CS3383, Winter 2019 Assignment # 1 Sample solutions

Rasoul Shahsavarifar
Faculty of Computer Science, UNB

Question 1: (6 marks) Consider the given functions bellow. Sort all of them using the asymptotic order (big-O). Provide short explanation for your answer.

- $3 \log n$
- $3 \log \log n$
- $n^{\log n}$
- 5^n
- $n^{n^{1/4}}$
- $(\frac{n}{4})^{n/4}$

Short Answer: By doing pairwise comparisons for all the above relations using either the *"limit method"* or *"derivative method"*, the order among the functions would be obtained as:

$$3 \log \log n \leq 3 \log n \leq n^{\log n} \leq n^{n^{1/4}} \leq 5^n \leq (\frac{n}{4})^{n/4}$$

Question 2: (4 marks) Among the following given functions, which one(s) is (are) representing the time complexity of a sub-quadratic algorithm. Explain your answer, and give a polynomial as an example for each part.

- $O(n^{\frac{3}{2}})$
- $\Omega(n^{\frac{3}{2}})$
- $n^{O(\frac{3}{2})}$
- $n^{\Omega(\frac{3}{2})}$

Short Answer: For this question you basically need to compare the given function with $n^{2-\varepsilon}$, where ε is a very small positive number. For all of the given relations, it is possible to find some $C > 0$ such that the relation represent the time complexity of a sub-quadratic algorithm.

Question 3: (5 marks) (From the DPU textbook, Exercise 1.4) Show that

$$\log(n!) = \Theta(n \log n).$$

(Hint: To show an upper bound, compare $n!$ with n^n. To show a lower bound, compare it with $(n/2)^{n/2}$.)

Sample answer:
We can lower bound $n!$ as

$$\left(\frac{n}{2}\right) * \left(\frac{n}{2}\right) * \ldots \left(\frac{n}{2}\right) \leq 1 * 2 * 3 * \ldots \left(\frac{n}{2}\right) * \left(\frac{n}{2} + 1\right) * \cdots * n$$

and upper bound it as

$$1 * 2 * 3 * \cdots * n \leq n * n * \cdots * n$$

Hence,

$$\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n! \leq n^n$$

Since *log* is an increasing function, we can apply it on the above inequalities without changing the direction of the inequality signs. Therefore,

$$\begin{aligned}
\log\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq \log(n!) \leq \log n^n &\Rightarrow \left(\frac{n}{2}\right)\log\left(\frac{n}{2}\right) \leq \log(n!) \leq n\log n \\
&\Rightarrow \left(\frac{n}{2}\right)(\log n - \log 2) \leq \log n! \leq n\log n \\
&\Rightarrow \left(\frac{n}{2}\right)(\log n - 1) \leq \log(n!) \leq n\log n \\
&\Rightarrow \left(\frac{1}{2}\right)(n\log n - n) \leq \log(n!) \leq n\log n \qquad (*)
\end{aligned}$$

Since $\frac{n\log n}{2}$ is asymptotically greater than n (in particular, $\frac{n\log n}{2} \geq n \; \forall n \geq 2$)),

$$\begin{aligned}
-\frac{n\log n}{2} \leq -n &\Rightarrow n\log n - \frac{n\log n}{2} \leq n\log n - n \\
&\Rightarrow \frac{1}{2}\left(n\log n - \frac{n\log n}{2}\right) \leq \frac{1}{2}(n\log n - n) \\
&\Rightarrow \frac{1}{2}\left(\frac{n\log n}{2}\right) \leq \frac{1}{2}(n\log n - n) \\
&\Rightarrow \frac{1}{4}(n\log n) \leq \frac{1}{2}(n\log n - n) \qquad (**)
\end{aligned}$$

From $(*)$ and $(**)$, we have:

$$\frac{1}{4}(n\log n) \leq \log(n!) \leq n\log n. \tag{1}$$

Equation (1) shows that

$$\log(n!) = O(n\log n), \quad c = 1\ ,\ n \geq 1 \tag{2}$$
$$\log(n!) = \Omega(n\log n), \quad c = 1/4\ ,\ n \geq 2 \tag{3}$$

Having (2) and (3) means that $\log(n!) = \Theta(n\log n)$.

Question 4: (10 marks) Asymptotically analyze the running time of the following algorithm.

Sample answer:

Upper Bound: The outer loop (on i) iterates exactly n times. For each of the outer iterations, the loop on j iterates at most n times. For each of these iterations, the loop on k iterates at most n times. For each of these iterations, the loop on h iterates at most n times, since $k \geq 1$ and $i \leq n$. The rest of the algorithm contributes at most a constant factor to the running time, so the algorithm as a whole runs in $O(n^4)$ time.

Lower Bound: Consider $\frac{n}{2} \leq i \leq \frac{3n}{4}$, $j \geq \frac{3n}{4}$, and $k \leq \frac{n}{4}$. Under these restrictions, the outer loop iterates at least $\lfloor \frac{n}{4} \rfloor$ times and the next loop (on j) iterates at least $\lfloor \frac{n}{4} \rfloor$ times for each outer iteration. For each of the j iterations, the loop on k iterates $\lfloor \frac{n}{4} \rfloor$ times. For each of the k iterations, there are at least $\lfloor \frac{n}{4} \rfloor$ iterations on h (since $k \leq \frac{n}{4}$ and $i \geq \frac{n}{2}$). Therefore the algorithm as a whole runs in $\Omega(\lfloor \frac{n}{4} \rfloor^4) = \Omega(n^4)$ time. Since the algorithm runs in $O(n^4)$ time and $\Omega(n^4)$ time, it runs in $\Theta(n^4)$ time.