Algorithm Hw 1 93327108 Chih-Hsiang Wang

1. if $\log_b n \le c_1 \cdot \log_2 n$, where $b>1, c_1>0 \Rightarrow \frac{\log_2 n}{\log_2 b} \le c_1 \cdot \log_2 n$. $c_1 \ge \frac{1}{\log_2 b}$. by definition $\Rightarrow \log_b n = O(\log_2 n)$

if $\log_b n \ge c_2 \cdot \log_2 n$, where $b>1, c_2>0 \Rightarrow \frac{\log_2 n}{\log_2 b} \ge c_2 \cdot \log_2 n$. $c_2 \le \frac{1}{\log_2 b}$. by definition $\Rightarrow \log_b n = \Omega(\log_2 n)$

because $\log_b n = O(\log_2 n)$ and $\log_b n = \Omega(\log_2 n) \Rightarrow \log_b n = \Theta(\log_2 n)$

2. if $\sum_{i=1}^{n} i^{15} \le c_1 \cdot n^{16}$, where $c_1>0 \Rightarrow c_1 \ge \frac{\sum_{i=1}^{n} i^{15}}{n^{16}}$, let $c_1 \ge \frac{n \cdot n^{15}}{n^{16}} = 1 \ge \frac{\sum_{i=1}^{n} i^{15}}{n^{16}}$. by definition $\Rightarrow \sum_{i=1}^{n} i^{15} = O(n^{16})$

$\Theta(n^{16}) = (\frac{n}{2})^{16}$. if $\sum_{i=1}^{n} i^{15} \ge c_2 \cdot (\frac{n}{2})^{16}$ $(c_2>0)$ $\Rightarrow 1^{15}+2^{15}+\dots(\frac{n}{2})^{15}+\dots+n^{15} \ge (\frac{n}{2})^{15}+\dots+n^{15} \ge (\frac{n}{2})^{15}+\dots+(\frac{n}{2})^{15} = \frac{n}{2}\cdot(\frac{n}{2})^{15} = (\frac{n}{2})^{16}$,

by definition $\Rightarrow \sum_{i=1}^{n} i^{15} = \Omega(n^{16})$

because $\sum_{i=1}^{n} i^{15} = O(n^{16})$ and $\sum_{i=1}^{n} i^{15} = \Omega(n^{16}) \Rightarrow \sum_{i=1}^{n} i^{15} = \Theta(n^{16})$

3.

$n! = 1\times 2\times 3\times\dots\times n \le c_1 \cdot n^n$ for $c_1>0 \Rightarrow n! = O(n^n)$

if $n^n = n\times n\times\dots\times n \le c_2\cdot n!$, where $c_2>0$, we cannot find such $c_2 \Rightarrow n^n \ne O(n!) \Rightarrow n^n \ne \Theta(n!)$

4. $f_n = \begin{cases} 0 & \text{if } n=0 \\ 1 & \text{if } n=1 \\ f_{n-1}+f_{n-2} & \text{if } n>1 \end{cases}$

we can say f_n grows exponentially if we find $r>1$ that $f_n \ge r^n$ for all n

clarm: let $r = \frac{1+\sqrt{5}}{2} \approx 1.62 \Rightarrow r^2 = r+1$. assume $f_n \ge r^{n-2} \Rightarrow$ need to prove $f_n \ge r^{n-2}, \forall n \ge 1$

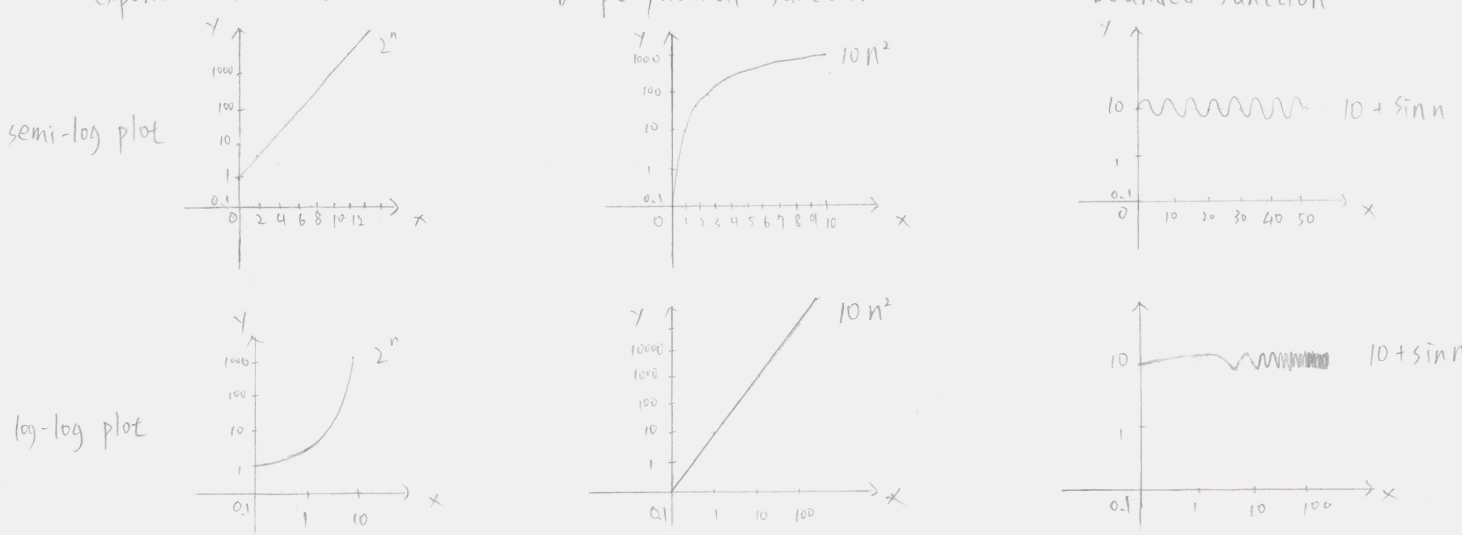
$n=1$: $f_1 = 1 \ge r^{1-2} = 1.62^{-1}$, $n=2$: $f_2 = 1 \ge r^{2-2} = 1.62^0 = 1$

$\Rightarrow$ assume $f_1, f_2, \dots f_n$ are all true, need to show $f_{n+1} \ge r^{-1}$, by induction method

$\Rightarrow f_{n+1} = f_n + f_{n-1} \Rightarrow f_{n+1} \ge r^{n-2} + r^{n-3} \Rightarrow f_{n+1} \ge r^{n-3}(r+1) \underset{\text{by clarm}}{\Longrightarrow} f_{n+1} \ge r^{n-3}\cdot r^2 \Rightarrow f_{n+1} \ge r^{-1} \Rightarrow$ proved

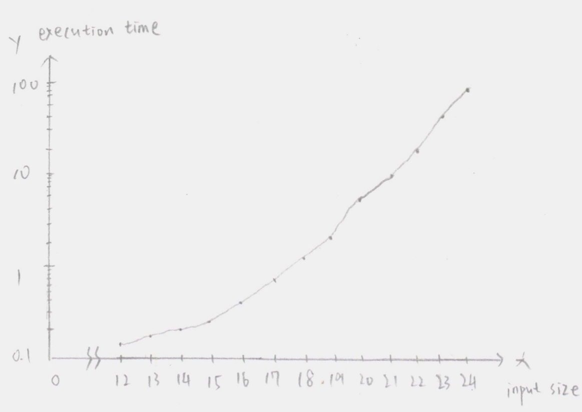
5. in every cases, $n \ge 0$

(a) exponential function (b) polynomial function (c) bounded function



$\Rightarrow$ exponential function gives straight line in semi-log plot, polynomial function gives straight line in log-log plot

6.



The best kind of graph for this kind of data is semi-log plot, because most time of datas is twice the amount of prior one. We can use semi-log plot to accomodate the datas while input size grows larger.

The complexity of this function is exponential because execution time approximately doubles when input size adds one.

7. $Hyp(B) = F(A,B)$ correctly moves the round up number of B from B to A

$Hyp(0)$, move 0 from B to A $\Rightarrow$ true

$Hyp(1)$, move 1 from B to A $\Rightarrow$ true

if $m > 2$, $Hyp(m)$, by $Hyp(m-1)$, move the round up number of $(m-1)$ from B to A

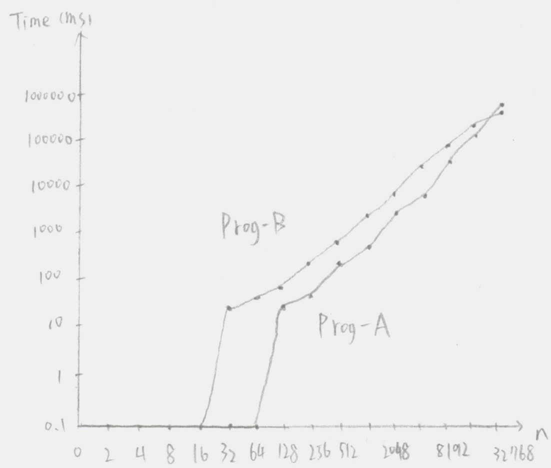
$\Rightarrow Hyp(m)$, move the round up number of m from B to A $\Rightarrow$ true

8. As n grows larger than 12, $T(n)/T(n-1)$ is approximately 2.0, so we can assume that the run time has the form $\Theta(2^n)$.

By the assumption, we can predict the run time for $n = 30$ is 301652×2^{13}

$\Rightarrow 2471133184$

9.



because run time is $\Theta(n^k) \Rightarrow \frac{T(m)}{T(m/2)} = \frac{c \cdot n^k}{c \cdot (m/2)^k} = 2^k$

$2^{k_A} \doteq 4$, $k_A = 2$

$2^{k_B} \doteq 3$, $k_B \doteq 1.73$

Because run times of Prog-A is $\Theta(n^2)$,

run times of Prog-B is $\Theta(n^{1.73})$

Prog-B is expected to be asymptotically faster