcd(10, 0) = 10 \)
- Correctness:
  - Lemma: gcd always terminates

Hypothesis: if \( a + b \leq n \), then gcd terminates.

Base case: \( n = 0 \)
  \[ a = b = 0 \checkmark \]

Induction: assume hypothesis hold for \( n \), consider \( a + b = n + 1 \) (without loss of generality \( a \geq b \))

- if \( a = n + 1 \), \( b = 0 \), \( \checkmark \)
- otherwise \( b + (a \mod b) < a + b = n + 1 \)

by induction hypothesis, \( \text{gcd}(b, a \mod b) \) terminates

\[ \text{gcd}(a, b) \]

- if \( a < b \) then
  - swap \((a, b)\)
- if \( b = 0 \) then
  - return \( a \)
- else return
  \[ \text{gcd}(b, a \mod b) \]

- Lemma: \( \text{gcd}(a, b) \) is correct.

  if \( b = 0 \) then \( \text{gcd}(a, b) = a \)

  if \( b \neq 0 \) let \( a \mod b = a - kb \) \((k: \text{integer})\)

  if \( c \) is common divisor of \( a, b \),
  \[
  \frac{a \mod b}{c} = \frac{a - kb}{c} = \frac{a}{c} - k \frac{b}{c}
  \]

  integer

  \( \Rightarrow \) \( c \) is a c.d. of \( b, a \mod b \)

  if \( c \) is c.d. of \( b, a \mod b = a - kb \)

  \[
  \frac{a}{c} = \frac{(a - kb) + kb}{c} = \frac{a - kb}{c} + k \frac{b}{c}
  \]

  \( \Rightarrow \) \( c \) is a c.d. of \( a, b \)

  \( \Rightarrow \) \( \text{gcd}(a, b) = \text{gcd}(b, a \mod b) \) \( \checkmark \)