Midterm 2 Version 1 Review

July 18, 2020

1. a) 1100100

   b) $-\sum_{i=0}^{n-1} 3^i$

   <u>Notes:</u>

   - Balanced Ternary
     - is a way of representing numbers
     - balanced ternary is in base 3, and has values 1,0 or -1

$$\sum_{i=0}^{n-1} d_i \cdot 3^i \text{ where } d_i \in \{0, 1, -1\} \tag{1}$$

   c) i. <u>$f(n) \in \Omega(n)$</u>

      True (since $n^2 + 10n + 2 \geq cn$)

      ii. <u>$g(n) \in \Omega(n)$</u>

      False (Let $c = 100, n_0 = 100$. Then $100 \log_2 n < 100n$)

      iii. <u>$f(n) \in \mathcal{O}(g(n))$</u>

      False ($f(n) = n^2 + 10n + 2$ grows faster than $g(n) = 100 \log_2 n$)

      iv. <u>$f(n) \in \Theta(g(n))$</u>

      True (Set $c_1 = -1, c_2 = 1, n_1 = 100$. Then $c_1 f(n) \leq g(n) \leq c_2 f(n)$)

      v. <u>$g(n) \in \Theta(\log_3 n)$</u>

      True (set $c_1 = -1, c_2 = 1, n_1 = 2$. Then $c_1 g(n) \leq \log_3 n \leq c_2 g(n)$)

vi. $\underline{g(n) \in \Theta(\log_3 n)}$

False (set $c_1 = -1, c_2 = 1, n_1 = 2$. Then $c_1 g(n) \leq \log_3 n \leq c_2 g(n)$)

vii. $\underline{f(n) + g(n) \in \Theta(f(n))}$

True (set $c_1 = -2, c_2 = 2, n_1 = 1$. Then $c_1(f(n) + g(n)) \leq f(n) \leq c_2(f(n) + g(n))$)

Notes:

- $g \in \Omega(f):\ \exists c, n_o \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \geq cf(n)$, where $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$
- $g \in \mathcal{O}(f):\ \exists c, n_o \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \leq cf(n)$, where $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$
- $g \in \Theta(f):\ g \in \mathcal{O}(f) \wedge g \in \Omega(f)$
  or
  $g \in \Theta(f):\ \exists c_1, c_2, n_1 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_1 \Rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n)$, where $f, g:\ \mathbb{N} \to \mathbb{R}^{\geq 0}$

d) $i = 3^{2^k}$

Since

k	0	1	2
i	3	9	81
	3^1	3^2	3^4

e) $k = \lceil \log_3(\log_2 n) - 1 \rceil$

Since

$$
\begin{aligned}
i^2 &\geq n && (1)\\
3^{2^k} &\geq n^{1/2} && (2)\\
2^k &\geq \log_3(n^{1/2}) && (3)\\
2^k &\geq (1/2)\log_3(n) && (4)\\
k &\geq \log_2((1/2)\log_3(n)) && (5)\\
&\geq \log_2(\log_3(n)) - 1 && (6)
\end{aligned}
$$

which gives $k = \lceil \log_2(\log_3(n)) - 1 \rceil$

2. Let $n \in \mathbb{N}$. Assume $n \geq 3$.

   I will prove $5^n + 50 < 6^n$ by induction.

   Base Step ($n = 3$):

Let $n = 3$.

Then,

$$5^3 + 50 = 715 < 6^3 = 216 \tag{1}$$

So, the base case holds.

Inductive Step

Let $n \in \mathbb{N}$. Assume $(5^n + 50 < 6^n)$.

I need to show $5^{n+1} + 50 < 6^{n+1}$.

Indeed we have

$$\begin{aligned} 5^{n+1} + 50 &= 5^n 5 + 50 && (2)\\ &= 5(5^n + 10) && (3)\\ &< 5(5^n + 50) && (4)\\ &< 56^n && (5)\\ &< 66^n && (6)\\ &< 6^{n=1} && (7) \end{aligned}$$

3. **Negation(expanded):** $\forall a \in \mathbb{R}, \forall c_1, c_2, n_1 \in \mathbb{R}^+, \exists n \in \mathbb{N}, (n \geq n_1) \wedge (c_1 g(n) > f(n)) \vee (f(n) > c_2 g(n))$

   Proof. Let $a \in \mathbb{R}$.

   I need to show $an + 1 \notin \Theta(n^3)$. That is, $an + 1 \notin \mathcal{O}(n^3) \vee an + 1 \notin \Omega(n^3)$. In other words, $\forall c, n_0 \in \mathbb{R}^+, \exists n \in \mathbb{N}, (n \geq n_0) \wedge (an + 1 > c \cdot n^3)$ or $\forall c_1, n_1 \in \mathbb{R}^+, \exists n \in \mathbb{N}, (n \geq n_1) \wedge (an + 1 < c_1 \cdot n^3)$.

   Let $c_1, c_2, n_1 \in \mathbb{R}^+$, and let $n = \lceil max(n_1, \sqrt{\frac{2a}{c_1}}, \sqrt[3]{\frac{2}{c_1}}) \rceil + 1$.

   Then, we can write

$$n = \lceil max(n_1, \sqrt{\frac{2a}{c_1}}, \sqrt[3]{\frac{2}{c_1}})) \rceil + 1 > \sqrt{\frac{2a}{c_1}} \quad (1)$$
$$n^2 > \frac{2a}{c_1} \quad (2)$$
$$\frac{c_1 n^3}{2} > an \quad (3)$$

And

$$n = \lceil max(n_1, \sqrt{\frac{2a}{c_1}}, \sqrt[3]{\frac{2}{c_1}})) \rceil + 1 > \sqrt[3]{\frac{2}{c_1}} \quad (4)$$
$$\frac{c_1 n^3}{2} > 1 \quad (5)$$

Thus, we can conclude

$$\frac{c_1 n^3}{2} + \frac{c_1 n^3}{2} > an + 1 \quad (6)$$
$$c_1 \cdot n^3 > an + 1 \quad (7)$$

$\square$

Notes:

- $g \in \Omega(f): \exists c, n_o \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow g(n) \geq cf(n)$, where $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$
- $g \in \mathcal{O}(f): \exists c, n_o \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow g(n) \leq cf(n)$, where $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$
- $g \in \Theta(f): g \in \mathcal{O}(f) \land g \in \Omega(f)$
  or
  $g \in \Theta(f): \exists c_1, c_2, n_1 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_1 \Rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n)$, where $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$

4. a) I need to evaluate the total number of iterations of loop 2.

   First, I need to evaluate the number of iterations of loop 2 per iteration of loop 1.

   The code tells us that the value of j increases by 3 per iteration k. That is, $j = 3k$.

Since the inner loop ends when $j \geq i$, the earliest iteration at which the loop ends per iteration of the outerloop is $k = \lceil \frac{i}{3} \rceil$.

Finally, the outer loop starts from $i = 0$ to $i = n^2$.

Thus, the total number of iterations in loop 2 is:

$$\sum_{i=0}^{n^2} \frac{i}{3} = \frac{n^2(n^2+1)}{6} \tag{1}$$

Correct Solution:

I need to evaluate the total number of iterations of loop 2.

First, I need to evaluate the number of iterations of loop 2 per iteration of loop 1.

The code tells us that the value of j increases by 3 per iteration k. That is, $j = 3k$.

Since the inner loop ends when $j \geq i$, the earliest iteration at which the loop ends per iteration of the outerloop is $k = \lceil \frac{i}{3} \rceil$.

Finally, the outer loop starts from $i = 0$ to $i = n^2$.

Thus, the total number of iterations in loop 2 is:

$$\sum_{i=0}^{n^2-1} \frac{i}{3} = \frac{(n^2-1)n^2}{6} \tag{1}$$

b) **Finding upperbound:**

Let $n \in \mathbb{N}$, and **lst** be arbitrary list of integers of length n.

The code tells us the worst case in loop occurs when there are odd numbers or no odd numbers at all.

In both of the cases, the loop runs from $i = 0$ to $i = n$.

So, the upperbound of my_alg is $\mathcal{O}(n)$

Finding worst-case lowerbound:

Let $n \in \mathbb{N}$, and let $n = [1, 2, 3, ..., n]$.

Then, at $n[i] = 3$, the **if** statement will occur.

Then, the inner loop will run from $i + 1$ to n, causing the rest of elements in **lst** to have even values, and the inner loop to have $n - (2 + 1) + 1 = n - 2$ iterations.

Then, the outer loop runs until $i = 3$ to $i = n$ without the **if** condition, resulting in the outloop to have n iterations.

Thus, the worst-case lower bound of my_alg is $\Omega(2n)$ or $\Omega(n)$.

Notes:

- Upperbound and lowerbound worstcase is determined by input :)

  Upperbound $\rightarrow$ arbitrary input

  Lowerbound $\rightarrow$ not arbitrary, but produces wost case values

  i.e. $[1, 2, 3, 4, ..., n]$