CS340 – Advanced Data Structures and Algorithm Design – Fall 2020
Assignment 1 – September 11, 2020

Dr. Sandra Zilles, Department of Computer Science, University of Regina

answer key

Problem 1 (1+1+1+1 marks). For each question,

- subtract half a mark if the solution is not the simplest possible, e.g., if someone writes $\Theta(0.1N^3\sqrt{N})$ or $\Theta(N+1)$,
- give no mark if the growth rate class given is not correct.

(a) $T(N) = 5.3 \cdot N^3\sqrt{N} = +22N^3\log(N) + 100N\sqrt{N} = \Theta(N^3\sqrt{N})$

(b) $T(N) = 0.6\frac{\log^2(N)}{N}12.5N = \Theta(N)$

(c) $T(N) = \Theta(N \cdot \log^3(N))$

(d) $T(N) = \frac{36\log^2(N)}{N^2} + \frac{2}{3}\frac{\log(N)\cdot 3^{N+1}}{N^2} + 57 \cdot \frac{\log(N)}{N^2} = \Theta(\frac{\log(N)\cdot 3^N}{N^2})$

Problem 2 (3+3 marks). (a) $T(N) = o(f(N))$ implies $T(N) = O(f(N))$ and $T(N) \neq \Omega(f(N))$. (0.5 marks)
$f(N) = \Theta(g(N))$ implies $f(N) = O(g(N))$. (0.5 marks)
$T(N) = O(f(N))$ and $f(N) = \Theta(g(N))$ together imply $T(N) = O(g(N))$. (1 mark)
$T(N) \neq \Omega(f(N))$ and $f(N) = O(g(N))$ implies $T(N) \neq \Omega(g(N))$ and thus $T(N) = o(g(N))$. (1 mark)

(b) $T(N) = 1$, $f(N) = \log(N)$, $g(N) = N$. (1.5 marks for correct choice, 0.5 marks for every correct relation, e.g., 1 mark if $f(N) = o(g(N))$ and $T(N) = O(g(N))$, but $T(N) = \Omega(f(N))$)
Then $T(N) = O(g(N))$, because N grows faster than 1 (a constant). (0.5 marks)
$f(N) = o(g(N))$, because $\log(N) = O(N)$ and $\log(N)$ does *not* grow as fast as N. (0.5 marks)
$T(N) \neq \Omega(f(N))$, because $\log(N)$ grows faster than 1 (a constant). (0.5 marks)

Problem 3 (3+3+3 marks). For each of the following code fragments, determine the best possible asymptotic upper bound on its running time, depending on `n`. Give a brief explanation for each of your answers.

(a) The statement inside the loops has a running time of $O(1)$. (1 mark)
The inner loop iterates $O(n)$ times (namely roughly $n/2$ times), thus making the overall running time of the inner loop $O(n)$. (1 mark)
The outer loop runs n times, resulting in an overall running time of $O(n^2)$. (1 mark)

(b) The statement inside the loops has a running time of $O(1)$. (0.5 marks)
The inner loop iterates $O(n)$ times (namely roughly $n/2$ times), thus making the overall running time of the inner loop $O(n)$. (1 mark)
The outer loop runs $O(\log(n))$ times, resulting in an overall running time of $O(n \cdot \log(n))$. (1.5 marks)

(c) The running time $T(n)$ can be described (asymptotically) by $T(n) = 1$ if $n = 1$ and $T(n) = T(n-1) + 1$ if $n > 1$. (1.5 marks)
This results in $T(n) = n = O(n)$. (1.5 marks)
(Other correct explanations will also be accepted.)

Problem 4 (4+2+4 marks).

(a) half marks for a reasonable approach that does not work because of a small error

(b) half marks for a reasonable approach that does not work because of a small error

(c) $n \log(n)$ is the only function for which the ratio by `result` calculated in (b) approaches a constant for growing n. Hence we may conclude that the asymptotic growth of the result of `assignment1Algorithm(n)` is $\Theta(n \cdot \log(n))$. (1.5 marks)
$n \log(n)$ is the only function for which the ratio by `timeUsed` calculated in (b) approaches a constant for growing n. Hence we may conclude that the asymptotic growth of the running time of `assignment1Algorithm(n)` is $\Theta(n \cdot \log(n))$. (1.5 marks)
$n \log(n)$ grows only slightly faster than n, whereas $n\sqrt{n}$ grows "significantly" faster than $n \log(n)$. (1 mark)