

VE477 HW2

Wu Jiayao 517370910257

October 2019

1 Ex. 1 — Basic complexity

1.1

Let $T(n)$ be the equation on the left of =.

a) Choose $n_0 = 1$ and $c = 1 + 3 + 1 + 1 = 6$, so that for any $n > n_0$,

$$T(n) \leq n^3 + 3n^3 + n^3 + 1 = c \cdot n^3$$

. Hence, $T(n) = \mathcal{O}(n^3)$. Choose $n_0 = 10$ and $c = \frac{1}{4}$, so that for any $n > n_0$,

$$T(n) \geq n^3 - \frac{3}{10}n^3 - \frac{1}{100}n^3 + 1 \geq \frac{69}{100} \cdot n^3 \geq c \cdot n^3$$

. Hence, $T(n) = \Omega(n^3)$. Hence,

$$T(n) = \Theta(n^3)$$

b) Choose $n_0 = 4$ and $c = 1$. Transform as followed.

$$\ln(2^n) = n \ln(2)$$

$$\ln(n^2) = 2 \ln(n)$$

When $n > 4$

$$\frac{d}{dn} 2 \ln(n) = \frac{2}{n} < \frac{d}{dn} n \ln(2) = \ln(2)$$

Hence, when $n > 4$

$$n^2 < 2^n$$

Hence, for $\forall n > 4$, there exists $c = 1$ that

$$n^2 < c \cdot 2^n$$

which means,

$$n^2 = \mathcal{O}(2^n)$$

1.2

a) $\forall n > 1$, there exists $c = 2$ that

$$f(n) = n\sqrt{n} < n^2 < 2n^2 - n^2 < 2(n^2 - 1) = c \cdot g(n)$$

$$f(n) = \mathcal{O}g(n)$$

1.3

a) Do not exist

b) $f(n) = n, g(n) = 1$

1.4

When $n \rightarrow \infty$

$$f_4(n) > f_1(n) > f_3(n) > f_2(n)$$

First,

$$\frac{f_3(n)}{f_2(n)} = \frac{\sqrt{n}}{\sqrt{\log(n)}} > 1 \Rightarrow f_3 > f_2$$

Since $\sqrt{n} > \sqrt{\log(n)}$

$$f_3 < n\sqrt{n} = n^{\frac{3}{2}} < 24n^{\frac{3}{2}} < f_4$$

To compare f_3 with f_1

$$f_1(n) = \sum_{i=1}^{n/2} (\sqrt{i} + \sqrt{n+1-i}) > \frac{n}{2}(\sqrt{n} + 1)$$
$$f_3(n) = \frac{n}{2}(2\sqrt{\log(n)})$$

When $n > 16$

$$(\sqrt{n} + 1)^2 - (2\sqrt{\log(n)})^2 > n - 4\log(n) > 0$$

Therefore,

$$\sqrt{n} + 1 > 2\sqrt{\log(n)}$$

which means, $\forall n > 16$

$$f_1(n) > \frac{n}{2}(\sqrt{n} + 1) > \frac{n}{2}(2\sqrt{\log(n)}) = f_3(n)$$

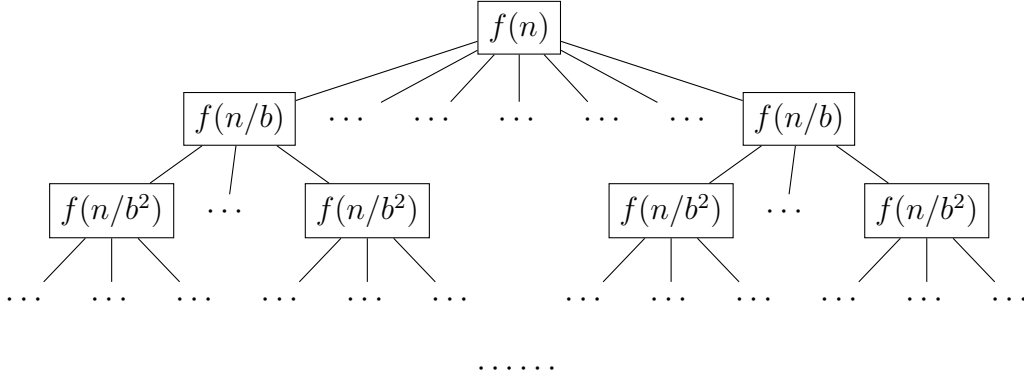
What's more, $\forall n > 1$

$$f_4(n) > n\sqrt{n} > f_1(n)$$

2 Ex. 2 Master Theorem

2.1

a)



Each node has b child nodes.

b)

The depth of the tree is $\log_b n + 1$.

The number of the leaves is $a^{\log_b n}$.

The total cost at depth k is $a^k \cdot f(\frac{n}{b^k})$

We can conclude that

$$\begin{aligned}
 T(n) &= \sum_{j=0}^{\log_b n} a^j f\left(\frac{n}{b^j}\right) \\
 &= \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) + a^{\log_b n} f(1) \\
 &= \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) + \Theta(a^{\log_b n}) \\
 &= \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) + \Theta(n^{\log_b a})
 \end{aligned}$$

where,

$$a^{\log_b n} = a^{\frac{\log_a n}{\log_a b}} = n^{-\log_a b} = n^{\log_b a}$$

2.2

a)i. Since $f(n) = \Theta(n^{\log_b a})$, there exists $n'_0 = n_0 \cdot b^j$, and $c_1 > 0, c_2 > 0$ that $\forall n > n'_0$

$$c_1 \cdot \frac{n^{\log_b a}}{b^j} \leq a^j \cdot f\left(\frac{n}{b^j}\right) = a^j \cdot \Theta\left(\frac{n}{b^j}\right)^{\log_b a} \leq c_2 \cdot \frac{n^{\log_b a}}{b^j}$$

By adding all the $\log_b n$ items together, we get that there exists $n'_0 = n_0 \cdot b^j$, and $c_1 > 0, c_2 > 0$ that $\forall n > n'_0$

$$c_1 \cdot \sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a} \leq g(n) \leq c_2 \cdot \sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}$$

which means that

$$g(n) = \Theta\left(\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}\right)$$

a)iii.

$$g(n) = \Theta\left(\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}\right) = \Theta(n^{\log_b a} \log_b n)$$

b)ii.

$$\begin{aligned} \sum_{i=0}^{\log_b n-1} a^i (n/b^i)^{\log_b a-\varepsilon} &= n^{\log_b a-\varepsilon} \sum_{i=0}^{\log_b n-1} a^i b^{-i \log_b a} b^{i\varepsilon} \\ &= n^{\log_b a-\varepsilon} \sum_{i=0}^{\log_b n-1} a^i a^{-i} b^{i\varepsilon} \\ &= n^{\log_b a-\varepsilon} \sum_{i=0}^{\log_b n-1} b^{\varepsilon i} \\ &= n^{\log_b a-\varepsilon} \frac{b^{\varepsilon(\log_b n)} - 1}{b^{\varepsilon} - 1} \\ &= n^{\log_b a-\varepsilon} \frac{n^{\varepsilon} - 1}{b^{\varepsilon} - 1} \end{aligned}$$

b)iii.

$$g(n) = n^{\log_b a-\varepsilon} \frac{n^{\varepsilon} - 1}{b^{\varepsilon} - 1} = n^{\log_b a-\varepsilon} \frac{n^{\varepsilon} - 1}{b^{\varepsilon} - 1} \leq n^{\log_b a-\varepsilon} \frac{n^{\varepsilon}}{b^{\varepsilon} - 1} = n^{\log_b a} \frac{1}{b^{\varepsilon} - 1}$$

Therefore,

$$g(n) = \mathcal{O}(n^{\log_b a})$$

2.3

i. Take $c = 1, n_0 = 1$, we observe that

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j) = f(n) + \sum_{j=1}^{\log_b n-1} a^j f(n/b^j) > f(n)$$

Hence,

$$g(n) = \Omega(f(n))$$

ii. Let $t = n/b^{j-1}$

$$a^j f(n/b^j) = a^j f(t/b) = a^{j-1} \cdot a f(t/b) \leq a^{j-1} c f(t) = c \cdot a^{j-1} f(n/b^{j-1})$$

Similarly,

$$a^j f(n/b^j) \leq c \cdot a^{j-1} f(n/b^{j-1}) \leq c^2 \cdot a^{j-2} f(n/b^{j-2}) \leq \dots \leq c^j f(n)$$

iv. Since $c < 1$

$$\begin{aligned} g(n) &= \sum_{j=0}^{\log n-1} a^j f(n/b^j) \\ &= a^0 f(n/b^0) + a^1 f(n/b^1) + a^2 f(n/b^2) + \dots + a^{\log_b n-1} f(n/b^{\log_b n-1}) \\ &\leq f(n) + c f(n) + c^2 f(n) + \dots + c^{\log_b n-1} f(n) \\ &< \frac{1}{1-c} f(n) \end{aligned}$$

Take $c_1 = 1, c_2 = \frac{1}{1-c}, n_0 = 1, \forall n > n_0$

$$c_1 f(n) \leq g(n) \leq c_2 f(n)$$

Hence,

$$g(n) = \Theta(f(n))$$

2.4

Consider the recurrence $T(n) = aT(n/b) + f(n)$, where a, b are constants, n is an exact power of b . Then

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & f(n) = \Theta(n^{\log_b a}) \\ \Theta(n^{\log_b a}) & f(n) = O(n^{\log_b a - \epsilon}) \\ \Theta(f(n)) & f(n) \geq \frac{a}{c} f(n/b) \end{cases}$$

3 Ex. 3 Ramanujam numbers

```

input : An integer n
output: All Ramanujam numbers smaller or equal to n
1 i ← 2;
2 cubeSum_list ← [];
3 while 1 do
4   if  $i^3 \geq n$  then
5     break;
6   for all  $j < i$  do
7     if  $i^3 + j^3 \leq n$  then
8       Push  $(i^3 + j^3)$  into cubeSum_list;
9   i++;
10 for All the numbers in cubeSum_list do
11   return All the numbers that have occurred twice

```

There are at most $1^2 + 2^2 + \dots + \sqrt[3]{n}^2 = \mathcal{O}(n)$ iterations. *cubeSum_list* contains at most $2n$ elements. The complexity is $\mathcal{O}(n)$, calculated as

$$\mathcal{O}(n) + 2n = \mathcal{O}(n)$$

4 Ex. 4 Critical Thinking

Suppose that No.1 pirate is the most senior pirate.

Situation 1: When there is only one pirate left, No.6. He can get 300.

Situation 2: When there are two left, No. 5 and No.6, No.5 will give himself 300, No.6 0. Since No.5 will agree with his distribution (1 of 2 agrees), it's ok.

Situation 3: When there are three left, No.4 - No.6. However No.4 distributes, No.5 will vote NO to get into Situation 2. Hence, No.5 will be distributed 0. To avoid Situation 2, No.6 will vote YES if he gets at least 1. No.6 will be distributed 1.

Situation 4: When there are four left, No.3 - No.6. No.3 needs two votes. No.4 will always vote NO to get 299 in Situation 3, where No.5 could get nothing. Hence, No.5 will vote YES if distributed

at least 1. Then here are two votes. No.5 will be distributed 1, No.4 himself 299.

Situation 5: When there are five left, three votes are needed. Similarly, No.3 will vote NO to get more in Situation 4. Since No.4 and No.6 could get nothing in Situation 4, they will vote YES if distributed at least 1. Hence, No.2 298, No.4 1, No.6 1.

Situation 6: It works the same. No.3 and No.5 will vote YES if distributed at least 1. Hence, No.1 298, No.3 1, No.5 1.

It works like this.

pirates alive	1	2	3	4	5	6
1	X	X	X	X	X	300
2	X	X	X	X	300	0
3	X	X	X	299	X	1
4	X	X	299	0	1	0
5	X	298	0	1	0	1
6	298	0	1	0	1	0