CSC236 Worksheet 6 Solution

Hyungmo Gu

May 9, 2020

Question 1

- *Proof.* Assume that for all $k \in \mathbb{N}$, $R(3^k) = k3^k$.

  I need to prove $R \in \mathcal{O}(n \lg n)$ and $R \in \Omega(n \lg n)$.

  I will do so in parts.

  Part 1 (Proving $R \in \mathcal{O}(n \lg n)$):

  Define $n^* = 3^{\lceil \log_3 n \rceil}$. Then, we have,

$$\lceil \log_3 n \rceil - 1 < \log_3 n \leq \lceil \log_3 n \rceil \Rightarrow n^*/3 < n \leq n^* \tag{1}$$

  I will also use the assumption (proved last week) that R is non-decreasing.

  Let $d = 6$. Then $d \in \mathbb{R}^+$. Let $B = 3$. Then $B \in \mathbb{N}^+$. Let n be an arbitrary natural number no smaller than B. Then,

$$\begin{aligned}
R(n) &\leq R(n^*) && \text{[Since } n < n^*\text{, and } R \text{ is non-decreasing]} && (2)\\
&= n^* \log_3 n^* && \text{[By assumption, and replacing } n^* \text{ for } 3^k\text{]} && (3)\\
&\leq 3n \log_3 3n && \text{[Since } n \leq n^* \Rightarrow 3n \leq 3n^*\text{]} && (4)\\
&\leq 3n(\log_3 n + 1) && && (5)\\
&\leq 3n(\log_3 n + \log_3 n) && \text{[Since } n \geq 3 \Rightarrow \log_3 n \geq 1\text{]} && (6)\\
&= 6n \log_3 n && && (7)\\
&\leq (6n \lg n)/\lg 3 && \text{[By change of basis to lg]} && (8)\\
&< 6n \lg n && && (9)\\
&= dn \lg n && \text{[Since } d = 6\text{]} && (10)
\end{aligned}$$

So $R \in \mathcal{O}(n \lg n)$, since $\log_3 n$ differs from $\lg n$ by a constant factor.

Part 2 (Proving $R \in \Omega(n \lg n)$):

Define $n^* = 3^{\lceil \log_3 n \rceil}$. Then, we have,

$$\lceil \log_3 n \rceil - 1 < \log_3 n \leq \lceil \log_3 n \rceil \Rightarrow n^*/3 < n \leq n^* \tag{11}$$

I will also use the assumption (proved last week) that R is non-decreasing.

Let $d = 1/(6 \lg 3)$. Then $d \in \mathbb{R}^+$. Let $B = 9$. Then $B \in \mathbb{N}^+$. Let n be an arbitrary natural number no smaller than B. Then,

$$\begin{aligned}
R(n) &\geq R(n^*/3) && \text{[Since } n^*/3 < n \text{, and } R \text{ is non-decreasing]} && (12)\\
&= (n^*/3) \cdot \log_3(n^*/3) && \text{[By assumption, and replacing } n^* \text{ for } 3^k\text{]} && (13)\\
&\geq (n/3) \cdot \log_3(n/3) && \text{[Since } n^* \leq n \Rightarrow n^*/3 \leq n/3\text{]} && (14)\\
&= (n/3) \cdot (\log_3 n - 1) && && (15)\\
&\geq (n/3) \cdot (\log_3 n - (\log_3 n)/2) && \text{[Since } n \geq 9 \Rightarrow (\log_3 n)/2 \geq 1\text{]} && (16)\\
&= (n/6) \cdot \log_3 n && && (17)\\
&= (n/6) \cdot (\lg n / \lg 3) && && (18)\\
&= (n/(6 \lg 3)) \cdot \lg n && && (19)\\
&= dn \cdot \lg n && \text{[Since } d = 1/(6 \lg 3)\text{]} && (20)
\end{aligned}$$

So, $R \in \Omega(n \lg n)$.

$\square$

Correct Solution:

Assume that for all $k \in \mathbb{N}$, $R(3^k) = k3^k$.

I need to prove $R \in \mathcal{O}(n \lg n)$ and $R \in \Omega(n \lg n)$.

I will do so in parts.

Part 1 (Proving $R \in \mathcal{O}(n \lg n)$):

Define $n^* = 3^{\lceil \log_3 n \rceil}$. Then, we have,

$$\lceil \log_3 n \rceil - 1 < \log_3 n \leq \lceil \log_3 n \rceil \Rightarrow n^*/3 < n \leq n^* \tag{1}$$

I will also use the assumption (proved last week) that R is non-decreasing.

Let $d = 6$. Then $d \in \mathbb{R}^+$. Let $B = 3$. Then $B \in \mathbb{N}^+$. Let n be an arbitrary natural number no smaller than B. Then,

$$\begin{aligned}
R(n) &\leq R(n^*) && \text{[Since } n < n^*\text{, and } R \text{ is non-decreasing]} && (2)\\
&= n^* \log_3 n^* && \text{[By assumption, and replacing } n^* \text{ for } 3^k\text{]} && (3)\\
&\leq 3n \log_3 3n && \text{[Since } n \leq n^* \Rightarrow 3n \leq 3n^*\text{]} && (4)\\
&= 3n(\log_3 n + 1) && && (5)\\
&\leq 3n(\log_3 n + \log_3 n) && \text{[Since } n \geq 3 \Rightarrow \log_3 n \geq 1\text{]} && (6)\\
&= 6n \log_3 n && && (7)\\
&= (6n \lg n)/\lg 3 && \text{[By change of basis to lg]} && (8)\\
&< 6n \lg n && && (9)\\
&= dn \lg n && \text{[Since } d = 6\text{]} && (10)
\end{aligned}$$

So $R \in \mathcal{O}(n \lg n)$, since $\log_3 n$ differs from $\lg n$ by a constant factor.

Part 2 (Proving $R \in \Omega(n \lg n)$):

Define $n^* = 3^{\lceil \log_3 n \rceil}$. Then, we have,

$$\lceil \log_3 n \rceil - 1 < \log_3 n \leq \lceil \log_3 n \rceil \Rightarrow n^*/3 < n \leq n^* \qquad (11)$$

I will also use the assumption (proved last week) that R is non-decreasing.

Let $d = 1/(6 \lg 3)$. Then $d \in \mathbb{R}^+$. Let $B = 9$. Then $B \in \mathbb{N}^+$. Let n be an arbitrary natural number no smaller than B. Then,

$$
\begin{aligned}
R(n) &\geq R(n^*/3) && \text{[Since } n^*/3 < n\text{, and } R \text{ is non-decreasing]} \quad (12)\\
&= (n^*/3) \cdot \log_3(n^*/3) && \text{[By assumption, and replacing } n^* \text{ for } 3^k\text{]} \quad (13)\\
&\geq (n/3) \cdot \log_3(n/3) && \text{[Since } n^* \leq n \Rightarrow n^*/3 \leq n/3\text{]} \quad (14)\\
&= (n/3) \cdot (\log_3 n - 1) && (15)\\
&\geq (n/3) \cdot (\log_3 n - (\log_3 n)/2) && \text{[Since } n \geq 9 \Rightarrow (\log_3 n)/2 \geq 1\text{]} \quad (16)\\
&= (n/6) \cdot \log_3 n && (17)\\
&= (n/6) \cdot (\lg n / \lg 3) && (18)\\
&= (n/(6 \lg 3)) \cdot \lg n && (19)\\
&= dn \cdot \lg n && \text{[Since } d = 1/(6 \lg 3)\text{]} \quad (20)
\end{aligned}
$$

So, $R \in \Omega(n \lg n)$, since $\log_3 n$ differs from $\lg n$ by a constant factor.

Notes:

- Learned that if there is trouble going from $\log_3 n - 1$ to $dn \lg n$, a good approach is to increase the value of B.
- Noticed that professor used 'Let $d =$ ____. Then $d \in \mathbb{R}^+$' to define variable's value as well as its type.
- $g \in \Theta(f) :\ g \in \mathcal{O}(f) \land g \in \Omega(f)$

  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 $f, g :\ \mathbb{N} \to \mathbb{R}^{\geq 0}$
- $g \in \Omega(f) :\ \exists c, n_o \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \geq cf(n)$, 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 $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$

Question 2

- *Proof.* Let $n \in \mathbb{N}$ and bits $b_0, \cdots, b_k \in \{0, 1\}$ be such that $n = \sum_{i=0}^{i=k} 2^i b_i$. I will use identities:

$$\lfloor n/2 \rfloor = \sum_{i=1}^{i=k} 2^i b_i \tag{1}$$
$$b_0 = n \mod 2 \tag{2}$$

Define $P(n)$: "If n is a natural number, then *decimal_to_binary(n)*" terminates and returns binary string representing n with no leading zeros, except if n is 0.

I will use complete induction to prove $\forall n \in \mathbb{N}, P(n)$.

Inductive Step

Let $n \in \mathbb{N}$. Assume $H(n) : \bigwedge_{i=0}^{i=n-1} P(i)$. I will show $P(n)$ follows.

Base Case ($n < 2$):

Let $n < 2$.

Then, the if part of *decimal_to_binary(n)* executes, and the program terminates by outputing the input value.

Since, binary rep of $n = 0$ is 0, and binary rep of $n = 1$ is 1, $P(n)$ follows in this step.

Case ($n \geq 2$):

Let $n \geq 2$.

Then, since $n \geq 2$, and $0 \leq n\%2 \leq n//2 < n$, the induction hypothesis tells us $P(n//2)$ and $P(n \,\%\, 2)$ holds.

Then, since $n//2$ is $\lfloor n/2 \rfloor = \sum_{i=1}^{i=k} 2^i b_i$, and $n \,\%\, 2$ is $n \mod 2 = b_0$, we have

$$b_0 + \lfloor n/2 \rfloor = b_0 + \sum_{i=1}^{i=k} 2^i b_i \tag{3}$$
$$= 2^0 \cdot b_0 + \sum_{i=1}^{i=k} 2^i b_i \tag{4}$$
$$= \sum_{i=0}^{i=k} 2^i b_i \tag{5}$$
$$= n \tag{6}$$

Thus, $P(n)$ follows from $H(n)$ in this step.

$\square$

Notes:

- Noticed professor used 'I will use identities: ...' to import discovered properties and rules from previous exercises and lectures.

  > I will use identities:
  > $$\lfloor n/2 \rfloor = \sum_{i=1}^{i=k} 2^i b_i \tag{1}$$
  > $$b_0 = n \mod 2 \tag{2}$$

- Noticed that in $P(n)$

  "*n is a natural number*" - is precondition

  > Define $P(n)$: "If n is a natural number ...

  "*returns binary string representing n with no leading zeros, except if n is 0*" - is postcondition.

  > Define $P(n)$: "... then *decimal_to_binary(n)*" terminates and returns binary string representing n with no leading zeros, except if n is 0.

- **Correct:** A program is correct if it produces output on every acceptable input
- **Precondition** and **Postcondition** are assertions involving some of the variables of the program
  - **Precondition** states what must be true *before* program starts execution
  - **Postcondition** states what must be true when the program *ends*