论题 2-4 作业

姓名：陈劭源　　学号：161240004

1 [CS] Problem 4.1-16

Failure occurs in inductive step. If we don't specify that the ears are nonadjacent, it is possible that two ears are connected by an edge. When rejoining two polygons into a larger one along the diagonal, the two ears of each polygons might be both incident to the diagonal, and they are eliminated once joined. Thus there is no ear in the new polygon, and we fail to complete the proof.

2 [CS] Problem 4.1-17

The relationship between the number of vertices in a polygon (V) and the number of triangles in any triangulation of tat polygon (N) is $V = N - 2$.

The base case is that the polygon is a triangle. In this case, $V = 3$, $N = 1$, so the relationship holds.

For every triangulated polygon, denoted by A, if it is not a triangle, then it must have at least diagonal. Split the polygon into two smaller polygons, denoted by B and C, and we have $V_B = N_B - 2$, $V_C = N_C - 2$. Then rejoin the two diagonals along the diagonal. Obviously we get $N_A = N_B + N_C$. When rejoining, two edges coincide to form the diagonal, so $N_C = V_B + V_C - 2 = N_B - 2 + N_C - 2 + 2 = N_A - 2$. Therefore, the relationship holds for the larger polygon.

By structural induction, we have proved that the relationship $V = N - 2$ holds for all polygons.

3 [CS] Problem 4.2-8

Assume the number of fish in the lake after n years is $T(n)$, then the recurrence is $T(n) = 2T(n-1) + 2000$.

This recurrence is a first-order linear recurrence. Apply Theorem 4.5, we obtain

$$\begin{aligned} T(n) &= 2^n T(0) + \sum_{i=1}^{n} 2^{n-i} \times 2000 \\ &= 2^n T(0) + 2000 \sum_{i=0}^{n-1} 2^i \\ &= 2^n T(0) + 2000(2^n - 1) \end{aligned}$$

4 [CS] Problem 4.2-11

Apply Theorem 4.5,

$$\begin{aligned} T(n) &= 2^n T(0) + \sum_{i=1}^{n} 2^{n-i} i 2^i \\ &= 2^n + \sum_{i=1}^{n} i 2^n \\ &= 2^n + 2^n \frac{n(n+1)}{2} \end{aligned}$$

5 [CS] Problem 4.2-17

Apply Theorem 4.5,

$$\begin{aligned} T(n) &= r^n T(0) + \sum_{i=1}^{n} r^{n-i} i \\ &= r^n + \sum_{i=1}^{n} i r^{n-i} \end{aligned}$$

Now we have to calculate $\sum_{i=1}^{n} i r^{n-i}$. Let $S(n)$ denote $\sum_{i=1}^{n} i r^{n-i}$, then we have

$$S(n) = \quad 1 \times r^{n-1} + \cdots + (n-1) \times r^1 + n \times r^0 \tag{1}$$

$$rS(n) = 1 \times r^n + 2 \times r^{n-1} + \cdots + n \times r^1 \tag{2}$$

Subtracting (1) from (2), we obtain

$$\begin{aligned} (r-1)S(n) &= \sum_{i=1}^{n} r^i - n \\ &= \frac{r^{n+1} - r}{r-1} - n \end{aligned}$$

Since $r \neq 1$, dividing both sides by $(r-1)$ yields

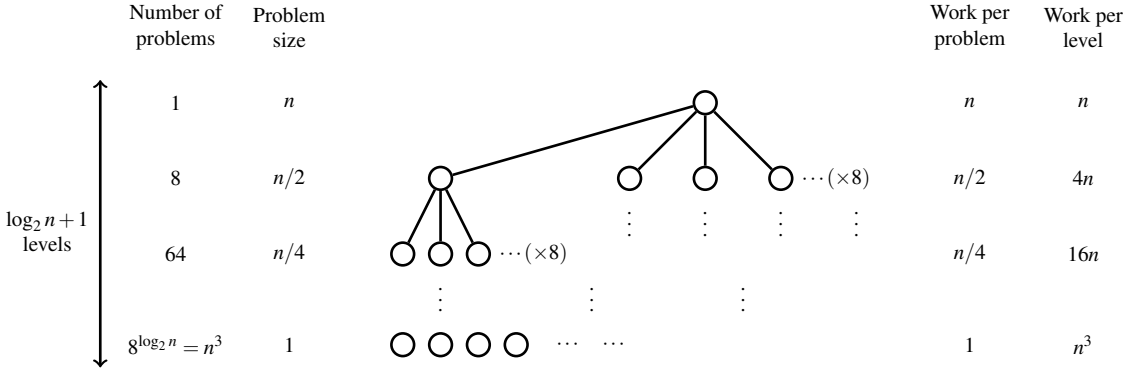
$$S(n) = \frac{r^{n+1} - r - nr + n}{(r-1)^2}$$

Therefore

$$T(n) = r^n + \frac{r^{n+1} - r - nr + n}{(r-1)^2}$$

6 [CS] Problem 4.3-9

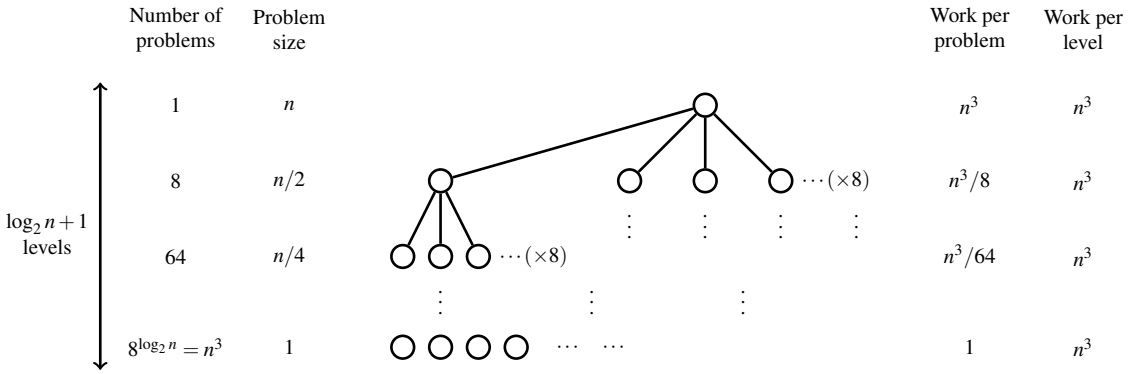
a. Recursion tree:



For the i-th level, the total work is $4^{i-1}n$. Summing over the levels, we get

$$\begin{aligned} T(n) &= \sum_{i=1}^{\log_2 n+1} 4^{i-1}n \\ &= n\sum_{i=0}^{\log_2 n} 4^i \\ &= n\frac{4^{\log_2 n+1}-1}{4-1} \\ &= \frac{4}{3}n^3 - \frac{n}{3} \\ &= \Theta(n^3) \end{aligned}$$

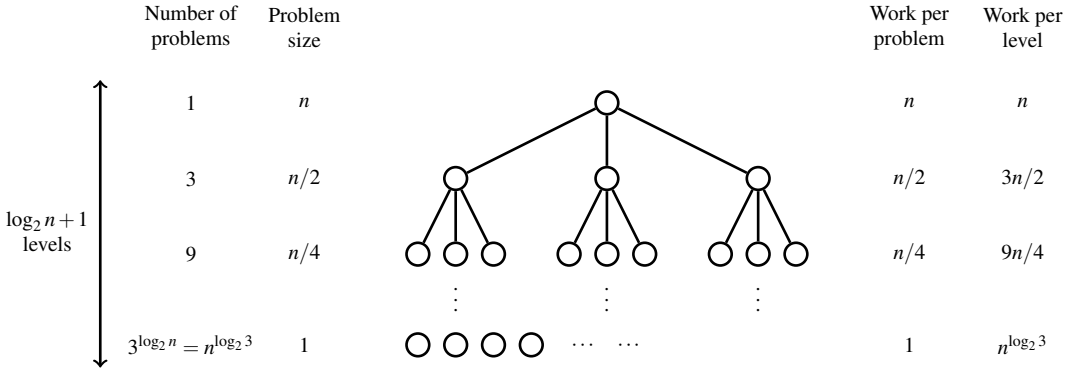
b. Recursion tree:



For every level, the total work is n^3, and there are $(\log_2 n+1)$ levels in total, therefore

$$T(n) = n^3(\log_2 n+1) = \Theta(n^3 \log n)$$

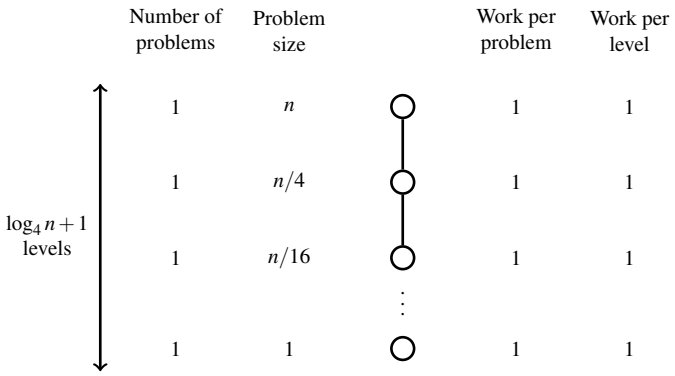
c. Recursion tree:



For the i-th level, the total work is $(3/2)^{i-1}n$. Summing over the levels, we get

$$\begin{aligned} T(n) &= \sum_{i=1}^{\log_2 n+1} (3/2)^{i-1}n \\ &= n\sum_{i=0}^{\log_2 n} (3/2)^i \\ &= n\frac{(3/2)^{\log_2 n+1}-1}{(3/2)-1} \\ &= n^{\log_2 3} - 2n \\ &= \Theta(n^{\log_2 3}) \end{aligned}$$

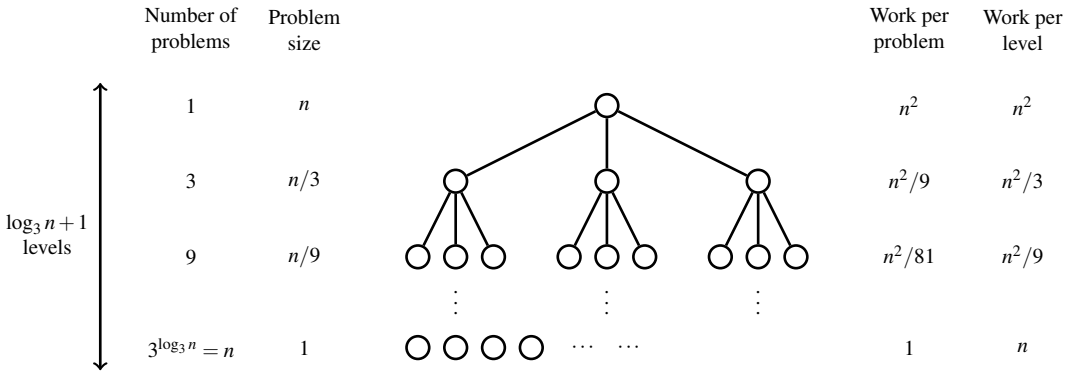
d. Recursion tree:



For every level, the total work is 1, and there are $(\log_4 n + 1)$ levels in total, therefore

$$T(n) = \log_4 n + 1 = \Theta(\log n)$$

e. Recursion tree:



For the i-th level, the total work is $n^2/3^{i-1}$. Summing over the levels, we get

$$\begin{aligned} T(n) &= \sum_{i=1}^{\log_3 n+1} n^2/3^{i-1} \\ &= n^2 \sum_{i=0}^{\log_2 n} (1/3)^i \\ &= \frac{n^2}{2}(3 - n^{-\log_2 3}) \\ &= \Theta(n^2) \end{aligned}$$

7 [CS] Problem 4.3-13

$a^{\log_b n} = b^{(\log_b a)\log_b n} = b^{(\log_b n)\log_b a} = n^{\log_b a}$

8 [CS] Problem 4.3-16

Apply Theorem 4.5, we have

$$S(n) = a^n S(0) + \sum_{i=1}^{n} a^{n-i} g(i)$$

On the one hand, we have

$$S(n) \geq a^n S(0)$$

on the other hand, we have

$$\begin{aligned} S(n) &\leq a^n S(0) + \sum_{i=1}^{n} a^{n-i} c^i \\ &= a^n S(0) + a^n \sum_{i=1}^{n} (c/a)^i \\ &\leq a^n S(0) + a^n \frac{c}{a-c} \\ &= a^n (S(0) + \frac{c}{a-c}) \end{aligned}$$

Therefore, $S(n) = \Theta(a^n)$.

9 [CS] Problem 4.4-1

a. In this recurrence, we have $a = 8$, $b = 2$, $d = 1$, $f(n) = n = \Theta(n^c)$ where $c = 1 < \log_b a = 3$. Apply the master theorem, case 3, and we obtain $T(n) = \Theta(n^{\log_a b}) = \Theta(n^3)$.

b. In this recurrence, we have $a = 8$, $b = 2$, $f(n) = n^3 = \Theta(n^c)$ where $c = 3 = \log_b a$. Apply the master theorem, case 2, and we obtain $T(n) = \Theta(f(n) \log n) = \Theta(n^3 \log n)$.

c. In this recurrence, we have $a = 3$, $b = 2$, $f(n) = n = \Theta(n^c)$ where $c = 1 < \log_b a = \log_2 3$. Apply the master theorem, case 3, and we obtain $T(n) = \Theta(n^{\log_a b}) = \Theta(n^{\log_2 3})$.

d. In this recurrence, we have $a = 1$, $b = 4$, $f(n) = 1 = \Theta(n^c)$ where $c = 0 = \log_b a$. Apply the master theorem, case 2, and we obtain $T(n) = \Theta(f(n) \log n) = \Theta(\log n)$.

e. In this recurrence, we have $a = 3$, $b = 3$, $f(n) = n^2 = \Theta(n^c)$ where $c = 2 > \log_b a$. Apply the master theorem, case 1, and we obtain $T(n) = \Theta(f(n)) = \Theta(n^2)$.

10 [CS] Problem 4.4-4

In this recurrence, $a = 3$, $b = 2$, $f(n) = \sqrt{n^4 + 3} = \Theta(n^c)$ where $c = 4 > \log_b a = \log_2 3$. Apply Theorem 4.11, case 1, and we obtain $T(n) = \Theta(f(n)) = \Theta(n^4)$.

11 [CS] Problem 4.4-6

The only difference is that, the work of the last level of the recursion tree, is no longer $n^c \left(\frac{a}{b^c}\right)^{\log_b n} = n^{\log_b a}$, but $dn^{\log_b a}$, where $d \geq 0$. Hence, rewrite the summation, we get

$$T(n) = n^c \sum_{i=0}^{\log_b n - 1} \left(\frac{a}{b^c}\right)^i + dn^{\log_b a}$$

1. If $\log_b a < c$, i.e. $a < b^c$, by Theorem 4.4, we have $T(n) = n^c \Theta(1) + O(n^c) = \Theta(n^c)$.
2. If $\log_b a = c$, i.e. $a = b^c$, we have $T(n) = n^c \Theta(\log_b n) + O(n^c) = \Theta(n^c \log n)$.
3. If $\log_b a > c$, i.e. $a > b^c$, by Theorem 4.4, we have

$$T(n) = n^c \Theta(\left(\frac{a}{b^c}\right)^{\log_b n - 1}) + O(n^{\log_b a}) = \Theta(\left(\frac{n}{n-1}\right)^c (n-1)^{\log_b a}) + O(n^{\log_b a}) = \Theta(n^{\log_b a})$$

Therefore, all three cases of the master theorem, preliminary version hold for general d.

12 [CS] Problem 4.5-8

To prove $T(n) = O(n^3)$, we have to prove that, there exists some positive constant k, such that for every positive integer n, $T(n) \leq kn^3 - n\log n$ holds.

For the base step, $n = 1$, we have $T(n) = d$, $n^3 = 1$ and $n\log n = 0$, therefore $T(n) \leq kn^3 - n\log n$ holds for every $k > d$ when $n = 1$.

For the induction step, assume that $T(m) \leq km^3 - m\log m$ holds for all $m < n$. Substituting into the recurrence yields

$$\begin{aligned} T(n) &= 8T(n/2) + n\log n \\ &\leq 8(k(n/2)^3 - (n/2)\log(n/2) + n\log n \\ &= kn^3 - 4n\log n - 4n\log 2 + n\log n \\ &\leq kn^3 - n\log n \end{aligned}$$

By mathematical induction, we conclude that $T(n) \leq kn^3 - n\log n$ where $k > d$. Hence, $T(n) = O(kn^3 - n\log n) = O(n^3)$.

13 [CS] Problem 4.5-9

Yes.

Define $S(n) = \begin{cases} 8S(n/2) + n & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases}$, it is true that $S(n) \leq T(n)$ because $n < n\log n$ holds for every $n \geq 2$. For $S(n)$, $a = 8$, $b = 2$, $f(n) = n = n^c$ where $c = 1 < \log_2 8 = 3$. Apply the master theorem, case 3, and we obtain $S(n) = \Theta(n^{\log_b a}) = \Theta(n^3)$. Because $S(n) \leq T(n)$, we get $T(n) = \Omega(n^3)$. We have proved that $T(n) = O(n^3)$ in Problem 8, so $T(n) = \Theta(n^3)$.

14 [CS] Problem 4.5-10

$T(n) = O(n)$.

To prove $T(n) = O(n)$, we have to prove that there exists some positive constant k, such that $T(n) \leq kn$ holds for every positive integer n.

For the base step, we have to prove that $T(n) \leq kn$ holds for every positive integer $n \leq 12$. This inequality holds if we choose $k > \max_{0<n<12} T(n)/n$.

For the induction step, assume that $T(m) \leq km$ holds for every $m < n$. Substituting into the recurrence yields

$$\begin{aligned} T(n) &= 2T(\frac{n}{3} - 3) + n \\ &\leq 2k(\frac{n}{3} - 3) + n \\ &= \left(\frac{2k}{3} + 1\right) n - 6k \\ &\leq kn \end{aligned}$$

The last step holds if we choose $k > 3$.

By mathematical induction, we conclude that $T(n) < kn$ holds for every positive integer if we choose $k > \max\{\max_{0<n<12} T(n)/n, 3\}$. Therefore $T(n) = O(n)$.