CS 32

Long Exam 1 - Answer Key

September 11, 2014

General Instructions

- Answer the items completely. Show your solutions/justifications when asked.
- Write as legibly as possible. Illegible or unreadable answers and solutions may not merit any points.
- Refrain from making unnecessary motions and sounds during the exam. Any suspicious behavior will be dealt with accordingly.
- Direct all questions to the proctor.
- If you need to go to the CR, hand your questionnaire, answer sheet, and scratch paper to the proctor before heading out. Only one person at any given time is allowed to go out.
- Once you're done with the exam (one way or the other), place your scratch papers and the questionnaire inside your blue book.

Questions

1. Given two functions, $f(n)$ and $g(n)$, prove that $g(n) = \Theta(f(n))$ *if and only if* $g(n) = O(f(n))$ and $g(n) = \Omega(f(n))$.
ANSWER: For the *if* part, $g(n) = \Theta(f(n))$ implies that there exist positive constants, K_1, K_2, and n_0 such that $0 \leq K_1 \cdot f(n) \leq g(n) \leq K_2 \cdot f(n), \forall n \geq n_0$. The "left" part of the inequality ($0 \leq K_1 \cdot f(n) \leq g(n)$) provides the condition to satisfy $g(n) = \Omega(f(n))$, while the "right" part ($0 \leq g(n) \leq K_2 \cdot f(n)$) provides the condition to satisfy $g(n) = O(f(n))$.

   For the *only if* part, $g(n) = O(f(n))$ implies the existence of positive constants K and n_0 such that $0 \leq g(n) \leq K \cdot f(n), \forall n \geq n_0$, and in a similar fashion, $g(n) = \Omega(f(n))$ implies the existence of positive constants K' and n'_0 such that $0 \leq K' \cdot f(n) \leq g(n), \forall n \geq n'_0$. Note at this point that $K \neq K'$ and $n_0 \neq n'_0$. Given that $f(n)$ asymptotically bounds $g(n)$ both from above and below, and the fact that the constants in the definition of the bounds are not necessarily unique, we could find a common constant n''_0 (simplest would be $n''_0 = \max(n_0, n'_0)$, or even greater), and two other positive constants C and C' (both not necessarily equal to K and K' respectively), such that both $0 \leq g(n) \leq C \cdot f(n)$ and $0 \leq C' \cdot f(n) \leq g(n)$ hold true for all $n \geq n''_0$. Therefore, this implies that if $g(n) = O(f(n))$ and $g(n) = \Omega(f(n))$, then $g(n) = \Theta(f(n))$.

2. Given the following EASY code below:
```
for i ←1 to n do
  for j ← 1 to (n - i) do
    for k ← (j + 1) to (j + i) do
       output i, j, k
    endfor
  endfor
endfor
```
Assume that any statement has constant cost equal to 1, and that the time complexity of the EASY code is represented as $T(n)$. Find the "smallest" function $f(n)$, such that $T(n) = O(f(n))$ and that the bound is tight. Show your solutions. *(HINT: It is sufficient to find the exact running time of the most significant part/s of the code to get the asymptotic running time of the whole code fragment.)*
ANSWER: $f(n) = n^3$. The most significant parts of the code are the the third for loop which is executed $\frac{n^3+3n^2-4n}{6}$ times, and the statement `output i, j, k` which is executed ($\frac{n^3-n}{6}$ times). Both have the same asymptotic upper bound of n^3.

3. Arrange the following functions in increasing order of complexity (assume we have sufficiently large value for n): $n^{5/2}, \log(n^2), n^2, 10^{\log(n^{3/2})}, \log^2(n)$
ANSWER: $\log(n^2), \log^2(n), 10^{\log(n^{3/2})}, n^2, n^{5/2}$

4. Four stacks coexist in an array of size 400. The state variables of the stacks are as follows:

- B(1:5) = (0, 100, 210, 290, 400)
- T(1:4) = (80, 200, 290, 379)
- OLDT(1:4) = (50, 200, 260, 350)

An item is pushed onto the 3rd stack, causing an overflow. Assuming the Garwick-Knuth algorithm was used to reallocate memory.

(a) What are the values of the allocation factors a and b?
ANSWER: $a = 1.25, b = 0.5$

(b) How much theoretical free space would have been given to stacks 1 to 3?
ANSWER: 34.25

(c) What is the new bottom of stack 4?
ANSWER: $NEWB(4) = 295$

5. Given the following set of predecessor-successor pair inputs: (G,F), (F,H), (H,B), (C,I), (A,I), (D,A), (E,D), (B,A), (E,B), (J,G). Perform topological sort using the algorithm shown in class and show the final state of the table *and* the topologically sorted list of distinct letters in the inputs.
ANSWER: *TABLE:*

	COUNT	LIST
A	2	→ [I] → Λ
B	2	→ [A] → Λ
C	0	→ [I] → Λ
D	1	→ [A] → Λ
E	0	→ [D] → [B] → Λ
F	1	→ [H] → Λ
G	1	→ [F] → Λ
H	1	→ [B] → Λ
I	2	→ Λ
J	0	→ [G] → Λ

OUTPUT: C E J D G F H B A I

Scoring Mechanics

1. For Item 1:
   - **1 point**: Proper proof given both directions, i.e. "if" and "only if" parts.
   - **0.75 point**: Proper proof given to one way, significant attempt to prove other direction was made but deficient.
   - **0.5 point**: Proper proof given one way, no significant attempt to prove other direction.
   - **0.25 point**: Significant attempt to prove both directions was made, but otherwise deficient.
2. For Item 2, the following points are awarded **provided solutions are shown**:
   - **1 point**: Exact running time for significant part/s of the EASY code given, as well as the proper asymptotic running time.
   - **0.9 point**: Exact running time for significant part/s of the EASY code given, but with improper or no asymptotic running time given.
   - **0.75 point**: Proper asymptotic time given, significant solution provided on the exact running time for the significant part/s of the EASY code, but otherwise erroneous deficient.
   - **0.5 point**: Proper asymptotic time given, solutions provided for the exact running time of the significant part/s of the EASY code are insignificant and erroneous.
   - **0.25 point**: Significant attempt to provide answers, but otherwise deficient.
3. **0.2 point** for each function in the correct position within the list.
4. Awarding of points follows the **all-or-nothing rule** for Item 4. **0.5 point** for sub-item (a) and **0.25 point** for sub-items (a) and (b).
5. **0.1 point** ***deduction*** for each erroneous feature in either table and list output.