Midterm 2 Version 3 Solution

Hyungmo Gu

April 5, 2020

Question 1

a.

$$\begin{aligned}
165 \div 2 &= 82, \text{remainders } \mathbf{1} \\
82 \div 2 &= 41, \text{remainders } \mathbf{0} \\
41 \div 2 &= 20, \text{remainders } \mathbf{1} \\
20 \div 2 &= 10, \text{remainders } \mathbf{0} \\
10 \div 2 &= 5, \text{remainders } \mathbf{0} \\
5 \div 2 &= 2, \text{remainders } \mathbf{1} \\
2 \div 2 &= 1, \text{remainders } \mathbf{0} \\
1 \div 2 &= 0, \text{remainders } \mathbf{1}
\end{aligned}$$

From the above, we can conclude the binary representation of the decimal number 165 is $(10100101)_2$

b. The largest number that can be expressed by an n-digit balanced ternary representation is

$$\sum_{i=0}^{n-1} 3^i = \frac{1}{2} \cdot (3^n - 1) \tag{1}$$

Notes:

- Geometric Series

$$\sum_{i=0}^{n} r^i = \frac{1 - r^{n+1}}{1 - r}, \text{ where } |r| > 1$$

c.

$f(n) \in \mathcal{O}(n)$	True	$g(n) \in \Omega(n)$	True	$f(n) \in \Omega(g(n))$	True
$f(n) \in \Theta(g(n))$	False	$g(n) \in \Theta(n)$	False	$f(n) + g(n) \in \Theta(g(n))$	False

Correct Solution:

$f(n) \in \mathcal{O}(n)$	True	$g(n) \in \Omega(n)$	True	$f(n) \in \Omega(g(n))$	True
$f(n) \in \Theta(g(n))$	False	$g(n) \in \Theta(n)$	False	$f(n) + g(n) \in \Theta(g(n))$	True

Notes:

- Note that for $f(n) + g(n) \in \Theta(g(n))$, large values of n causes $g(n) = n^{\log_2 n}$ to dominate $f(n) = \frac{3n}{\log_2 n+8}$. This causes the inequality to be simplified to

$$c_1 \cdot n^{\log_2 n} \leq n^{\log_2 n} \leq c_2 \cdot n^{\log_2 n} \tag{1}$$

  It follows from above the answer is True.

d.

k	0	1	2
$i * i$	$3 = 3^{2^0}$	$9 = 3^{2^1}$	$81 = 3^{2^4}$

From the rough work, we can deduce the value of i after k iterations is

$$3^{2^k} \tag{1}$$

e. Loop termination occurs when $i_k \geq n^3$.

We need to find the smallest value of k, and the value is

$$\lceil \log_2 3 \log_3 n \rceil \tag{1}$$

Question 2

- **Statement:** $\forall n \in \mathbb{N},\ n \geq 1 \Rightarrow \sum_{i=1}^{n} \frac{1}{\sqrt{i}} > \sqrt{n} - 1$

Proof. Let $n \in \mathbb{N}$. Assume $n \geq 1$.

We will prove the statement using induction on n.

Base Case ($n = 1$)**:**

Let $n = 1$.

We want to show $\sum_{i=1}^{1} \frac{1}{\sqrt{i}} > \sqrt{1} - 1$

Starting from the left hand side of the inequality, we can calculate

$$\begin{aligned}
\sum_{i=1}^{1} \frac{1}{\sqrt{i}} &= \frac{1}{\sqrt{1}} && (1)\\
&= 1 && (2)\\
&> 0 && (3)\\
&= \sqrt{1} - 1 && (4)
\end{aligned}$$

Inductive Case:

Let $n \in \mathbb{N}$. Assume $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} > \sqrt{n} - 1$.

We want to show $\sum_{i=1}^{n+1} \frac{1}{\sqrt{i}} > \sqrt{n+1} - 1$.

Starting from the left hand side of the inequality, we can conclude

$$\sum_{i=1}^{n+1} \frac{1}{\sqrt{i}} = \sum_{i=1}^{n} \frac{1}{\sqrt{i}} + \frac{1}{\sqrt{n+1}} \tag{5}$$

Then, it follows from induction hypothesis (i.e. $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} > \sqrt{n} - 1$) that

$$\sum_{i=1}^{n+1} \frac{1}{\sqrt{i}} > \sqrt{n+1} - 1 + \frac{1}{\sqrt{n+1}} \tag{6}$$

The hint tells us the following

$$\forall n \in \mathbb{Z}^{+},\ \sqrt{n+1} - \sqrt{n} = \frac{1}{\sqrt{n+1} + \sqrt{n}},\text{ and } \frac{1}{\sqrt{n+1} + \sqrt{n}} < \frac{1}{\sqrt{n+1}} \tag{7}$$

Using the hint, we can calculate

$$\sum_{i=1}^{n+1} \frac{1}{\sqrt{i}} > (\sqrt{n} - 1) + \frac{1}{\sqrt{n+1} + \sqrt{n}} \tag{8}$$

$$= (\sqrt(n) - 1) + (\sqrt{n+1} - \sqrt{n}) \tag{9}$$

$$= \sqrt{n+1} - 1 \tag{10}$$

$\square$

Question 3

- **Statement:** $\forall a \in \mathbb{R}^+$, $a > 1 \Rightarrow a^n + 3 \in \Theta(2^n)$

  Negation Statement: $\exists a \in \mathbb{R}^+$, $(a > 1) \wedge \Big[\forall c_1, c_2, n_0 \in \mathbb{R}^+, \exists n \in \mathbb{N}, (n \geq n_0) \wedge \big((c_1 \cdot (a^n + 3) > 2^n) \vee (2^n > c_2 \cdot (a^n + 3))\big)\Big]$

  Proof. Let $a = \frac{3}{2}$. Let $c_1, c_2, n_0 \in \mathbb{R}^+$, and $n = \left\lceil max(n_0, \frac{\log(2c_1)}{\log \frac{4}{3}}, \frac{\log(6c_1)}{2}) \right\rceil + 1$.

  We need to show $a > 1$, $n > n_0$ and $a^n + 3 < \frac{1}{c_1} \cdot 2^n$.

  We will do so in parts.

  Part 1 (showing $a > 1$):

  We need to show $a > 1$.

  Because we know $a = \frac{3}{2}$ from header, we can conclude

$$a > 1 \tag{1}$$

  Part 2 (showing $n > n_0$):

  We need to show $n \geq n_0$.

  Using the fact $\left\lceil max(n_0, \frac{\log(2c_1)}{\log \frac{4}{3}}, \frac{\log(6c_1)}{2}) \right\rceil \geq n_0$, we can conclude

$$n = \left\lceil max(n_0, \frac{\log(2c_1)}{\log \frac{4}{3}}, \frac{\log(6c_1)}{2}) \right\rceil + 1 > n_0 \tag{2}$$

Part 3 (Showing $a^n + 3 < \frac{1}{c_1} \cdot 2^n$):

We need to show $a^n + 3 < \frac{1}{c_1} \cdot 2^n$.

We will do so by showing $a^n < \frac{2^n}{2c_1}$, $3 < \frac{2^n}{2c_1}$, and then combining the two.

For the inequality $a^n < \frac{2^n}{2c_1}$, using the following fact

$$\frac{\log(2c_1)}{\log\frac{4}{3}} < \left\lceil max(n_0, \frac{\log(2c_1)}{\log\frac{4}{3}}, \frac{\log(6c_1)}{2}) \right\rceil + 1 = n \tag{3}$$

we can calculate

$$\begin{align}
\log(2c_1) &< n \log\frac{4}{3} \tag{4}\\
2c_1 &< \left(\frac{4}{3}\right)^n \tag{5}\\
2c_1 &< \frac{2^n}{\left(\frac{3}{2}\right)^n} \tag{6}\\
\left(\frac{3}{2}\right)^n &< \frac{2^n}{2c_1} \tag{7}\\
\tag{8}
\end{align}$$

Then, since $a = \frac{3}{2}$, we can conclude

$$a^n < \frac{2^n}{2c_1} \tag{9}$$

For the inequality $3 < \frac{2^n}{2c_1}$, using the following fact

$$\frac{\log(6c_1)}{2} < \left\lceil max(n_0, \frac{\log(2c_1)}{\log\frac{4}{3}}, \frac{\log(6c_1)}{2}) \right\rceil + 1 = n \tag{10}$$

we can conclude

$$\begin{align}
6c_1 &< 2^n \tag{11}\\
3 &< \frac{2^n}{2c_1} \tag{12}
\end{align}$$

Finally, by combining the two, we can conclude

$$a^n + 3 < \frac{2^n}{2c_1} + \frac{2^n}{2c_1} \tag{13}$$

$$a^n + 3 < \frac{2^n}{c_1} \tag{14}$$

□

Question 4

a. *Proof.* Let $n \in \mathbb{N}$.

First, we need to analyze the number of iterations of loop 2 per iteration of loop 1

The code tells us loop 2 starts at $j = 0$, ends at $j = i - 1$, and j increases by 2 per iteration.

Using the fact, we can conclude that the number of iterations of loop 2 is

$$\left\lceil \frac{i - 1 - 0 + 1}{2} \right\rceil = \left\lceil \frac{i}{2} \right\rceil \tag{1}$$

or

$$\frac{i}{2} \tag{2}$$

since we are ignoring floor and ceiling symbols.

Next, we need to add this number over all iterations of loop 1.

The code tell us us it starts at $i = 0$ and ends at $i = \sqrt{n - 1}$ with i increasing by 1 per iteration.

Using this fact, the total number of iterations of loop 2 is

$$\frac{0}{2} + \frac{1}{2} + \cdots + \frac{\sqrt{n-1}}{2} = \sum_{i=0}^{\sqrt{n-1}} \frac{i}{2} \tag{3}$$

$$= \frac{1}{2} \cdot \sum_{i=0}^{\sqrt{n-1}} i \tag{4}$$

Then, it follows from the fact $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ that,

$$\frac{1}{2} \cdot \sum_{i=0}^{\sqrt{n-1}} i = \frac{1}{2} \cdot \frac{(\sqrt{n-1})(\sqrt{n-1}+1)}{2} \tag{5}$$

$$= \frac{(\sqrt{n-1})(\sqrt{n-1}+1)}{4} \tag{6}$$

□

Correct Solution:

Let $n \in \mathbb{N}$.

First, we need to analyze the number of iterations of loop 2 per iteration of loop 1

The code tells us loop 2 starts at $j = 0$, ends at $j = i - 1$, and j increases by 2 per iteration.

Using the fact, we can conclude that the number of iterations of loop 2 is

$$\left\lceil \frac{i-1-0+1}{2} \right\rceil = \left\lceil \frac{i}{2} \right\rceil \tag{1}$$

or

$$\frac{i}{2} \tag{2}$$

since we are ignoring floor and ceiling symbols.

Next, we need to add this number over all iterations of loop 1.

The code tell us us it starts at $i = 0$ with i increasing by 1 per iteration until the terminating condition of $i^2 \geq n$.

And from the inequality, we know the loop finishes at $i = \sqrt{n} - 1$.

Using these facts, the total number of iterations of loop 2 is

$$\frac{0}{2} + \frac{1}{2} + \cdots + \frac{\sqrt{n}-1}{2} = \sum_{i=0}^{\sqrt{n}-1} \frac{i}{2} \tag{3}$$

$$= \frac{1}{2} \cdot \sum_{i=0}^{\sqrt{n}-1} i \tag{4}$$

Then, it follows from the fact $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ that,

$$\frac{1}{2} \cdot \sum_{i=0}^{\sqrt{n}-1} i = \frac{1}{2} \cdot \frac{(\sqrt{n}-1)(\sqrt{n}-1+1)}{2} \tag{5}$$

$$= \frac{(\sqrt{n}-1)\sqrt{n}}{4} \tag{6}$$

Notes:

- How can I construct a proof in a situation where the slight change in a problem causes the flow of proof to deviate from the main?

  One example of how I mean by above is the following

  The code tell us us it starts at $i = 0$ with i increasing by 1 per iteration until the terminating condition of $i^2 \geq n$.

  And from the inequality, we know the loop finishes at $i = \sqrt{n} - 1$.

- I get the same feeling when solving question 4.b

b. *Proof.* **Part 1 (Determining the upper bound worst case running time):**

Let $n \in \mathbb{N}$.

First, we need to analyze the number of iterations in loop 2 per iteration in loop 1.

The code tells us loop 2 starts at $j = 0$ and ends at $j = n - 1$ with j increasing by 1 per iteration.

Using these facts, the number of iterations in loop 2 is

$$\lceil n - 1 - (i + 1) - 1 \rceil = n - i - 1 \tag{1}$$

Next, we need to add this number over all iterations in loop 1 to calculate the total number of loop 2.

Because we know the if condition **if lst[i] $>$ 1** will be satisfied at most n times, from $i = 0$ to $i = n - 1$ with i increasing by 1 per iteration, we can conclude the total number of iterations of loop 2 is at most

$$\begin{align} (n - 0 - 1) + (n - 1 - 1) + \cdots + (n - (n - 1) - 1) &= \sum_{i=0}^{n-1} i \tag{2} \\ &= \frac{n(n-1)}{2} \tag{3} \end{align}$$

Then, we can conclude the upper bound worst case running time is $\mathcal{O}(n^2)$.

Part 2 (Determining the lower bound worst case running time):

Let $n \in \mathbb{N}$, and $lst = [4n + 0, 4n + 1, 4n + 2, \ldots, 4n + (n - 1)]$.

First, we need to analyze the number of iterations in loop 2 per iteration in loop 1.

The code tells us loop 2 starts at $j = i + 1$ and ends at $j = n - 1$ with j increasing by 1 per iteration.

Using these facts, we can calculate that the number of iterations in loop 2 is

$$\lceil n - 1 - (i + 1) + 1 \rceil = n - i - 1 \tag{4}$$

Next, we need to evaluate the total number of iterations of loop 2 over loop 1.

Because we know the if condition if **if list[i] $>$ i** is true for all i (i.e. $i = 0, 1, 2, 3, \ldots, n-1$), we can conclude the total number of iterations of loop 2 is

$$\begin{align} (n - 0 - 1) + (n - 1 - 1) + \cdots + (n - (n - 1) - 1) &= \sum_{i=0}^{n-1} i \tag{5} \\ &= \frac{n(n-1)}{2} \tag{6} \end{align}$$

Then, we can conclude the lower bound worst-case running time of algorithm is $\Omega(n^2)$.

Furthermore, since the value in $\mathcal{O}$ and Ω are the same, $\Theta(n^2)$ is also true.

$\square$