Homework 1
Fundamental Algorithms, Fall 2002, Professor Yap

Due: Tue September 24, in class

INSTRUCTIONS:

- Please read questions carefully.
- Note that we do not accept late homeworks because we would like to publish solutions in a timely fashion.
- We put a link from our homework page to the old homework from a previous class, including solutions. It is a good idea to read them, as this gives a clue to what is emphasized in this course.

---

1. (10 Points) Please carry out by hand (!) a simulation of Karatsuba's algorithm for the following input: $X = 11 = (1011)_2$, and $Y = 5 = (0101)_2$. Organize your computation in any way you find reasonable, as long as you explain clearly what you are doing.

   SOLUTION:

   Please go to the last page for the diagram illustrating the recursion using Karatsuba's algorithm. The Z or z at each node represents the value of the multiplication at that node. Also, the variables x_1, x_0, y_1, y_0 have been used to carry out the algorithm for $(X_1 + X_0)(Y_1 + Y_0)$. At the bottom level, the operands of the multiplication are simply represented by x, y.

2. (20 Points) Use the EGVS Method (i.e., Rote Method) to solve the following recurrences:
   (a) $T(n) = T(n-1) + n$
   (b) $T(n) = 4T(n/2) + 1$
   The method involves 4 steps (Expand, Guess, Verify, Stop). Make sure that each step is clearly marked and explained. Be sure to tell us what initial condition you choose.

   SOLUTION:

   (a)

$T(n)$	$=$	$T(n-1) + n$	Expand
	$=$	$T(n-2) + n - 1 + n$	Expand again
	$=$	$T(n-3) + n - 2 + n - 1 + n$	Expand again
	$\vdots$		
	$=$	$T(n-i) + \sum_{j=n-i}^{n} j$	Guess
	$=$	$T(n-(i+1)) + n - i + \sum_{j=n-i}^{n} j$	Now verify
	$=$	$T(n-(i+1)) + \sum_{j=n-(i+1)}^{n} j$	Verified
	$\ldots$		
	$=$	$T(0) + \sum_{j=1}^{n} j$	Choose i=n to stop
	$=$	$\frac{n(n+1)}{2}$	choosing $T(0) = 0$

Hence $T(n) = \Theta(n^2)$.

(b)

$$\begin{aligned}
T(n) &= 4T(\tfrac{n}{2}) + 1 && \text{Expand} \\
&= 4(4T(\tfrac{n}{2^2}) + 1) + 1 && \text{Expand again} \\
&\vdots \\
&= 4^jT(\tfrac{n}{2^j}) + \textstyle\sum_{i=0}^{j-1} 4^i && \text{Guess} \\
&= 4^j(4T(\tfrac{n}{2^{j+1}}) + 1) + \textstyle\sum_{i=0}^{j-1} 4^i && \text{Verify} \\
&= 4^{j+1}T(\tfrac{n}{2^{j+1}}) + \textstyle\sum_{i=0}^{j} 4^i \\
&\vdots \\
&= 4^kT(\tfrac{n}{2^k}) + \textstyle\sum_{i=0}^{k-1} 4^i && \text{Choose } k = \lg n \text{ to stop} \\
&= \tfrac{4^k-1}{4-1}\text{Choose } T(\tfrac{n}{2^k}) = 0
\end{aligned}$$

Hence $T(n) = \Theta(n^2)$.

3. (2 Points) Question 3.1-3, page 50.

   Explain why the statement "The running time of algorithm A is at least $O(n^2)$ is meaningless".

   SOLUTION:

   $f(n) = O(n^2)$ implies that $f(n)$ is bounded above by cn^2 for some $c > 0$ and for all $n \geq n_0$. It's an upper bound on $f(n)$. So, saying "The running time of algorithm A is at least $O(n^2)$" is meaningless.

4. (4 Points) Question 3.1-4, page 50.

   Is $2^{n+1} = O(2^n)$?

   SOLUTION:

   $2^{n+1} = 2 \times 2^n$. Taking $c = 2, n_0 = 1$, we get $2^{n+1} \leq c2^n$ for all $n \geq n_0$. Hence $2^{n+1} = O(2^n)$.

   Is $2^{2n} = O(2^n)$?

   SOLUTION:

   Suppose $2^{2n} = O(2^n)$. Then for some $c > 0$ and for all $n \geq n_0$, we have, $2^{2n} \leq c2^n$ for all $n \geq n_0$. Then $2^n \leq c$ for all $n \geq n_0$. But c is a constant and $2^n \to \infty$, so we get a contradiction.

5. (10 Points) Show that $\sum_{i=1}^{n} \log i = \Theta(n \log n)$.

   NOTE: You must divide this argument into two steps (show an upper bound, then a lower bound). You MUST explicitly show the constants c_1, c_2 and n_0 in the definition of the Θ-notation.

   SOLUTION:

   To show that $\sum_{i=1}^{n} \lg i = \Theta(n \lg n)$.

   We now show the upper bound i.e., $\sum_{i=1}^{n} \lg i = O(n \lg n)$.

We have $\sum_{i=1}^{n} \lg i \leq \sum_{i=1}^{n} \lg n = n \lg n$. Therefore, we may choose, $c_2 = 1$ and n_0 is 1.

To prove the lower bound, we note that

$$\begin{aligned}
\sum_{i=1}^{n} \lg i &= \lg 1 + \lg 2 + \cdots + \lg \lfloor \tfrac{n}{2} \rfloor + \underbrace{\lg \left\lceil \frac{n}{2} \right\rceil + \cdots + \lg n}_{\lceil \frac{n}{2} \rceil \text{ terms}} \\
&\geq \lg \lceil \tfrac{n}{2} \rceil + \cdots + \lg n \\
&\geq \lceil \tfrac{n}{2} \rceil \lg \lceil \tfrac{n}{2} \rceil \\
&\geq \tfrac{n}{2} \lg \tfrac{n}{2}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{i=1}^{n} \lg i &\geq \tfrac{n}{2} \lg \tfrac{n}{2} \\
&= \tfrac{n}{2} \lg n - \tfrac{n}{2}
\end{aligned}$$

We note that the choice $c_1 = \frac{1}{2}$ and $n_0 \geq 2$ satisfies the equation $c_1 n \lg n \leq \sum_{i=1}^{n} \lg i$.

6. (10 Points) Question 3-3, page 58. Do part (a) only.

To help us grade your answer, we want you write each equivalence class in one line of answer. Also, begin with the slowest growing functions first.

SOLUTION:

We use the following useful notation that

$$\begin{array}{lll}
f \asymp g & \text{if} & f = \theta(g) \\
f \preceq g & \text{if} & f = O(g) \\
f \prec g & \text{if} & f = O(g) \text{ but } g \neq O(f)
\end{array}$$

Please use this notation for future assignments. The ordering of the given functions is as follows:

- $1 \asymp n^{\frac{1}{\lg n}}$
- $\lg \lg^* n \preceq \lg^* \lg n \asymp \lg^* n \preceq 2^{\lg^* n}$
- $\ln \ln n \prec (\lg n)^{\frac{1}{2}} \prec \ln n \prec \lg^2 n$
- $(\sqrt{2})^{\lg n} \asymp n^{\frac{1}{2}} \prec n \asymp 2^{\lg n} \prec \lg n! \asymp n \lg n \prec n^2 \asymp 4^{\lg n} \prec n^3$
- $(\lg n)! \asymp n^{\lg \lg n} \asymp (\lg n)^(\lg n) \prec (\frac{3}{2})^n \prec 2^n \prec e^n \prec n2^n \prec (n!) \prec (n+1)! \prec 2^{2^n} \prec 2^{2^{n+1}}$

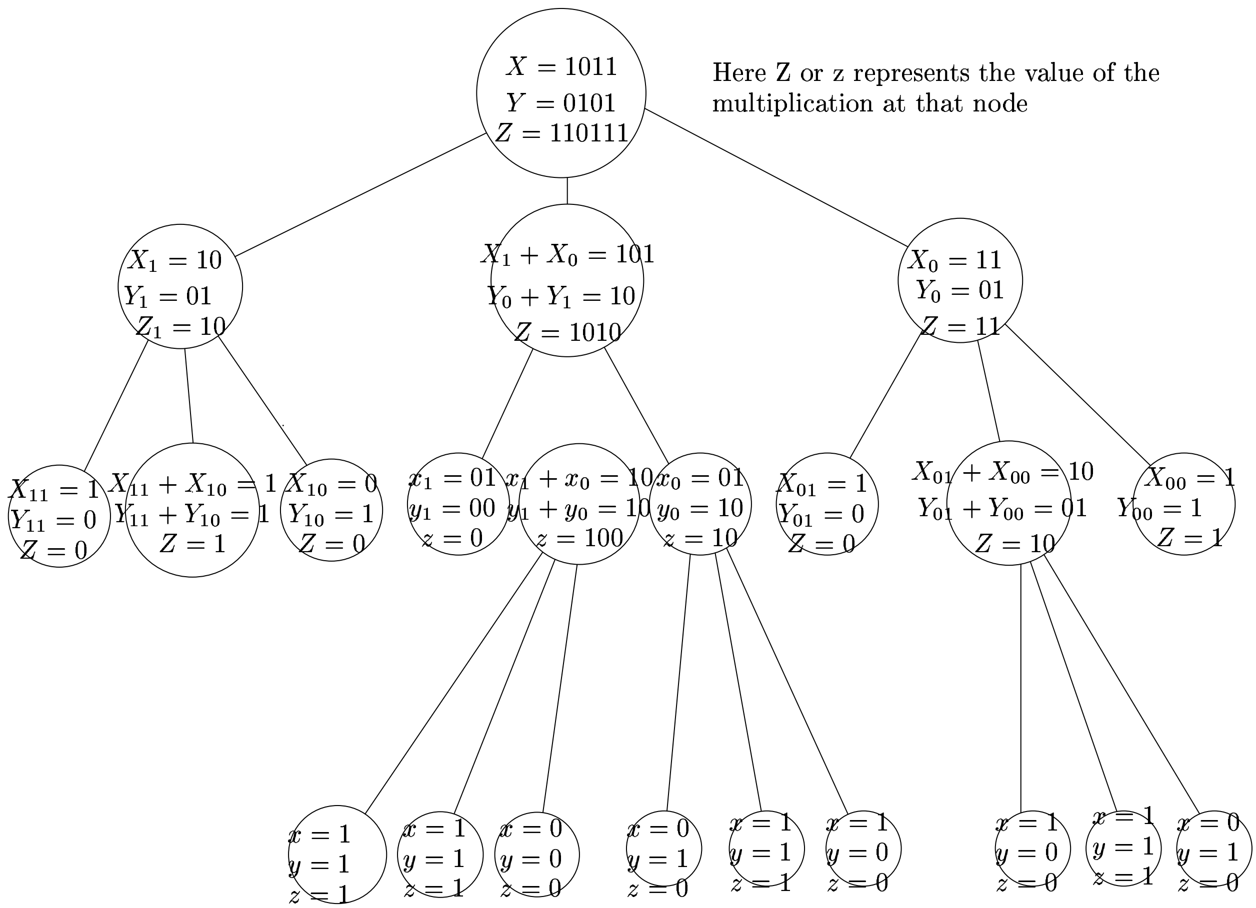

Figure 1: Recursion Tree for Karatsuba