CS 3510 Homework 1

Wenqi He

September 6, 2017

1

(a)

By definition of big-O notation, if $f(n) \leq \mathcal{O}(g(n))$ and $g(n) \leq \mathcal{O}(h(n))$, then there exists some constants c_1 and c_2 such that for large enough n

$$f(n) \leq c_1 g(n)$$

$$g(n) \leq c_2 h(n)$$

Combining the two inequalities,

$$f(n) \leq c_1 g(n) \leq c_1 c_2 h(n)$$

In other words, there exists $c = c_1 c_2$ such that for large enough n

$$f(n) \leq ch(n),$$

therefore,

$$f(n) \leq \mathcal{O}(h(n))$$

(b)

The innermost loop is linear. The two outer loops are similar to the loops in insertion sort, but since i and j can only take $\Theta(\log(n))$ values, so the time complexity is $\Theta^2(\log(n))$. The time complexity of the entire nested loops is therefore

$$\Theta(\log^2(n)) \cdot \Theta(n) = \Theta(n \log^2(n))$$

(c)

The statement is false.
Let $g(n) = c \cdot f(n)$, then for all n,

$$g(n) \leq cf(n) \text{ and } g(n) \geq cf(n).$$

By definition of big-O notations, $f(n) \leq \mathcal{O}(g(n))$ and $f(n) \geq \Omega(g(n))$, which is an counterexample. Q.E.D.

(d)

The statement is true.
Since $\log^a n \le \mathcal{O}(n^b)$ for any positive a and b,

$$\log^{10} n \le \mathcal{O}(n^{0.1}),$$

$$\begin{aligned} n^2 \log^{10} n &\le \mathcal{O}(n^2 \log^{10} n) \\ &= \mathcal{O}(n^2) \cdot \mathcal{O}(\log^{10} n) \\ &\le \mathcal{O}(n^2) \cdot \mathcal{O}(n^{0.1}) \\ &= \mathcal{O}(n^{2.1}) \end{aligned}$$

Q.E.D.

(e)

The statement is false.
$\forall c \in (0, \infty)$, there exists a N large enough such that $2^N = c$, then $\forall n > N$,

$$2^{2n} = 2^n \cdot 2^n > 2^N \cdot 2^n = c \cdot 2^n$$

In other words, $2^{2n} = 2^n \cdot 2^n$ always exceeds $c \cdot 2^n$ as n goes to infinity no matter how large c is. Q.E.D.

2

(a)

$$a = 3, \quad b = 4, \quad d = 1$$

By Master Theorem, since $\log_4(3) < 1$,

$$T(n) \leq \mathcal{O}(n)$$

(b)

$$a = 8, \quad b = 4, \quad d = 1.5$$

By Master Theorem, since $\log_4(8) = 1.5$,

$$T(n) \leq \mathcal{O}(n^{1.5} \log n)$$

(c)

The recurrence relation is

$$T(n) = 3T\left(\frac{n}{4}\right) + \mathcal{O}(1)$$

$$a = 3, \quad b = 4, \quad d = 0$$

By Master Theorem, since $\log_4(3) > 0$,

$$T(n) \leq \mathcal{O}(n^{\log_4(3)})$$

(d)

The recurrence relation is

$$T(n) = 2^n T\left(\frac{n}{2}\right) + \mathcal{O}(n),$$

which is not in the form

$$T(n) = aT\left(\frac{n}{b}\right) + \mathcal{O}(n^d),$$

and therefore Master Theorem does not apply.

3

(a)

$$T(n) = 3T\left(\frac{3n}{5}\right) + \mathcal{O}(n)$$
$$= 3T\left(\frac{n}{5/3}\right) + \mathcal{O}(n)$$
$$a = 3, b = 5/3, d = 1$$

Since

$$3 > \frac{5}{3},$$
$$\log_{\frac{5}{3}}(3) > 1$$

By Master Theorem,

$$T(n) \leq \mathcal{O}(n^{\log_{\frac{5}{3}}(3)})$$

(b)

For example:

$$1, 4, 5, 2, 3$$

After the first sort,

$$1, 4, 5, 2, 3$$

After the second sort,

$$1, 4, 2, 3, 5$$

After the last sort,

$$1, 2, 4, 3, 5$$

The sequence is not sorted when the algorithm terminates.

4

(a)

The coefficient of the i th term is

$$\begin{aligned}
c_i &= \sum_{\max\{0,i-n\}\le j\le \min\{i,n\}} a_j \cdot b_{i-j} \\
&\le \sum_{\max\{0,i-n\}\le j\le \min\{i,n\}} n^2 \\
&= \big(\min\{i,n\} - \max\{0,i-n\}\big)\cdot n^2
\end{aligned}$$

Since $\min\{i,n\} \le n$ and $\max\{0, i-n\} \ge 0$

$$\min(i,n) - \max(0, i-n) \le n$$

Therefore,

$$c_i \le n \cdot n^2 = n^3$$

By definition of big-O notation,

$$c_i \le \mathcal{O}(n^3)$$

Because all coefficients of $p(x)$ and $q(x)$ are non-negative, the sum must also be non-negative, therefore

$$c_i \in [0, \mathcal{O}(n^3)]$$

(b)

A polynomial of degree n (suppose n is even) can be split into two halves

$$\begin{aligned}
p(x) &= (a_0 + \cdots + a_{\frac{n}{2}-1}x^{\frac{n}{2}-1}) + (a_{\frac{n}{2}}x^{\frac{n}{2}} + \cdots + a_n x^n) \\
&= (a_0 + \cdots + a_{\frac{n}{2}-1}x^{\frac{n}{2}-1}) + (a_{\frac{n}{2}} + \cdots + a_n x^{\frac{n}{2}})x^{\frac{n}{2}} \\
&= (a_0 + \cdots + a_{\frac{n}{2}-1}x^{\frac{n}{2}-1} + 0\cdot x^{\frac{n}{2}}) + (a_{\frac{n}{2}} + \cdots + a_n x^{\frac{n}{2}})x^{\frac{n}{2}} \\
&= p_0(x) + p_1(x)x^{\frac{n}{2}}
\end{aligned}$$

where $p_0(x), p_1(x)$ are both degree $\frac{n}{2}$ polynomials.
By analogy with integer multiplication,

$$\begin{aligned}
p(x)\cdot q(x) &= (p_0(x) + p_1(x)x^{\frac{n}{2}})(q_0(x) + q_1(x)x^{\frac{n}{2}}) \\
&= [p_0(x)q_0(x)] + [p_0(x)q_1(x) + p_1(x)q_0)]\cdot x^{\frac{n}{2}} + [p_1(x)q_1(x)]\cdot x^n
\end{aligned}$$

To compute $p(x)\cdot q(x)$, first compute:

1. $z_0 = p_0(x)q_0(x)$
2. $z_1 = p_1(x)q_1(x)$

3. $z_2 = [p_0(x) + p_1(x)] \cdot [q_0(x) + q_1(x)]$

Then
$$p(x)q(x) = z_0 + (z_2 - z_0 - z_1) \cdot x^{\frac{n}{2}} + z_1 \cdot x^n$$

Assuming integer arithmetics are $\mathcal{O}(1)$ operations, then the addition in step 3 take $\mathcal{O}(n)$ time. The last step produces a polynomial of degree $2n$, so the addition also takes $\mathcal{O}(n)$ time. The runtime recurrence for this algorithm is

$$T(n) = 3T\left(\frac{n}{2}\right) + \mathcal{O}(n)$$

Since $\log_2(3) \approx 1.58 > 1$, by Master Theorem,

$$T(n) \leq \mathcal{O}(n^{\log_2(3)}) \leq \mathcal{O}(n^{1.6})$$

Q.E.D.

(c)

The coefficient of the x^{n-s} term is

$$\begin{aligned}
&\sum_{\max\{0,-s\} \leq j \leq \min\{n,n-s\}} a_j \cdot b_{n-s-j} \\
&= \sum_{0 \leq j \leq n-s} y_j \cdot z_{n-(n-s-j)} \\
&= \sum_{j=0}^{n-s} y_j \cdot z_{s+j}
\end{aligned}$$

Q.E.D.