Algorithm Design and Analysis
Assignment 1
Wu Jiabao
Deadline: Oct 17, 2024

1. (a) 1. $f = O(g)$. Since $g(n) = n^{\log_2 5}$, $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0 < \infty$
   2. $f = \Theta(g)$. Since $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 100 \in (0, \infty)$.
   3. $f = \Omega(g)$. Since $f(n) = e^{\log n \log\log n}, g(n) = e^{\log n - \log\log n}$, $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \infty > 0$.
   4. $f = O(g)$. Since $f(n) = e^{\log n \log\log n}, g(n) = e^{\log n \log_2 n}$, $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0 < \infty$.
   5. $f = O(g)$. Since $f(n) = \Sigma_{i=1}^n i^k \leq n * n^k = g(n)$ for any $n > n_0 = 1$, $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} \leq 1 < \infty$.

   (b) Obviously:

$$\frac{f(n)}{g(n)} = \begin{cases} n+1, & \text{n is odd,} \\ \frac{1}{n+1}, & \text{n is even} \end{cases}.$$

   Suppose $f = O(g)$ is true. Then there exists $n_0 \in \mathbb{Z}^+, c \in \mathbb{R}^+$ such that $\frac{f(n)}{g(n)} \leq c$ for any $n > n_0$. But $\forall n = 2k+1$, where $k \in \mathbb{N}^+$, $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, which causes a contradiction.

2. The running time of solving all size-1 problem is:

$$a^{\log_b n} \cdot O(1) = O(n^{\log_b a})$$

The total combining is:

$$c \cdot (n^d \log^w n) + ca \cdot \left((\frac{n}{b})^d \log^w \frac{n}{b}\right) + \cdots + ca^k \left((\frac{n}{b^k})^d \log^w \frac{n}{b^k}\right) + \cdots + a^{\log_b n} \cdot c$$

Since $\log^w \frac{n}{b^k} \leq \log^w n$, The simplification is:

$$T(n) \leq cn^d \log^w n(1 + \frac{a}{b^d} + \cdots + (\frac{a}{b^d})^{\log_b n})$$

Thus $T(n) = O(n^d \log^w n)(1 + \frac{a}{b^d} + \cdots + (\frac{a}{b^d})^{\log_b n})$.

if $\frac{a}{b^d} < 1$, $T(n) = O(n^d \log^w n)$;

if $\frac{a}{b^d} = 1$, $T(n) = O(n^d \log^w n)(\log_b n + 1) = O(n^d \log^{w+1} n)$;

if $\frac{a}{b^d} > 1$, the final term conquers the sum. $T(n) \leq c \cdot a^{\log_b n} = O(n^{\log_b a})$.

3. (a)
   1. Construct set $C = A \cup B$, thus $|C| = n$. Then sort C in descending order by merge sort, which causes $O(n \log n)$ time.
   2. Iterate through sorted C from head to end and let $m = |B|$, $res = 0$ at first, which causes $O(n)$ time.
   3. For each vector i in sorted C, if $i \in A$, $res = m + res$; If $i \in B$, $m = m - 1$.
   4. print res.

   It's time complexity is $T(n) = O(n \log n) + O(n) = O(n \log n)$.

   (b) count(A, B): Use median-of-medians to find the median of x's value from $A \cup B$, denoted x_0;
   Then divide A and B into four parts:
   $A_1 = \{(x, y)|x \leq x_0, (x, y) \in A\}$
   $A_2 = \{(x, y)|x > x_0, (x, y) \in A\}$
   $B_1 = \{(x, y)|x \leq x_0, (x, y) \in B\}$
   $B_2 = \{(x, y)|x > x_0, (x, y) \in B\}$
   Then we compare A_2 and B_1's y's value. Use descending order sort $A_2 \cup B_1$ and iterate their y's value. As it is claimed in (a), it takes $O(n \log n)$ time.
   Finally we recursively divide A_2B_2 and A_1B_1 as we did above. Thus it turns $T(n)$ into $2T(\frac{n}{2}) + O(n \log n)$.
   According to 2 we can claim that the total time complexity is:

   $$T(n) = O(n \log^2 n) = O(n^{1.1})$$

   (c) We discuss about $d = 3$ situation.
   As the same above, we find the median of the first dimention and divide $A \cup B$ into two parts; Then for A_2 and B_1, we only need to consider them in 2 dimentions.
   Thus it time complexity is:

   $$T(n) = 2T(\frac{n}{2}) + O(n \log^{d-1} n) = O(n \log^d n)$$

4. (a) Since $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^2 = \begin{bmatrix} a^2+bc & b(a+d) \\ c(a+d) & bc+d^2 \end{bmatrix}$, the computation of$A^2$ requires to compute $a^2, d^2, bc, b(a+d), c(a+d)$.

   (b) In(a) we find that $bc, b(a+d), c(a+d)$ is not the square of a $n \times n$ matrix, thus it can not be computed in 5 multiplications.

   (c) 1. $AB + BA = (A+B)^2 - A^2 - B^2$, thus it takes $3S(n) + O(n^2)$ to compute.

   2. $AB + BA = \begin{bmatrix} 0 & XY \\ 0 & 0 \end{bmatrix}$.

   3. Since reading the numbers takes $O(n^2)$ time, $c > 2$. Then $3S(2n) + O(n^2) = 3O((2n)^c) + O(n^2) = O(n^c)$.

5. It takes me about 8 hours to finish the assignment. The difficulty is 4 according to me. I discussed with my roommates, they are Wang Kun and Li Haochen.