COMPSCI.220.S1 – Algorithms and Data Structures

Assignment 1 – Math Basics & Time Complexity of Algorithms

Due: Friday, 16th of March, 2017

Examples of solutions

Answers to Questions 1 – 2 assume an imaginary student *Peter John Ranger-Stout* with the two given names and the two-word family name. Note the valid solutions for Questions 1 – 2 are not unique and may differ from the examples provided.

1. According to the definition of a set (e.g., on Slide 8 of Lecture 1), the set $S_{\mathrm{g}} = \{\mathrm{p}, \mathrm{e}, \mathrm{t}, \mathrm{r}\}$ is the minimum-cardinality set of letters to form the first given name "peter". Accordingly, the set $S_{\mathrm{f}} = \{\mathrm{r}, \mathrm{a}, \mathrm{n}, \mathrm{g}, \mathrm{e}\}$ is the minimum-cardinality set of letters to form the first family name "ranger".

2. According to the definitions of set operations (e.g., on Slides 12 and 13 of Lecture 1), the union, $S_{\mathrm{u}} = S_{\mathrm{g}} \bigcup S_{\mathrm{f}} = \{\mathrm{p}, \mathrm{e}, \mathrm{t}, \mathrm{r}, \mathrm{a}, \mathrm{n}, \mathrm{g}\}$, and the complement, $S_{\mathrm{i}} = S_{\mathrm{g}} \setminus S_{\mathrm{f}} = \{\mathrm{p}, \mathrm{t}\}$.

3. The inner loop is executed only once (for $k = 1$), so it contributes C operations. The middle loop is executed m times where $3^{m-1} < n \leq 3^m$, or $m \approx \log_3 n$. The outer loop is executed ν times where $5(\nu - 1) < n \leq 5\nu$, or $\nu \approx n/5$. Thus $T(n) = \frac{1}{5} C n \log_3 n$ operations. More precisely, $T(n) = C \lceil \frac{n}{5} \rceil \lceil \log_3 n \rceil$.

4. The first inner loop is executed m times where $2(m-1) < n \leq 2m$, or $m \approx n/2$, thus contributing to $Cn/2$ operations.

   The loop variable j in the second inner loop is changing as $2^{2^0}, 2^{2^1}, 2^{2^2}, \ldots, 2^{2^{m-1}} < n \leq 2^{2^m}$, that is, the second inner loop is executed m times where $m - 1 < \log \log_2 n \leq m$. Hence, $m \approx \log_2 \log_2 n$, and this loop contributes $C \log_2 \log_2 n$ operations.

   The outer loop is executed μ times where $\frac{n}{3^\mu} \geq 1 > \frac{n}{3^{\mu+1}}$ so that $\mu \geq \log_3 n > \mu + 1$, that is, $\mu \approx \log_3 n$.

   Thus, we get $T(n) = C \log_3 n \, (\log_2 \log_2 n + n/2)$. More precisely,
   $T(n) = C \lfloor \log_3 n \rfloor \left(\lceil \log_2 \log_2 n \rceil + \lceil n/2 \rceil \right)$.

5. The algorithms A and B spend $T_A(n) = c_A n^3$ and $T_B(n) = c_B n^2 \log_{10} n$ time units, respectively, to process n items. The constants c_A and c_B follow from the given time for $n = 100$ items:

   $$c_A = \frac{2}{100^3} = \frac{2}{10^6} \text{ and } c_B = \frac{10}{100^2 \log_{10} 100} = \frac{1}{2000}$$

   Therefore, $T_A(10^6) = \frac{2}{10^6} 10^{18} = 2 \cdot 10^{12}$ time units and $T_B(10^6) = \frac{1}{2000} (10^6)^2 \log_{10} 10^6 = \frac{1}{2000} \cdot 6 \cdot 10^{12} = 3 \cdot 10^9$ time units.

6. $T(n)$ is $\Omega(n)$ because $T(n) = 5n \log_2 n + 500n \geq 505n$ for all $n \geq 2$.

   $T(n)$ is $O(n^{1+\varepsilon})$ because

   - by the Limit Rule $\lim_{n \to \infty} \frac{T(n)}{n^{1+\varepsilon}} = 0$, i.e.

     $$\lim_{n \to \infty} \frac{5n \log_2 n + 500n}{n^{1+\varepsilon}} = \lim_{n \to \infty} \left(5 \frac{\log_2 n}{n^\varepsilon} \right) + \lim_{n \to \infty} \left(500 \frac{1}{n^\varepsilon} \right) = 0$$

   - For $n \to \infty$, the last term $\frac{1}{n^\varepsilon}$ tends to zero
   - By the L'Hopital's rule of calculus for $n \to \infty$, the first term $5 \frac{\log_2 n}{n^\varepsilon}$ also tends to zero:

     $$\lim_{x \to \infty} \frac{\log_2 x}{x^\varepsilon} = \lim_{x \to \infty} \frac{x^{-1} \log_2 \mathrm{e}}{\varepsilon x^{\varepsilon - 1}} = \lim_{x \to \infty} \frac{\log_2 \mathrm{e}}{x^\varepsilon} = 0$$

     where $\mathrm{e} = 2.71828\ldots$ is the base of the natural logarithms.

7. No, to conclude that $T(n)$ is $\Theta(n^2)$ the processing time has to be simultaneously $\Omega(n^2)$ and $O(n^2)$. The derived bounds are insufficient for such a conclusion: actually $T(n)$ can be anywhere within the range between $\Theta(n)$ to $\Theta(n^3)$ but not necessarily $\Theta(n^2)$.

8. (a) Because $n > log_2 n$ and $6n\left(\log_2 n\right)^2 > 0$ for all $n \geq 2$, $0.01n^2 \log_2 n < T(n) < 6.01n^2 \log_2 n$ for $n \geq 2$, so that $T(n)$ is $\Theta(n^2 \log n)$.

   (b) The statement $T(n)$ is $O(n^3)$ holds because $n > \log_2 n$ and thus $T(n) < 6.01n^3$ for all $n \geq 2$.

   (c) The statement $T(n)$ is $\Omega(n^2)$ holds because $\log_2 n \geq 1$ and thus $T(n) > 0.01n^2$ for all $n \geq 2$.

9.
   - For $n = 5^m$ the recurrence is written as $T\left(5^m\right) = 5T\left(5^{m-1}\right) + 5$.
   - **Telescoping:**

$$
\begin{array}{lcl}
T\left(5^m\right) & = & 5T\left(5^{m-1}\right) + 5 \\
5T\left(5^{m-1}\right) & = & 5^2 T\left(5^{m-2}\right) + 5^2 \\
5^2 T\left(5^{m-2}\right) & = & 5^3 T\left(5^{m-3}\right) + 5^3 \\
\ldots & \ldots & \ldots \\
5^{m-2} T\left(5^2\right) & = & 5^{m-1} T(5) + 5^{m-1} \\
5^{m-1} T(5) & = & 5^m \underbrace{T\left(5^0\right)}_{T(1)=0} + 5^m
\end{array}
$$

   - **Restoring the closed-form formula** by substitution:

$$
T\left(5^m\right) = 5 + 5^2 + \ldots + 5^m = 5\frac{5^m - 1}{5 - 1} = \frac{5}{4}(n-1)
$$

   i.e. $T(n) = 1.25(n-1)$.

Marking scheme

Clear structure of your report and detailed explanations	up to 20%
Correctness of the final answers	up to 30%
Correctness of the intermediate steps in deriving the answers	up to 30%
Detailed explanations of all steps with references, if necessary, to the textbook	up to 20%

Total: up to 100% of the full marks