In-class Test
COMS10007 Algorithms 2018/2019

12.03.2019

Reminder: $\log n$ denotes the binary logarithm, i.e., $\log n = \log_2 n$. We also write $\log^c n$ as an abbreviation for $(\log n)^c$.

Make sure to put your name on every piece of paper that you hand in!

1 O-notation

1. Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be a function. Define the set $\Omega(f(n))$.
2. Give a formal proof of the statement:

$$10 \log n \in O(\log^2 n) \ .$$

3. For each of the following statements, indicate whether it is true of false: (no justification needed)
   (a) $n \in O(n^2)$
   (b) $\log n \in O(n^3)$
   (c) $\log n \in O(\sqrt{\log n})$
   (d) $n! \in O(2^n)$
   (e) $2^{\sqrt{\log n}} = O(\log^2 n)$
   (f) $f(n) \in O(g(n))$ implies $g(n) \in \Omega(f(n))$
   (g) $f(n) \notin O(g(n))$ implies $g(n) \in O(f(n))$

2 Sorting Algorithms

Let A be an array of length n with $A[i] = A[j]$, for every $0 \leq i, j \leq n - 1$.

1. What is the runtime of Heapsort on A? (in Θ-notation, no justification needed)
2. What is the runtime of Mergesort on A? (in Θ-notation, no justification needed)
3. What is the runtime of Insertionsort on A? (in Θ-notation, no justification needed)
4. What are the best-case and worst-case runtimes of Mergesort? (no justification needed)
5. Illustrate how the Mergesort algorithm sorts the following array (for example using a recursion tree):

9 3 2 7 1 6 11 4

Continued on next page...

3 Loop-Invariant

Consider the following algorithm: (it takes two parameters, an array A of length n of positive integers, and an integer x)

Algorithm 1

Require: A is an array of n positive integers, x is an integer

1: $c \leftarrow 0$
2: **for** $i \leftarrow 0, 1, \ldots, n-1$ **do**
3:     **if** $A[i] < x$ **then**
4:         $c \leftarrow c + 1$
5:     **end if**
6: **end for**
7: **return** c

1. Consider the for-loop of the algorithm. One of the following options is a correct loop-invariant:

   At the beginning of iteration i (i.e., after i is updated in Line 2 and before the code in Lines 3, 4, and 5 is executed) ...

   (a) ... $c = |\{j \ : \ 0 \leq j < i \text{ and } A[j] < x\}|$
   (b) ... $c = |\{j \ : \ 0 \leq j \leq i \text{ and } A[j] < x\}|$
   (c) ... $c = |\{j \ : \ 0 \leq j < i \text{ and } A[j] \leq x\}|$
   (d) ... $c = |\{j \ : \ 0 \leq j \leq i \text{ and } A[j] \leq x\}|$

   State which one is correct.

2. *Initialization:* Consider the correct invariant. Argue that at the beginning of the first iteration, i.e. when $i = 0$, the loop-invariant holds.

3. *Maintenance:* Consider the correct invariant. Suppose that the loop invariant holds at the beginning of iteration i. Argue that the loop-invariant then also holds at the beginning of iteration $i + 1$.

4. *Termination:* What does the algorithm compute? Argue that this follows from the correct loop invariant.