Homework 2
Fundamental Algorithms, Fall 2020, Professor Yap

Due: Mon Oct 12, in GradeScope by 11pm.

INSTRUCTIONS:

- Remember that we have a "no late homework" policy. Special permission must be obtained in advance if you have a valid reason.
- Only hand in answers to the questions with positive points attached to them. Do the others at your leisure.

---

1. (8 Points) Carry out Karatsuba's algorithm for $X = 6 = (0110)_2$ and $Y = 11 = (1011)_2$. Draw the recursion tree in which each node has 3 binary values (X, Y, Z) where $Z = XY$. Be sure to write X, Y, Z in binary, not decimal.

   DETAILS: If $|X| \neq |Y|$, you should pad one of them so that $|X| = |Y| = n$. When we split the argument X into two, assume $|X_0| = \lceil n/2 \rceil$ and $|X_1| = \lfloor n/2 \rfloor$. The leaves of the recursion tree represent the product of two binary bits. Each internal node has 3 children corresponding to the recursive results Z_2, Z_1, Z_0 (in this order please).

2. (20 Points) Normally, our complexity bounds are Θ-order. In this question, we want to take into account the multiplicative constants that are hidden by the Θ-notations.

   (i) Argue that a more "honest" worst case recurrence for Karatsuba's algorithm should be

   $$T(n) = 3T(\lceil n/2 \rceil + 1) + 5n + O(1). \tag{1}$$

   Please justify all the constants $(1, 2, 3, 5)$ appearing in (1).

   NOTE: since we are interested in constants in (1), we must tell you the cost to add two n-bit numbers: the cost is exactly n. Also, the cost to compute Z from Z_0, Z_1, Z_2 is $2n$ (see ¶II.3, p. 7). But we don't really care about the $O(1)$ term in (1) (so we are still slightly abstract!).

   (ii) Henceforth, assume $T(n)$ eventually satisfies the recurrence (1) *without the $O(1)$ term*. We want to prove an upper bound $T(n)$ with explicit multiplicative constants. Consider a function of the form

   $$U(n) = (n+3)^{\lg 3} - Kn \tag{2}$$

   for some $K \geq 0$. Suppose $T(n) \leq U(n)$ (ev.). Determine the smallest possible value of the constant K. HINT: $\lceil n/2 \rceil + 1 \leq (n+3)/2$.

   (iii) *This part is open-ended.* Argue why the upper bound (2) is STILL unrealistic (not "honest"). How would do you suggest providing a realistic upper bound for $T(n)$?

3. (18 Points) Exercise II.6.4 (page 42): growth types.
   QUESTION:

   - Polynomial Sums

   $$\sum_{i \geq 1}^{n} i \log i = \Theta(n^2 \log n), \qquad \sum_{i \geq 1}^{n} \log i = \Theta(n \log n), \qquad \sum_{i \geq 1}^{n} i^a = \Theta(n^{a+1}) \ (a \geq 0). \tag{3}$$

   - Exponentially Increasing Sums

   $$\sum_{i \geq 1}^{n} b^i = \Theta(b^n) \ (b > 1), \qquad \sum_{i \geq 1}^{n} i^{-5} 2^{2^i} = \Theta(n^{-5} 2^{2^n}), \qquad \sum_{i \geq 1}^{n} i! = \Theta(n!) \quad . \tag{4}$$

- Exponentially Decreasing Sums

$$\sum_{i\geq 1}^{n} b^{-i} = \Theta(1)\ (b>1), \qquad \sum_{i\geq 1}^{n} i^2 i^{-i} = \Theta(1), \qquad \sum_{i\geq 1}^{n} i^{-i} = \Theta(1) \quad . \tag{5}$$

Verify that the examples in (3), (4) and (5) are, indeed, as claimed, polynomial type or exponential type. □

4. (16 Points) Exercise II.8.2 (p.47), solving two recurrences with Range and Domain transformations.

5. (15 Points) Consider the expression $E(n) := f(n)^{g(h(n))}$ where $\{f, g, h\} = \{2^n, 1/n, \lg n\}$. These are $6 = 3!$ possibilities for $E(x)$:

$E(n)$	f	g	h
E_1	2^n	$1/n$	$\lg n$
E_2	2^n	$\lg n$	$1/n$
E_3	$\lg n$	2^n	$1/n$
E_4	$\lg n$	$1/n$	2^n
E_5	$1/n$	2^n	$\lg n$
E_5	$1/n$	$\lg n$	$1/n$

Determine the domination relation between these functions.

6. (10 Points) Exercise II.13.4 (p.76). Ordering 5 functions, (a)-(e).

7. (30 Points) Consider the following recurrence

$$T(n) = d(n) + G(n,\ T(b_1(n)), T(b_2(n))). \tag{6}$$

The sequence (d, G, b_1, b_2) of functions in (6) is called the **scheme** of the recurrence. Let

$$SOL = SOL(d, G, b_1, b_2)$$

denote the set of complexity functions $T(n)$ that eventually satisfy the recurrence (6). The scheme is **regular** if the following properties are eventually satisfied non-vacuously:

- $d(n) \geq 1$ and non-decreasing. E.g., $d(n) = 1$ or $d(n) = \log n$.
- $G(n, x_1, x_2) > 0$ and is increasing in each component;
- $G(n, x_1, x_2)$ is **sublinear**: this means for all $C > 0$,

$$G(n, Cx_1, Cx_2) \leq C \cdot G(n, x_1, x_2).$$

  E.g., $G(n, x_1, x_2) = n^2 + c_1 x_1 + c_1 x_2$ where $c_i > 0$.
- Each b_i is **Archimedean** relative to a constant $\gamma > 0$: this means $0 < b_i(n) < n - \gamma$, and also $b_i \ggg 1$. E.g., $b_i(n) = n/c_i$ where $c_i > 1$, or $b_i(n) = n - c$ where $c > 0$.

Call $\gamma > 0$ the **gap constant** of the scheme. Another constant of the scheme is $n_0 = n_0(d, G, b_1, b_2)$ such that for all $n \geq n_0$ and $x_i \geq n_0$ $(i = 1, 2)$, the above properties are non-vacuously satisfied. Use real induction to prove the following properties:
(a) (Non-emptiness) $SOL(d, G, b_1, b_2)$ is non-empty. HINT: suffices to construct one solution.
(b) (Increasing) Each $T \in SOL(d, G, b_1, b_2)$ is eventually strictly increasing:

$$(\exists n_2)(\forall n, n')[(n_2 < n < n') \rightarrow T(n) < T(n')].$$

(c) (Θ-Robustness) Show that if $T, T' \in SOL(d, G, b_1, b_2)$ then $T = \Theta(T')$.