Worksheet 16 Solution

March 29, 2020

Question 1

a. **Part 1.a - Finding minimum possible change for a loop in a single iteration**

The minimum possible change in a loop occurs when i increments by 1.

Part 1.b - Finding maximum possible change for a loop in a single iteration

The maximum possible change in a loop occurs when i increments by 6.

Part 2.a - Determine formula for an exact lower bound on the value

Since the loop starts at $i = 0$ and ends at $n - 1$, the loop has

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

iterations.

Since the smallest step increases by 1 per iteration, the total cost of the loop at minimum possible change is

$$(n) \cdot 1 = n \tag{2}$$

steps.

Part 2.a - Determine formula for an exact upper bound on the value

Since the loop starts at $i = 0$ and ends at $n - 1$, the loop has

$$n - 1 + 1 = n \tag{3}$$

iterations.

Part 2.b - Determine formula for an exact lower bound on the value

Since the largest step increases by 6 per iteration, the total cost of the loop at minimum possible change is

$$\left\lceil \frac{n}{6} \right\rceil \tag{4}$$

steps.

Part 3.a - Determine formula for an exact upper bound on the value Is it n?

Part 3.a - Determine formula for an exact upper bound on the value Is it $\lceil \frac{n}{6} \rceil$?

Part 4 - Determine Big Oh and Big Omega

The big Oh bound of running time is $\mathcal{O}(n)$, and the big theta of running time is $\Omega(n)$.

Since n in $\mathcal{O}(n)$ and $\Omega(n)$ are the same, $\Theta(n)$ is also true.

<u>**Correct Solution:**</u>

Part 1.a - Finding minimum possible change for a loop in a single iteration

The minimum possible change in a loop occurs when i increments by 1.

**Part 1.b - Finding maximum possible change for a loop in a

single iteration

The maximum possible change in a loop occurs when i increments by 6.

Part 2.a - Determine formula for an exact upper bound on the value

The upper bound of loop termination is when $k \geq n$

Part 2.b - Determine formula for an exact lower bound on the value

The lower bound of loop termination is when $6k \leq n$

Part 3.a - Use the formula to determine the exact number of loops that will occur for upper bound

Since the loop starts from 0 and ends at $n-1$, the loop has total of

$$n - 1 - 0 + 1 = n \tag{5}$$

iterations.

Since 1 step is taken for each iteration, the upper bound total cost of loop iteration is

$$n \cdot 1 = n \tag{6}$$

Since the statement on line 2 has cost of 1, the upper bound total cost of the algorithm is $n + 1$, or $\mathcal{O}(n)$.

Part 3.b - Use the formula to determine the exact number of loops that will occur for lower bound

Since the loop starts from 0 and ends at $n-1$, the loop has total of

$$n - 1 - 0 + 1 = n \tag{7}$$

iterations.

Since 6 steps are taken for each iteration, the lower bound total cost of loop iteration is

$$\left\lceil \frac{n}{6} \right\rceil \tag{8}$$

Since the statement on line 2 has cost of 1, the lower bound total cost of the algorithm is $\left\lceil \frac{n}{6} \right\rceil + 1$, or $\Omega(n)$

Part 4 - Determine Big Oh and Big Omega

The big Oh bound of running time is $\mathcal{O}(n)$, and the big theta of running time is $\Omega(n)$.

Since n in $\mathcal{O}(n)$ and $\Omega(n)$ are the same, $\Theta(n)$ is also true.

b. **Part 1.a - Finding minimum possible change for a loop in a single iteration**

The minimum possible change for a look in a single iteration is when i increases by a factor of 2

Part 1.b - Finding maximum possible change for a loop in a single iteration

The maximum possible change for a look in a single iteration is when i increases by a factor of 3

Part 2.a - Determine formula for an exact upper bound of the loop variable after k iterations

The exact upper bound of the loop variable after k iteration is $2^k \geq n$

Part 2.b - Determine formula for an exact lower bound of the loop variable after k iterations

The exact lower bound of the loop variable after k iteration is $3^k \geq n$

Part 3.a - Use the formula to determine the exact number of loops that will occur for upper bound

The upper bound of loop iteration is $\lceil \log n \rceil$, or $\mathcal{O}(\log n)$

Part 3.b - Use the formula to determine the exact number of loops that will occur for lower bound

The lower bound of loop iteration is $\lceil \log_3 n \rceil$, or $\Omega(\log n)$

Part 4 - Determine Big Oh and Big Omega

For the upper bound, we have $\mathcal{O}(\log n)$.

For the lower bound, we have $\Omega(\log n)$

Since Big Oh and Big Omega have the same value, $\Theta(\log n)$ is also true.

Question 2

a. Since **helper1** has cost of n steps, and **helper2** has cost of n^2 steps, the algorithm has total run time of $n^2 + n$ steps, or $\Theta(n^2)$

Question 3