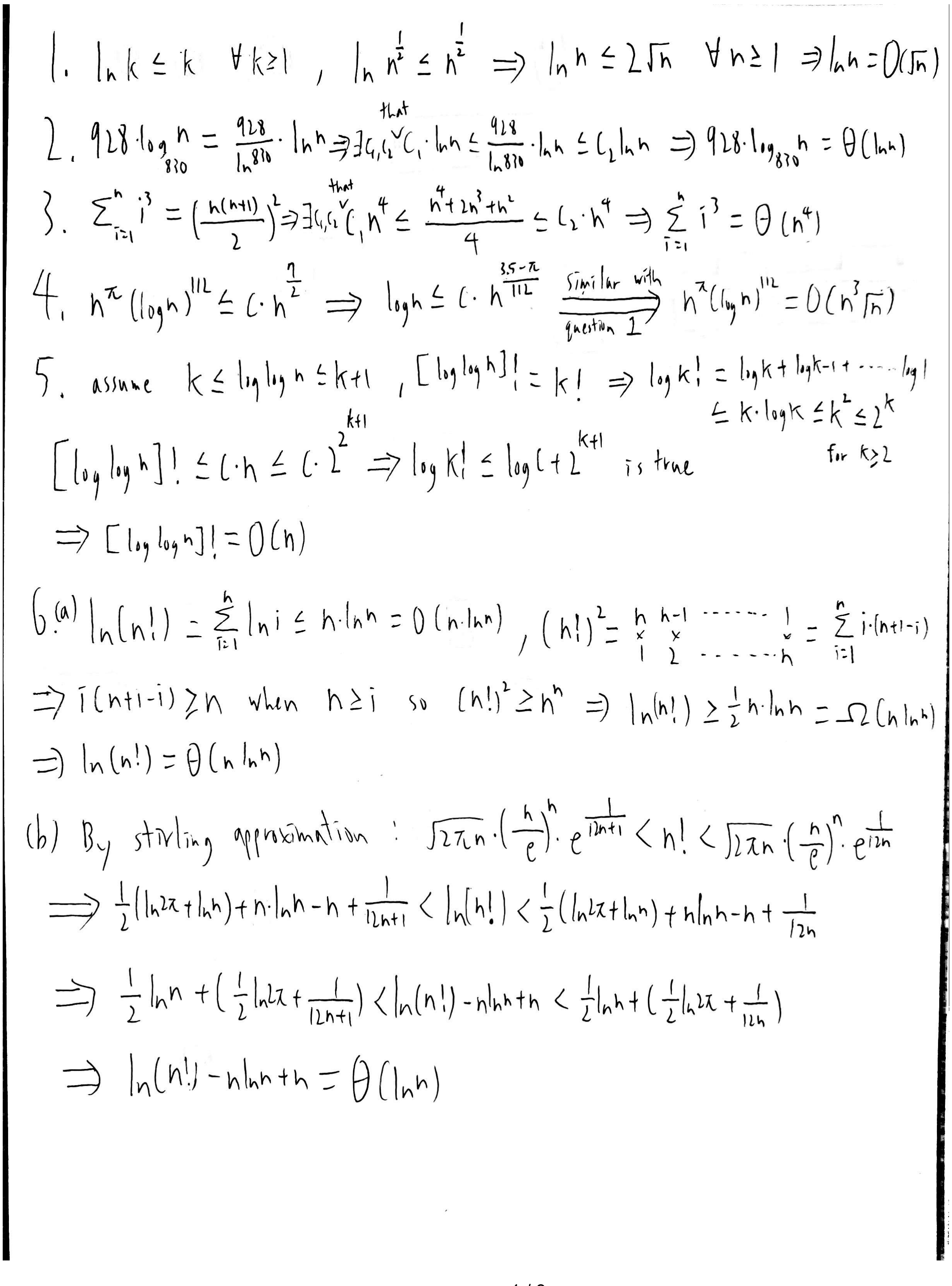
# Homework 1

**B10902033 林祐辰**

Problem 5:

(a) Complexity:

1.Prove 2.Prove 3.Prove 4.Prove 5.Prove 6.(a)Prove 6.(b)Prove(refer to formula in text book 3-20 and 3-21)



1. $\ln k \le k \quad \forall k \ge 1 \;,\; \ln n^{\frac{1}{2}} \le n^{\frac{1}{2}} \Rightarrow \ln n \le 2\sqrt{n} \quad \forall n \ge 1 \Rightarrow \ln n = O(\sqrt{n})$

2. $928 \cdot \log_{830} n = \frac{928}{\ln 830} \cdot \ln n \Rightarrow \exists c_1, c_2 \overset{\text{that}}{\vee} c_1 \cdot \ln n \le \frac{928}{\ln 830} \cdot \ln n \le c_2 \ln n \Rightarrow 928 \cdot \log_{830} n = \Theta(\ln n)$

3. $\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 \Rightarrow \exists c_1, c_2 \overset{\text{that}}{\vee} c_1 n^4 \le \frac{n^4 + 2n^3 + n^2}{4} \le c_2 \cdot n^4 \Rightarrow \sum_{i=1}^{n} i^3 = \Theta(n^4)$

4. $n^{\pi} (\log n)^{112} \le c \cdot n^{\frac{7}{2}} \Rightarrow \log n \le c \cdot n^{\frac{3.5 - \pi}{112}} \xrightarrow[\text{question } 1]{\text{Similar with}} n^{\pi} (\log n)^{112} = O(n^3 \sqrt{n})$

5. assume $k \le \log\log n \le k+1 \;,\; [\log\log n]! = k! \Rightarrow \log k! = \log k + \log k-1 + \cdots \log 1$
$\le k \cdot \log k \le k^2 \le 2^k$ for $k \ge 2$

$[\log\log n]! \le c \cdot n \le c \cdot 2^{2^{k+1}} \Rightarrow \log k! \le \log c + 2^{k+1}$ is true

$\Rightarrow [\log\log n]! = O(n)$

6.(a) $\ln(n!) = \sum_{i=1}^{n} \ln i \le n \cdot \ln n = O(n \cdot \ln n) \;,\; (n!)^2 = \begin{matrix} n & n-1 & \cdots & 1 \\ \times & \times & & \times \\ 1 & 2 & \cdots & n \end{matrix} = \sum_{i=1}^{n} i \cdot (n+1-i)$

$\Rightarrow i(n+1-i) \ge n$ when $n \ge i$ so $(n!)^2 \ge n^n \Rightarrow \ln(n!) \ge \frac{1}{2} n \cdot \ln n = \Omega(n \ln n)$

$\Rightarrow \ln(n!) = \Theta(n \ln n)$

(b) By stirling approximation : $\sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n \cdot e^{\frac{1}{12n+1}} < n! < \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n \cdot e^{\frac{1}{12n}}$

$\Rightarrow \frac{1}{2}(\ln 2\pi + \ln n) + n \cdot \ln n - n + \frac{1}{12n+1} < \ln(n!) < \frac{1}{2}(\ln 2\pi + \ln n) + n \ln n - n + \frac{1}{12n}$

$\Rightarrow \frac{1}{2} \ln n + \left(\frac{1}{2} \ln 2\pi + \frac{1}{12n+1}\right) < \ln(n!) - n \ln n + n < \frac{1}{2} \ln n + \left(\frac{1}{2} \ln 2\pi + \frac{1}{12n}\right)$

$\Rightarrow \ln(n!) - n \ln n + n = \Theta(\ln n)$

(b) Fill-in: 4 < 1 < 2 < 5 < 3 < 6

(c) Recursion: 1. O(n ^ 1.5) 2. O(n)

---

1. By Master Theorem $T(n) \begin{cases} O(1) & n \le 1 \\ 26T(\frac{n}{9}) + n^{1.5} & n > 1 \end{cases}$, $a = 26$, $b = 9$

$\log_b a = \log_9 26 < 1.5 \Rightarrow$ Case 3: $n^{1.5} = \Omega(n^{\log_9 26 + \epsilon})$ for some $\epsilon > 0$,

and $26 \cdot (\frac{n}{9})^{1.5} < c \cdot n^{1.5}$ for some $c < 1$, so $T(n) = \Theta(n^{1.5})$

2. By Recursion-Tree Method $T(n) \begin{cases} O(1) & n \le 1 \\ 3n + T(\frac{2n}{3}) + T(\frac{2n}{9}) & n > 1 \end{cases}$

$$\begin{tikzcd}[column sep=tiny]
 & & & 3n \arrow[dll, dash] \arrow[drr, dash] & & & & 3n \\
 & 3\cdot(\frac{2n}{3}) \arrow[dl, dash] \arrow[dr, dash] & & & & 3\cdot(\frac{2n}{9}) \arrow[dl, dash] \arrow[dr, dash] & & 3\cdot(\frac{8}{9})n \\
3\cdot(\frac{4n}{9}) & & 3\cdot(\frac{4n}{27}) & & 3(\frac{4n}{27}) & & 3(\frac{4n}{81}) & 3\cdot(\frac{8}{9})^2 n \\
 & & & & & & & \vdots
\end{tikzcd}$$

$$\Rightarrow T_n \le 3 \cdot (\frac{1}{1 - \frac{8}{9}}) n = 27n = O(n)$$

Problem 6:
(a)
1.False 2.True 3.False 4.True 5.True
(b)
Algorithm:
First do a merge sort on the n strings and in the merge functions, compair two strings character by character, when compairing put two characters in the gene comparator, if the results is two strings are the same then put them in a group, else continue doing the merge sort.
Correctness:
First two strings in the same group must be identical because the algorithm only put them in the same group if they are the same. Second the algorithm will correctly seperate different strings since merge sort will traverse through every strings at least one time.
Time complexity:
Because the time complexity of merge sort is O(n * log n) and in the merge function comparing two strings need O(k), so the total time complexity will be O(n * k * log n).
(c)
Algorithm: The concept of finding similar pairs in algorithm is that we can cut the strings in half and if we find the left side is the same than we do the algorithm again at the right, or else if the right side is the same than we do the algorithm again at the left.
So to accomplish this we can use the algorithm in question (b), first throw all n strings in (b) without cutting than we can take out one string in each group, and cut the strings in half throw the left side to (b) which will form same left group and recursively do the algorithm at the right side in same left group, similar for the right side.And at last when the string has two characters with same right or same left the one with different character left is the similar pair, so mutiply the number of group members in the same strings group which are similar and add them up, and we will get the total number of similar pairs.

(d)
Correctness:
The algorithm is correct cause while doing the divide and conquer we will only dismiss the strings that have different left side and different right side which won't be similar strings, and there won't be strings that are identical since we first seperate them into deifferent group, so the remaining ones will be similar pairs.
(e)

$$f(a) = a \cdot \log a \quad , \quad f(b) = b \cdot \log b \quad , \quad f(a+b) = a \cdot \log(a+b) + b \cdot \log(a+b)$$

$$\log(a+b) > \log a, \log b \implies a \cdot \log a + b \cdot \log b \leq a \cdot \log(a+b) + b \cdot \log(a+b)$$
$$\implies f(a) + f(b) \leq f(a+b)$$

(f)
Time complexity:
Because by divide and conquer we call the algorithm in (b) O(log k) times, so the time complexity would be O(n * k * log n * log k).
(g)

Reference:
Problem 1: https://www.geeksforgeeks.org/divide-and-conquer/
Problem 2: B10902034 李沛宸
Problem 3: B10902034 李沛宸, https://www.geeksforgeeks.org/counting-inversions/
Problem 4: B10902034 李沛宸
Problem 5: B10902034 李沛宸, textbook
Problem 6: B10902034 李沛宸