CS 260 - Fundamentals of the Design and Analysis of Algorithms **Fall 2016**

HomeWork 2

September 27, 2016 *Liangjian Chen*

1. (Problem 2)

   $10^{10} * 60 * 60 = 3.6 * 10^{13}$

   (a) $n = \sqrt{3.6 * 10^{13}} = 6 * 10^6$

   (b) $n = \sqrt[3]{3.6 * 10^{13}} = 33019$

   (c) $n = 6 * 10^5$

   (d) $n \approx 1.4 * 10^{11}$(when base = e)

   (e) $n = 45$

   (f) $n = 5$

2. (Problem 4)

   order is $g_1, g_5, g_3, g_4, g_2, g_7, g_6$.

3. (Problem 5)

   (a) False, $f(n) = 10, g(n) = 1$

   (b) False, $f(n) = n^2 + n, g(n) = n^2$

   (c) True,
   $\forall n \geq n_0, f(n) \leq c * g(n)$
   $\forall n \geq n_0, f(n)^2 \leq c^2 * g(n)^2$
   $f(n)^2 = O(g(n)^2)$

4. (Problem 6) $f(n)$ can be calculated directly.

$$
\begin{aligned}
f(n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} (j - i + 1) \\
&= \frac{1}{2} \sum_{i=1}^{n} (n - i + 2) * (n - i + 1) \\
&= \frac{1}{2} \sum_{i=1}^{n} i(i + 1) \\
&= \frac{n(n+1)(2n+1)}{12} + \frac{n(n+1)}{4} \\
&= \frac{n(n+1)(n+2)}{6}
\end{aligned}
$$

   (a) $O(f(n)) = n^3$

(b) $\Omega(f(n)) = n^3$

(c) Original algorithm makes a lot of redundant work. Instead of iterating from $a[i]$ to $a[j]$, $B[i, j]$ could be simply transfered from either $B[i, j-1]$ or $B[i, j+1]$.

Algorithm 1 Improved Algorithm:

```
1: for i = 1, 2, ..., n do
2:     B[i, i] ← A[i]
3:     for j = i − 1, i − 2, ..., 1 do
4:         B[i, j] ← B[i, j + 1] + A[j]
5:     end for
6:     for j = i + 1, i + 2, ..., n do
7:         B[i, j] ← B[i, j − 1] + A[j]
8:     end for
9: end for
```

In this algorithm,

$$\begin{aligned} g(n) &= \sum_{i=1}^{n} (1 + \sum_{j=1}^{i-1} 1 + \sum_{j=i+1}^{n} 1) \\ &= \sum_{i=1}^{n} n \\ &= \frac{n(n+1)}{2} \\ &= \Theta(n^2) \end{aligned}$$