# Problem Set 1

**Name:** Kim Heesuk

## Problem 1-1.

**(a)** $f_1 = \log n^n = n \log n \in O(f_2)$, $f_3 = \log n^{6006} = 6006 \log n \in \Theta(\log n)$. So that $f_3 \in O(f_1)$. Meanwhile $f_4 \in O(f_2)$, and $f_5 \in O(f_3)$, the answer is $(f_5, f_3, f_1, f_4, f_2)$.

**(b)** $f_2 \in O(f_5)$ and $f_5 \in O(f_4)$. Also $f_1 \in O(f_2)$, $f_5 \in O(f_3)$. For $f_3$ and $f_4$,

$$\begin{aligned} \log \frac{f_3}{f_4} &= \log \frac{2^{6006^n}}{6006^{2^n}} \\ &= 6006^n \log 2 - 2^n \log 6006 \\ &= 2^n(3003^n \log 2 - \log 6006) \to \infty \end{aligned} \tag{1}$$

$f_4 \in O(f_3)$ and the answer is $(f_1, f_2, f_5, f_4, f_3)$.

**(c)** Firstly,

$$\begin{aligned} f_2 = \binom{n}{n-6} &= \frac{n!}{6!(n-6)!} \\ &\in \Theta\left(n(n-1)\dots(n-)\right) \\ &= \Theta(n^6), \end{aligned} \tag{2}$$

so $f_5 = n^6 \in \Theta(f_2)$. For $f_4$, by the Stirling's approximation,

$$\begin{aligned} \log f_4 = \log \binom{n}{n/6} &= \log \frac{n!}{(n/6)!(5n/6)!} \\ &\sim \log \frac{\sqrt{2\pi n}(n/e)^n}{\sqrt{2\pi n}(n/6e)^{n/6}\sqrt{2\pi n}(5n/6e)^{5n/6}} \\ &= \log \frac{(6/5^{5/6})^n}{\sqrt{2\pi n}} \\ &\in \Theta\left(n \log(6/5^{5/6}) - \frac{1}{2}(\log n + \log 2\pi)\right) = \Theta(n). \end{aligned} \tag{3}$$

since $6/5^{5/6} > 1$. Hence $f_2(\in \Theta(f_5)) \in O(f_4)$, $f_4 \in O(f_1)$ since $f_4 \in \{2^p | p \in \Theta(n)\} \subset \Omega(n^6)$ and $\log f_1 \in \Theta(n \log n)$. Also by the Stirling's approximation,

$$\begin{aligned} f_3 = (6n)! &\sim \sqrt{12\pi n}(6n/e)^{6n} \\ &\in \Theta((6n)^{6n}) \subset \Omega(n^n) = \Omega(f_1). \end{aligned} \tag{4}$$

Thus the answer is $(\{f_2, f_5\}, f_4, f_1, f_3)$.

**(d)** Using the Stirling's approximation, $f_1 \sim n^{n+4} + \sqrt{2\pi n}(n/e)^n \in \Theta(n^{n+4})$. Also considering $f_5/n^{12} = n^{1/n} \to 1$, $f_5 = n^{12+1/n} \sim n^{12} \in \Theta(n^{12})$. It is obvious that $f_2 = n^{7\sqrt{n}} \in O(n^{n+4}) = O(f_1)$ and $f_3 = 4^{3n\log n} \in O(7^{n^2}) = O(f_4)$. Finally for $f_3$ and $n^{n+4} \in \Omega(f_1)$,

$$\begin{aligned} \log \frac{f_3}{n^{n+4}} &= \log \frac{4^{3n\log n}}{n^{n+4}} \\ &= 3n\log n \cdot \log 4 - (n+4)\log n \\ &= ((3\log 4 - 1)n - 4)\log n \to \infty, \end{aligned} \tag{5}$$

$f_3 \in O(n^{n+4}) = O(f_1)$. The order should be $(f_5, f_2, f_1, f_3, f_4)$.

**Problem 1-2.**

**(a)** $T(k) = O(\log n) + T(k-1), T(k) = O(k \log n)$

```
def reverse(D, i, k): # T(k)
  if k < 2:
    return
  D.insert_at(i + k - 1, D.delete_at(i))      # O(log(n))
  reverse(D, i, k - 1)                          # T(k - 1)
```

If we call `reverse(D, i, 1)`, it returns and it is correct. Assume `reverse(D, i, k)` works right, `reverse(D, i, k + 1)` should work right because it deletes element of $i$'th index, and insert it to $i+k$ (next to the pre-deletion index $i+k$) and reverses $k$ items starting at index $i$ of `D`, which resulting reverse $k+1$ items starting at index $i$.

**(b)** $T(k) = O(\log n) + O(\log n) + T(k-1), T(k) = O(k \log n)$

```
def move(D, i, k, j):          #T(k)
  if k < 1:
    return
  temp = D.delete_at(i)        # O(log(n))
  if j < i:
    D.insert_at(j + 1, temp)   # O(log(n))
  else:
    D.insert_at(j, temp)       # O(log(n))
  move(D, i, k - 1, j)         # T(k-1)
```

Note that when $j \geq i + k$ (which can be just $\neg(j < i)$ since $i \leq j < i + k$ is false), `D.delete_at(i)` makes the index of element with pre-deletion index $j$ be $j - 1$.

**Problem 1-3.**

**Problem 1-4.**

**(a)**

**(b)**

**(c)**

**(d)** Submit your implementation to `alg.mit.edu`.