Fundamental Algorithm Techniques

Problem Set #1

Due: October 4, 2025

Problem 1 (Seems smart?). *Consider a sparse vector of size n with only 0's and a single 1:*

$$v = [0, 0, \ldots, 1, \ldots, 0]$$

(a) *Write pseudo recursive code that perform* **just** *the binary divide and conquer method without creating a copy of the vector and runs until vector fully decomposed into sizes 1.*

- *function divide(v , .., 2)*
- *function divide(v, .., m) for any $m \ll n$*

(b) *Analyse complexity of above divide and conquer: m=2, m=3,... tertiary division, which $T(n)$ and which $\mathcal{O}$?*

(c) *Next, once the division has reach sizes of 1, we collect the unique 1 and its position.*

- *Evaluate the cost $f(n)$*
- *What is the recurrence relation $T(n)$*
- *is complexity now $\mathcal{O}(log(n))$ or $\mathcal{O}(n)$ (use master Theorem)*

(d) *Compare with simpler approach: run over all indices...*

Problem 2 (School Multiplication was easy back then!). *Similar to Paesant multiplication, but in basis 10. it is just the first multiplication you learned at school.*

represent x, y in $\mathbb{N}^+$ with the vectors/arrays X and Y, such that
$x = \sum_{i=0}^{n_x} X[i] \cdot 10^i$, $y = \sum_{j=0}^{n_y} Y[j] \cdot 10^j$.

1. *write a multiplication pseudocode using above vector*
2. *tweak code such that results can be larger than the limit of your standard integer*
3. *what is the time complexity of your code?*
4. *Explain how above multiplication can be described with the divide and conquer recursion to find $T(n) \approx 4T(\frac{n}{2})$ or Karatsuba algorithm: $T(n) \approx 3T(\frac{n}{2})$, finding with Master Theorem resp. $\mathcal{O}(n^2)$ and $\mathcal{O}(n^{1.585})$* **hard**

   Hint 1: $x \cdot y = \left(x_1 \cdot 10^{n/2} + x_0\right)\left(y_1 \cdot 10^{n/2} + y_0\right)$
   Hint 2: $x \cdot y = z_2 \cdot 10^n + z_1 \cdot 10^{n/2} + z_0$
5. *$n!$ can be computed by one application of school multiplication: $n! = \frac{1}{2} mult(v, w)$ with which v, w?* **hard** *but simple math...*