Introduction to Algorithm Homework 1

Mou Zhang

February 3rd, 2020

1 Problem 1

1.1 Prove that, if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = k$, where k is a positive constant, then $f(n) = \Theta(g(n))$.

Proof. By the definition of limit, when n becomes large enough, $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ will equal to k or k_0 which nearly reaches k. Then when $n \geq n_0$ for a large enough number n_0, $f(n) \leq k_0 \cdot g(n)$. Thus $f(n) = O(g(n))$. Similarly, by the definition of limit, there exists a $k_1 < k$ such that when $n \geq n_1$ for a large enough number n_1, $f(n) \geq k_1 \cdot g(n)$. Thus $f(n) = \Omega(g(n))$. Sums up, $f(n) = \Theta(g(n))$.

1.2

1 $\frac{n^2}{\log n} = \Theta(n)$.
False
Since $\lim_{n\to\infty} \frac{\frac{n^2}{logn}}{n} = \lim_{n\to\infty} \frac{n}{logn} = \infty$, we know that there exists a positive constant C and some large n_0, such that $\frac{n^2}{logn} > C \cdot n$ for $n \geq n_0$. Thus $\frac{n^2}{logn} = \omega(n)$ and $\frac{n^2}{logn} \neq O(n)$.

2 $2^n = O(3^n)$.
True
Since $\lim_{n\to\infty} \frac{2^n}{3^n} = \lim_{n\to\infty} \frac{2}{3}^n = 0$, we know that for any n, $2^n < C \cdot 3^n$ for $n \geq n_0$. Thus $2^n = O(3^n)$.

3 $\sqrt{n} = \Theta(2 \log n^2)$.
False
Since $\lim_{n\to\infty} \frac{\sqrt{n}}{2 \log n^2} = \lim_{n\to\infty} \frac{\sqrt{n}}{logn} = \infty$, we know that there exists a positive constant C and some large n_0, such that $\sqrt{n} > C \cdot 2 \log n^2$ for $n \geq n_0$. Thus $\sqrt{n} = \omega(2 \log n^2)$ and $\sqrt{n} \neq O(2 \log n^2)$.

4 $3n \log n + n = O(\frac{n^2-n}{2})$.
True
Since $\lim_{n\to\infty} \frac{3n \log n+n}{\frac{n^2-n}{2}} = \lim_{n\to\infty} \frac{n\ logn}{n^2} = \frac{\log n}{n} = 0$, we know that there exists a positive constant C and n_0, such that $3n \log n + n \leq C \cdot \frac{n^2-n}{2}$ for $n \geq n_0$. Thus $3n \log n + n = O(\frac{n^2-n}{2})$.

5 Let f and g be positive functions. If $f(n) + g(n) = \Omega(f(n))$ then $g(n) = O((f(n))^2)$.
False
Since $f(n) + g(n) = \Omega(f(n))$, $g(n)$ is lower bounded by $f(n)$ but g(n) has no upper bound. Let $g(n) = f(n)^3$, then $g(n)$ meets the above condition but $g(n) \neq O((f(n)^2)$

1.3 Prove that

$$\sum_{i=1}^{n} \log i = \Theta(n \log n).$$

Proof. Since $\log i$ is an increasing function, which means $\log n > \log i$ for any $i < n$. Therefore, $\sum_{i=1}^{n} \log i \leq n \cdot \log n$. Thus, $\sum_{i=1}^{n} \log i = O(n \log n)$.
Besides, $\log 1 + ... + \log \frac{n}{2} + ... + \log n \geq \log \frac{n}{2} + ... + \log n \geq \log \frac{n}{2} + ... + \log \frac{n}{2} = \frac{n}{2} \cdot \log \frac{n}{2} = \Omega(n \log n)$.
Thus $\sum_{i=1}^{n} \log i = \Omega(n \log n)$.
Therefore, $\sum_{i=1}^{n} \log i = \Theta(n \log n)$.

2 Problem 2

2.1 1 . *Proof.*

I. When n = 1, $\frac{1}{2} = \frac{1}{2}$. True.

II. Assume when $n = k$, we have $\sum_{i=1}^{k} \frac{1}{i(i+1)} = \frac{k}{k+1}$. We need to show that when $n = k + 1$, $\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \frac{k+1}{k+2}$.

So we have $\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \frac{k}{k+1} + \frac{1}{(k+1)(k+2)} = \frac{k(k+2)+1}{(k+1)(k+2)} = \frac{(k+1)^2}{(k+1)(k+2)} = \frac{k+1}{k+2}$. True.

2 . There are $\binom{10}{3} = \frac{10\times 9\times 8}{3\times 2\times 1} = 120$ ways to distribute.

3 . We totally have n^n ways to distribute all n books. And to make sure there no empty shelves when we are placing all n books, we have to make sure that every book is placed at a different shelve from all books that are placed before. Therefore, there are $n \cdot (n-1) \cdot (n-2) ... \cdot 1 = n!$ ways to make no empty selves. Thus, the probability is $\frac{n!}{n^n}$.

2.2 .

Step 1: When $k > 2$: Now we have 2^k disks($k > 2$). We split them equally into 4 groups g1,g2,g3,g4. There are 2^{k-2} disks in each group.

Step 2: We test g1 and g2. If their weight are equal, then they are both good disks. We can join g3 and g4 into a new group and return Step1 until k = 2. If the weight of g1 and g2 are not equal, than there's a bad disk in g1 and g2. So we can join g1 and g2 into a new group and return to Step1. Therefore, every time we compare the weight of two groups, we half the problem by finding out half good disks. Finally we got $2^2 = 4$ disks left and we use the equivalence tester $k - 2$ times.

Step 3: When $k = 2$: we sign them as g1,g2,g3,g4. First we test g1 and g2. If their weights are equal, g1 and g2 are both good. Then we test g1 and g3, if their weights are not equal, g3 is the bad one; else g4 is the bad one. This takes 2 steps. If the weights of g1 and g2 are not equal, on of them is bad. Then we test g1 and g3, if their weights are not equal, g1 is the bad disk; else g2 is the bad disk. This also takes 2 steps. Therefore ,when $k = 2$, we need only 2 step to find the bad disk.

To sum up, we have found the bad disk in $(k - 2) + 2 = k$ steps.