CS3343 Analysis of Algorithms Fall 2017

Homework 1

Due 9/8/17 before 11:59pm

Justify all of your answers with comments/text in order to receive full credit. Completing the assignment in Latex will earn you extra credit on Midterm 1.

1. Sorting (10 points)

Consider the sorting algorithm below which sorts the array $A[1 \ldots n]$ into increasing order by repeatedly bubbling the minimum element of the remaining array to the left.

Algorithm 1 mysterysort(int $A[1 \ldots n]$)

$i = 1$ // C1 — 1
while $i <= n$ **do** // C2 — n + 1
  //(I) The elements in $A[1 \ldots (i-1)]$ of the array are in sorted order and are all smaller than the elements in $A[i \ldots n]$
  $k = n$; // C3 — n
  // t_k = # of times while loop executes
  while $(k >= i+1)$ **do** // C4 — $\sum\limits_{1}^{n}(t_k + 1)$
    //Inner while loop moves the smallest element in $A[i \ldots n]$ to $A[i]$
    if $A[k-1] > A[k]$ **then** // C5 — $\sum\limits_{1}^{n} t_k$
      swap $A[k]$ with $A[k-1]$ // C6 — Worst Case: $\sum\limits_{1}^{n} t_k$ Best Case: 0
    end if
    $k--$; // C7 — $\sum\limits_{1}^{n} t_k$
  end while
  $i++$; // C8 — n
end while

(1) (2 points) Consider running the above sort on the array $[5, 3, 1, 4, 2]$. Show the sequence of changes which the algorithm makes to the array.

5	3	1	4	2	⟶	1	5	3	2	4	⟶	1	2	5	3	4	⟶
1	2	3	5	4	⟶	1	2	3	4	5							

(2) (4 points) Use induction to prove the loop invariant (I) is true and then use this to prove the correctness of the algorithm. Specifically complete the following:

(a) Base case

→ $i = 1 \Rightarrow$ No elements in $A[1 \ldots i-1]$, so loop invariant trivially holds true

(b) Inductive step
(**Hint:** you can assume that the inner loop moves the smallest element in $A[i \dots n]$ to $A[i]$)
$\rightarrow$ i_{old} = value before iteration
$\rightarrow$ $A[1 \dots i_{old} - 1]$ are sorted and smaller than $A[i_{old} \dots n]$
$\rightarrow$ Inner loop moves the smallest element in $A[i_{old} \dots n]$ to $A[i_{old}] \Rightarrow$ $A[1 \dots i_{old}]$ are now in sorted order and smaller than $A[i_{old} + 1 \dots n]$
$\rightarrow$ Since $i_{new} = i_{old} + 1$, $A[1 \dots i_{new} - 1]$ are sorted and smaller than $A[i_{new} \dots n]$, thus maintaining the loop invariant after iteration

(c) Termination step
$\rightarrow$ Terminates when $i > n$, i.e. $i = n + 1$
$\rightarrow$ Loop invariant says $A[1 \dots i - 1]$ are sorted and smaller than $A[i \dots n]$
$\rightarrow$ $i = n + 1 \Rightarrow A[1 \dots n]$ are sorted and smaller than $A[n + 1 \dots n]$, i.e. the entire array is now sorted and the algorithm terminates correctly producing a sorted array

(3) (2 points) Give the best-case and worst-case runtimes of this sort in asymptotic (i.e., O) notation.
Worst Case:
$\rightarrow$ Array is in reverse order $\Rightarrow C1 + (C3 + C8)n + C2(n + 1) + (C5 + C6 + C7)\sum_1^n t_k + C4\sum_1^n (t_k + 1)$
$\rightarrow$ $C4\sum_1^n (t_k + 1) = C4n + C4\sum_1^n t_k$
$\rightarrow$ $t_k = n - k \Rightarrow \sum_1^n (n - k) = n - 1 + n - 2 + ... + 2 + 1 \approx \frac{n(n-1)}{2} \in O(n^2)$
Best Case:
$\rightarrow$ Array is in sorted order $\Rightarrow C1 + (C3 + C8)n + C2(n + 1) + (C5 + C7)\sum_1^n t_k + C4\sum_1^n (t_k + 1) + C6\sum_1^n 0$
$\rightarrow$ Only C6 is not executed, so the runtime is the same as worst case $\Rightarrow$ $f(n) \in O(n^2)$

2. Asymptotic Notation (8 points)

Show the following using the definitions of O, Ω, and Θ.

(1) (2 points) $2n^3 + n^2 + 4 \in \Theta(n^3)$
$f(n) \in O(n^3)$:
$\rightarrow$ $2n^3 + n^2 + 4 \leq 2n^3 + n^3 + 4n^3 = 7n^3$
$\rightarrow$ $7n^3 \leq Cn^3$ for $C = 7$ and $n \geq n_0 = 1 > 0 \Rightarrow f(n) \in O(n^3)$
$f(n) \in \Omega(n^3)$:
$\rightarrow$ $2n^3 + n^2 + 4 \geq n^3 + 0 + 0 = n^3$
$\rightarrow$ $n^3 \geq Cn^3$ for $C = 1$ and $n \geq n_0 = 1 > 0 \Rightarrow f(n) \in \Omega(n^3)$
Since $f(n) \in O(n^3)$ and $f(n) \in \Omega(n^3) \Rightarrow f(n) \in \Theta(n^3)$

(2) (2 points) $3n^4 - 9n^2 + 4n \in \Theta(n^4)$
(**Hint:** careful with the negative number)
$f(n) \in O(n^4)$:
$\rightarrow\ 3n^4 - 9n^2 + 4n \leq 3n^4 - 0 + 4n(n^3) = 3n^4 + 4n^4 = 7n^4$
$\rightarrow\ \leq 7n^4$ for $C = 7$ and $n_0 = 1 \Rightarrow f(n) \in O(n^4)$
$f(n) \in \Omega(n^4)$:
$\rightarrow\ 3n^4 - 3 \cdot 3n^2 + 4n \geq 3n^4 - n \cdot n \cdot n^2 + 0 = 3n^4 - n^4 = 2n^4$
$\rightarrow\ \geq 2n^4$ for $C = 2$ and $n_0 = 3 \Rightarrow f(n) \in \Omega(n^4)$
Since $f(n) \in O(n^4)$ and $f(n) \in \Omega(n^4) \Rightarrow f(n) \in \Theta(n^4)$

(3) (4 points) Suppose $f(n) \in O(g_1(n))$ and $f(n) \in O(g_2(n))$. Which of the following are true? Justify your answers using the definition of O. Give a counter example if it is false.

(a) $f(n) \in O(\ 5 * g_1(n) + 100\ )$
$\rightarrow\ f(n) \in O(g_1(n)) \Rightarrow f(n) \leq C \cdot g_1(n)$ for $C = k_1$ and $n_0 = k_2$
$\rightarrow\ \leq 5 \cdot k_1 \cdot g_1(n) \leq 5 \cdot k_1 \cdot g_1(n) + 100$
$\rightarrow$ By definition $f(n) \in O(5g_1(n) + 100)$ for $C = k_1$ and $n_0 = k_2$

(b) $f(n) \in O(\ g_1(n) + g_2(n)\ )$
$\rightarrow\ f(n) \in O(g_1(n)) \Rightarrow f(n) \leq C_1 \cdot g_1(n)$ for $C_1 = k_1$ and $n_0 = k_2$
$\rightarrow\ f(n) \in O(g_2(n)) \Rightarrow f(n) \leq C_2 \cdot g_2(n)$ for $C_2 = m_1$ and $n_0 = m_2$
$\rightarrow\ f(n) \leq k_1 \cdot g_1(n) + m_1 \cdot g_2(n) \leq k_1 \cdot m_1(g_1(n) + g_2(n))$
$\rightarrow$ By definition $f(n) \in O(g_1(n) + g_2(n))$ for $C = k_1 \cdot m_1$ and $n_0 = max(k_2, m_2)$

(c) $f(n) \in O(\ \frac{g_1(n)}{g_2(n)}\ )$
$\rightarrow$ False, Counterexample: $f(n) = n$, $g_1(n) = n$, $g_2(n) = n^2$
$\rightarrow\ f(n) \in O(n)$, $f(n) \in O(n^2)$, $f(n) \notin O(\frac{n}{n^2}) \Rightarrow f(n) \notin O(n^{-1})$

(d) $f(n) \in O(\ \max(g_1(n), g_2(n))\ )$
$\rightarrow$ Trivially true, since $f(n) \leq C_1 \cdot g_1(n)$ and $f(n) \leq C_2 \cdot g_2(n)$
$\rightarrow$ Clearly, $f(n)$ will be less than or equal to the larger of the two functions.
$\rightarrow$ Thus, $f(n) \in O(max(g_1(n), g_2(n)))$

3. Summations (4 points) Find the order of growth of the following sums.

(1) $\sum_{i=5}^{n}(4i + 1)$.
$\Rightarrow\ 4 \cdot \sum_{i=5}^{n} i + \sum_{i=5}^{n} 1$
$\Rightarrow\ 4(\sum_{i=1}^{n} i - \sum_{i=1}^{4} i) + (n - 4) = 4(\frac{n(n+1)}{2} - 10) + (n - 4)$
$\Rightarrow\ 2n^2 + 3n - 44 \Rightarrow$ sum is $O(n^2)$

(2) $\sum_{i=0}^{\log_2(n)} 2^i$ (for simplicity you can assume n is a power of 2)
$\Rightarrow$ From formula sum $= \frac{1-2^{\log_2(n)+1}}{1-2}$
$\Rightarrow\ \frac{1-2^{\log_2(n)}.2}{-1} = \frac{1-2n}{-1} = 2n - 1 \Rightarrow$ sum is $O(n)$