

Worksheet 16 Review

April 3, 2020

Question 1

a. Let $k \in \mathbb{N}$.

Here, the minimum possible change occurs for the loop variable in a single iteration when $i = i + 1$.

The maximum possible change occurs for the loop variable in a single iteration when $i = i + 6$.

The exact upper bound of the variable after k iteration is

$$i_k \leq 6k \tag{1}$$

The exact lower bound of the variable after k iteration is

$$k \leq i_k \tag{2}$$

Using the fact that the termination occurs when $i_k = n$, we can calculate that for the upper bound, the loop terminates when

$$6k \geq n \quad (3)$$

$$k \geq \frac{n}{6} \quad (4)$$

Because we know $\frac{n}{6}$ may be a decimal, we can conclude the closest value at which the loop terminates is when

$$k = \left\lceil \frac{n}{6} \right\rceil \quad (5)$$

Using the same fact, we can calculate that for the lower bound, the loop terminates when

$$k \geq n \quad (6)$$

It follows from above that for the lower bound, the smallest value of k at which the loop termination occurs is when

$$k = n \quad (7)$$

Then, we can conclude the function has asymptotic lower bound of $\Omega(n)$, and asymptotic upper bound of $\mathcal{O}(n)$.

Then, since both Ω and \mathcal{O} have the same value, $\Theta(n)$ is also true.

Correct Solution:

Here, the minimum possible change occurs for the loop variable in a single iteration when $i = i + 1$.

The maximum possible change occurs for the loop variable in a single iteration when $i = i + 6$.

The exact upper bound of the variable after k iteration is

$$i_k \leq 6k \quad (8)$$

The exact lower bound of the variable after k iteration is

$$k \leq i_k \quad (9)$$

Using the fact that the termination occurs when $i_k = n$, we can calculate that for the upper bound, the loop terminates when

$$6k \geq n \quad (10)$$

$$k \geq \frac{n}{6} \quad (11)$$

Because we know $\frac{n}{6}$ may be a decimal, we can conclude the closest value at which the loop terminates is when

$$k = \left\lceil \frac{n}{6} \right\rceil + 1 \quad (12)$$

Using the same fact, we can calculate that for the lower bound, the loop terminates when

$$k \geq n \quad (13)$$

It follows from above that for the lower bound, the smallest value of k at which the loop termination occurs is when

$$k = n + 1 \tag{14}$$

Then, we can conclude the function has asymptotic lower bound of $\Omega(n)$, and asymptotic upper bound of $\mathcal{O}(n)$.

Since both Ω and \mathcal{O} have the same value, $\Theta(n)$ is also true.

Notes:

- where is $+1$ coming from? Is it coming from the loop variable $i = 0$?

b. Let $k \in \mathbb{N}$.

Part 1 (Determining maximum and minimum possible change in a single iteration):

It follows from observation that the minimum possible change occurs when $i = i \cdot 2$, and the maximum possible change when $i = i \cdot 3$.

Part 2 (Determining lower bound and upper bound of loop iteration):

Because we know the smallest possible change occurs when $i = i \cdot 2$ occurs repeatedly, we can conclude that at k^{th} iteration i_k has the lower bound of 2^k .

Similarly, because we know largest possible change occurs when $i = i \cdot 3$ occurs repeatedly, we can conclude that at k^{th} iteration, i_k has the upper bound of 3^k .

Then, by putting together, we can conclude that

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

Part 3 (Determining exact number of iterations for the lower bound and upper bound):

Because we know the loop runs until $i_k < n$, we can conclude that at lower bound, termination occurs when

$$i_k \geq n \quad (2)$$

$$2^k \geq n \quad (3)$$

$$\log_2 2^k \geq \log_2 n \quad (4)$$

$$k \geq \log_2 n \quad (5)$$

Using the fact that we are looking for smallest value of k , we can calculate that for lower bound

$$k = \lceil \log_2 n \rceil + 1 \quad (6)$$

Similarly, for the upper bound, loop terminates when

$$i_k \geq n \quad (7)$$

$$3^k \geq n \quad (8)$$

$$\log_3 3^k \geq \log_3 n \quad (9)$$

$$k \geq \log_3 n \quad (10)$$

Using the fact, we can calculate that for upper bound,

$$k = \lceil \log_3 n \rceil + 1 \quad (11)$$

Part 4 (Determining Big-Oh and Omega):

Because we know $\log_2 n$ dominates $\log_3 n$, we can conclude $\log_2 n$ is the asymptotic upper bound, and $\log_3 n$ is the asymptotic lower bound.

Then, we can conclude the algorithm has $\mathcal{O}(\log_2 n)$ and $\Omega(\log_3 n)$.

Notes:

- How come in solution, **+1** doesn't exist? What rules of thumb i can follow to better determine whether **+1** should be included?

Question 2

- a. Because we know $n \in \Theta(n^2)$, we can conclude the algorithm has runtime of $\Theta(n^2)$.

Correct Solution:

Since **helper1** has cost of n and **helper2** has cost of n^2 , we can conclude the algorithm has total cost of $n + n^2$.

It follows from above the algorithm has runtime of $\Theta(n^2)$.

Notes:

- When is \in in $n \in \Theta(n^2)$ used?

Is $\in \Theta$ used when \mathcal{O} and Ω exists with different values to choose which value works for both lower and upper bound of the algorithm?

- Noticed that professor evaluates total runtime before Theta

- b. Because we know loop 1 starts at $i = 0$ and finishes at $i = n - 1$ with i increasing by 2 per iteration, we can conclude loop 1 has

$$\left\lceil \frac{n - 1 - 0 + 1}{2} \right\rceil = \left\lceil \frac{n}{2} \right\rceil \quad (1)$$

iterations.

Since each iteration in loop 1 takes n step, as required by **helper 1** function, we can conclude loop 1 has total cost of

$$n \cdot \left\lceil \frac{n}{2} \right\rceil \quad (2)$$

steps.

For loop 2, because we know it starts at $j = 0$ and finishes at $j = 9$, we can conclude loop 2 has

$$\lceil 9 - 0 + 1 \rceil = 10 \quad (3)$$

iterations.

Since each iteration in loop 2 takes n^2 step as required by **helper 2** function, we can conclude loop 2 has total of

$$10 \cdot n^2 \quad (4)$$

steps.

Since $i = 0$ and $j = 0$ have cost of 1 step each, the total cost of algorithm is

$$n \cdot \left\lceil \frac{n}{2} \right\rceil + 10n^2 + 2 \quad (5)$$

Then, we can conclude the algorithm has running time of $\Theta(n^2)$

Correct Solution:

Because we know loop 1 starts at $i = 0$ and finishes at $i = n - 1$ with i increasing by 2 per iteration, we can conclude loop 1 has

$$\left\lceil \frac{n - 1 - 0 + 1}{2} \right\rceil = \left\lceil \frac{n}{2} \right\rceil \quad (1)$$

iterations.

Since each iteration in loop 1 takes n step, as required by **helper 1** function, we can conclude loop 1 has total cost of

$$n \cdot \left\lceil \frac{n}{2} \right\rceil \quad (2)$$

steps.

For loop 2, because we know it starts at $j = 0$ and finishes at $j = 9$, we can conclude loop 2 has

$$\lceil 9 - 0 + 1 \rceil = 10 \quad (3)$$

iterations.

Since each iteration in loop 2 takes n^2 step as required by **helper 2** function, we can conclude loop 2 has total of

$$10 \cdot n^2 \quad (4)$$

steps.

Combining together , the total cost of algorithm is

$$n \cdot \left\lceil \frac{n}{2} \right\rceil + 10n^2 \quad (5)$$

Then, we can conclude the algorithm has running time of $\Theta(n^2)$

Notes:

- Noticed professor doesn't count loop variables toward the total cost of algorithm.

If other lines such as **return False** and **n = len(lst)** are included, would these count towards the total cost of the algorithm?

- c. For loop 1, because we know it starts at $i = 0$ and finishes at $i = n - 1$ with each iteration having cost of i steps, we can conclude loop 1 has cost of

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

steps.

For loop 2, because we know it starts at $j = 0$ and finishes at $j = 9$ with each iteration costing j^2 steps, we can conclude loop 2 has

$$\sum_{j=0}^9 j^2 = \frac{9(9-1)(2(9)-1)}{6} \quad (2)$$

$$= \frac{9 \cdot 8 \cdot 17}{6} \quad (3)$$

$$= 204 \quad (4)$$

steps.

Combining together, the total cost of algorithm is

$$\frac{n(n-1)}{2} + 204 \quad (5)$$

steps.

Then, we can conclude the running time of algorithm is $\Theta(n^2)$.

Question 3

- a. **Predicate Logic:** $\forall x \in \mathbb{Z}^+, k \in \mathbb{N}, (\text{Three iterations occur}) \Rightarrow x_{3(k+1)} \leq \frac{x_{3k}}{2}$

Assume three iterations occur in loop.

We will prove the statement by cases.

Case 1 (x divisible by 2 three times):

Proof. Let $k \in \mathbb{N}$.

Assume x is divisible by 2 at least 3 times.

Then, because we know $x \bmod 2 == 0$ in all iterations, we can conclude the line $\mathbf{x} = \mathbf{x} // \mathbf{2}$ will run 3 times.

Then, it follows from above that the value of x at $3(k+1)^{th}$ iteration is

$$x_{3(k+1)} = \frac{x_{3k}}{2^3} \quad (6)$$

Then, because we know $\frac{1}{2^3} \leq \frac{1}{2}$, we can conclude

$$x_{3(k+1)} \leq \frac{1}{2}x_{3k} \quad (7)$$

□

Case 2 (x divisible by 2 two times):

Case 3 (x divisible by 2 once):

Case 4 (x is an odd number):

b. Let $k \in \mathbb{N}$.

We need to find the smallest value of k in terms of n at which the loop termination occurs.

The result in previous problem tells us that every 3 iterations, x decreases by half. So, initially we have

$$x_3 \leq \frac{x_0}{2} \quad (1)$$

Then, we can conclude that at $3k$ iterations,

$$x_3 \leq \frac{x_0}{2^k} \quad (2)$$

Then, since $x_0 = n$,

$$x_3 \leq \frac{n}{2^k} \quad (3)$$

Using the fact that the loop termination occurs when $x \leq 1$, we can calculate

$$2^k \geq n \tag{4}$$

$$k \geq \log n \tag{5}$$

Then, it follows from above the smallest value of k at which termination occurs is

$$\lceil \log n \rceil \tag{6}$$

Then, we can conclude the running time of algorithm is $\Theta(\log n)$