CS240, Spring 2022
Assignment 1: Question 5

Q5a) Analyse the following pieces of pseudocode, and give a tight bound of the running time.

To begin, lets look at the time complexity of the content the inner most loop (Line 4). We are doing a linear amount of work, which we will represent by the constant c. Thus:

$$\text{Inner loop conent's time complexity } = c$$

The rest of the code can now be converted into a series of summations:

$$= \sum_{i=1}^{n} \sum_{j=1}^{i^2} \sum_{k=1}^{\log(n)} (\text{Inner loop conent})$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{i^2} \sum_{k=1}^{\log(n)} c$$

Using algebra we will simplify these summations:

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{i^2} \sum_{k=1}^{\log(n)} c &= \sum_{i=1}^{n} \sum_{j=1}^{i^2} c\log(n) \\ &= \sum_{i=1}^{n} c\log(n)(i^2) \\ &= c\log(n)\sum_{i=1}^{n}(i^2) \end{aligned}$$

Since $c\log(n)$ is not in terms of i we can separate it out of the summation, therefore our equations becomes:

$$\begin{aligned} c\log(n)\sum_{i=1}^{n}(i^2) &= c\log(n)\frac{n(n+1)(n+2)}{6} \\ &= c\log(n)\frac{n(2n^2+3n+1)}{6} \\ &= c\log(n)\frac{2n^3+3n^2+n}{6} \\ &= \frac{2c}{6}n^3\log(n) + \frac{3c}{6}n^2\log(n) + \frac{c}{6}n\log(n) \end{aligned}$$

Since we maintained equality we can conclude that the code has a time complexity of:

$$\frac{2c}{6}n^3\log(n) + \frac{3c}{6}n^2\log(n) + \frac{c}{6}n\log(n) \in \boldsymbol{\Theta}(n^3\log(n))$$

Q5b) Analyse the following pieces of pseudocode, and give a tight bound of the running time.

To start, we will again look at the content of inner most loop (line 5). Since we are doing a linear amount of work, which we will represent by the constant c. Thus:

$$\text{Inner loop conent } = c_1$$

Therefore our first loop ($j = 1$ to n) will have a time complexity of:

$$\begin{aligned} &= \sum_{j=1}^{n} c_1 \\ &= c_1 n \end{aligned}$$

Each time the while loop iterates it will take $c_1 n$ k, plus the constant amount of time to square i and store it (line 6). We will represent this time taken to square i by the constant c_2. Thus the while loop has an inner time complexity of:

$$c_1 n + c_2$$

The while loop will run while $i < n$. We know that at the start of the while loop $i = 2$ and i will square itself each iteration. Therefore:

$$i = 2^{2^{\#\text{ of iterations}}}$$

And so the loop will run while:

$$\begin{aligned} 2^{2^{\#\text{ of iterations}}} &< n \\ 2^{\#\text{ of iterations}} &< \log(n) \\ \#\text{ of iterations} &< \log(\log(n)) \end{aligned}$$

Therefore the loop will terminate when $\#$ of iterations $\geq log(n)$. We will now move forward using the "sloppy method"

Solving for upper bound: The number of iterations before terminating will be upper bounded by $2\log(log(n))$. Thus our time complexity can be represented by:

$$\begin{aligned} \text{Code's time complexity} &\leq \sum_{x=1}^{2\log(\log(n))} c_1 n + c_2 \\ &\leq (c_1 n + c_2) 2\log(\log(n)) \\ &\leq 2c_1 n \log(\log(n)) + 2c_2 \log(\log(n)) \end{aligned}$$

And so our time complexity for the upper bound is:

$$2c_1 n \log(\log(n)) + 2c_2 \log(\log(n)) \in O(n \log(\log(n)))$$

Solving for lower bound: The number of iterations before terminating will be upper bounded by $\frac{\log(log(n))}{2}$. Thus our time complexity can be represented by:

$$\begin{aligned}
\text{Code's time complexity} &\geq \sum_{x=1}^{\frac{\log(log(n))}{2}} c_1 n + c_2 \\
&\geq (c_1 n + c_2)\frac{\log(log(n))}{2} \\
&\geq \frac{c_1 n \log(log(n))}{2} + \frac{c_2 \log(log(n))}{2}
\end{aligned}$$

And so our time complexity for the lower bound is:

$$\frac{c_1 n \log(log(n))}{2} + \frac{c_2 \log(log(n))}{2} \in \Omega(n \log(\log(n)))$$

Therefore since the upper bound and lower bound are the same, we can conclude:

$$\text{Code's time complexity } \in \boldsymbol{\Theta}(n \log(\log(n)))$$

Q5c) Analyse the following pieces of pseudocode, and give a tight bound of the running time.

To start, we will represent all lines where we do constant amounts of work (line 4, line 5-6, line 7) with the following constants c_1, c_2, c_3. Thus our time complexity for the content inside the inner while loop is:

$$\text{Inner while loop content } = c_2$$

The time complexity for the content inside the outer while loop is:

$$\text{Outer while loop content } = c_1 + \text{ Inner while loop } + c_3$$

Before we use the "sloppy" method, we will first try to represent the number of iterations for each while loop. Starting with the first while loop, we know that j is equal to:

$$j = n^3 - (\text{\# of inner loop iterations})$$

Therefore the inner while loop will run while:

$$\begin{aligned} j &> i \\ n^3 - i(\text{\# of inner loop iterations}) &> i \\ -i(\text{\# of inner loop iterations}) &> i - n^3 \\ \text{\# of inner loop iterations} &< \frac{n^3}{i} - 1 \end{aligned}$$

Solving for the outer while loop, we know that we can express i as:

$$i = 1 + 5(\text{\# of outer loop iterations})$$

Therefore the outer loop will run while:

$$\begin{aligned} i &< 5n \\ 1 + 5(\text{\# of outer loop iterations}) &< 5n \\ 5(\text{\# of outer loop iterations}) &< 5n - 1 \\ \text{\# of outer loop iterations} &< n - \frac{1}{5} \end{aligned}$$

Therefore the inner while loop will terminate when (# of inner loop iterations) $\geq n^3 - i$ and the outer while loop will terminate when (# of outer loop iterations) $\geq n - \frac{1}{5}$.

Solving for upper bound: The number of iterations before terminating the outer while loop will go through is upper bounded by $n+2$. Thus our time complexity can be represented by:

$$\text{Code's time complexity} \leq \sum_{x=1}^{n+2} \left(c_1 + \sum_{y=1}^{n^3-i} c_2 + c_3 \right)$$

However since x represents the number of outer loop iterations, we can express i as:

$$\begin{aligned} i &= 1 + 5(\text{\# of outer loop iterations}) \\ &= 1 + 5x \end{aligned}$$

Solving for the time complexity we get the following:

$$\begin{aligned}
&\leq \sum_{x=1}^{n+2} \left(c_1 + \sum_{y=1}^{\frac{n^3}{i}-1} c_2 + c_3 \right) \\
&\leq \sum_{x=1}^{n+2} \left(c_1 + \frac{n^3}{i} c_2 - c_2 - 5xc_2 + c_3 \right) \\
&\leq \sum_{x=1}^{n+2} \left(\frac{n^3}{i} c_2 \right) + \sum_{x=1}^{n+2} (-5xc_2) + \sum_{x=1}^{n+2} (c_1 + -c_2 + c_3) \\
&\leq n^3 c_2 \sum_{x=1}^{n+2} \left(\frac{1}{1+5x} \right) + -5c_2 \sum_{x=1}^{n+2} (x) + (c_1 + -c_2 + c_3)(n+2)
\end{aligned}$$

Note that $\sum_{x=1}^{n+2} \left(\frac{1}{1+5x}\right) \leq \sum_{x=1}^{n+2} \left(\frac{1}{x}\right)$ for all values of $n > 0$ so we can substitute using the equation found it the textbook, we will let c_4 represent $\gamma + o(1)$ and get:

$$\begin{aligned}
&\leq n^3 c_2 \sum_{x=1}^{n+2} \left(\frac{1}{x} \right) - 5c_2 \frac{(n+2)(n+3)}{2} + (c_1 - c_2 + c_3)(n+2) \\
&\leq n^3 c_2 (\log(n+2) + c_4) - 5c_2 \frac{n^2 + 5n + 6}{2} + (c_1 n - c_2 n + c_3 n) + (2c_1 - 2c_2 + 2c_3)
\end{aligned}$$

From this we can see that the upper bound on the time complexity is:

$$n^3 c_2 (\log(n+2) + c_4) - 5c_2 \frac{n^2 + 5n + 6}{2} + (c_1 n - c_2 n + c_3 n) + (2c_1 - 2c_2 + 2c_3 \in O(n^3 log(n))$$

Solving for lower bound: The number of iterations before terminating the outer while loop will go through is upper bounded by $n-2$. Thus our time complexity can be represented by:

$$\text{Code's time complexity} \geq \sum_{x=1}^{n-2} \left(c_1 + \sum_{y=1}^{\frac{n^3}{i}-1} c_2 + c_3 \right)$$

However since x represents the number of outer loop iterations, we can express i as:

$$\begin{aligned} i &= 1 + 5(\text{\# of outer loop iterations}) \\ &= 1 + 5x \end{aligned}$$

Solving for the time complexity we get the following:

$$\begin{aligned}
&\geq \sum_{x=1}^{n-2}\left(c_1 + \sum_{y=1}^{\frac{n^3}{i}-1} c_2 + c_3\right)\\
&\geq \sum_{x=1}^{n-2}\left(c_1 + \frac{n^3}{i}c_2 - c_2 - 5xc_2 + c_3\right)\\
&\geq \sum_{x=1}^{n-2}\left(\frac{n^3}{i}c_2\right) + \sum_{x=1}^{n-2}(-5xc_2) + \sum_{x=1}^{n-2}(c_1 + -c_2 + c_3)\\
&\geq n^3 c_2 \sum_{x=1}^{n-2}\left(\frac{1}{1+5x}\right) + -5c_2\sum_{x=1}^{n-2}(x) + (c_1 + -c_2 + c_3)(n-2)
\end{aligned}$$

Note that $\sum_{x=1}^{n+2}\left(\frac{1}{1+5x}\right) \geq \sum_{x=1}^{n+2}\left(\frac{1}{20x}\right)$ for all values of $n > 0$ so we can substitute using the equation found it the textbook, we will let c_4 represent $\gamma + o(1)$ and get:

$$\begin{aligned}
&\leq n^3 c_2 \sum_{x=1}^{n+2}\left(\frac{1}{20x}\right) - 5c_2\frac{(n+2)(n+3)}{2} + (c_1 - c_2 + c_3)(n-2)\\
&\leq n^3\frac{c_2}{20}(\log(n+2) + c_4) - 5c_2\frac{n^2+5n+6}{2} - (c_1 n - c_2 n + c_3 n) + (2c_1 - 2c_2 + 2c_3)
\end{aligned}$$

From this we can see that the lower bound on the time complexity is:

$$n^3\frac{c_2}{20}(\log(n+2) + c_4) - 5c_2\frac{n^2+5n+6}{2} - (c_1 n - c_2 n + c_3 n) + (2c_1 - 2c_2 + 2c_3) \in \Omega(n^3\log(n))$$

Therefore since the upper bound and lower bound are the same, we can conclude:

$$\text{Code's time complexity } \in \boldsymbol{\Theta}(n^3\log(n))$$