CIS 575: Introduction to Algorithm Analysis
Exam I with suggested answers

February 22, 2017, 2:30-3:20pm

General Notes

- You can have one sheet (each side may be used) of notes, but no other material and no use of laptops or other computing devices.
- If you believe there is an error or ambiguity in any question, mention that in your answer, and *state your assumptions*.
- Please write your name on this page.

Good Luck!

NAME:

1. *Asymptotic Behavior, 20p.*

In this question, you are asked to compare pairs of functions, and for each pair (f, g), and each $X \in \{O, \Omega, \Theta, o, \omega\}$, say whether $f \in X(g)$ or not. The first entry has been given, reflecting that $n^2 \in O(n^3)$ and even $n^2 \in o(n^3)$.

f	g	O	Ω	Θ	o	ω
n^2	n^3	**YES**	**NO**	**NO**	**YES**	**NO**
$\sqrt{n}$	$\lg(n)$	**NO**	**YES**	**NO**	**NO**	**YES**
$n\lg(n)$	$n\ln(n)$	**YES**	**YES**	**YES**	**NO**	**NO**
n^3	2^n	**YES**	**NO**	**NO**	**YES**	**NO**
$\lg(n^2)$	$\lg(n)$	**YES**	**YES**	**YES**	**NO**	**NO**
$\lg(n!)$	$n\lg(n)$	**YES**	**YES**	**YES**	**NO**	**NO**

2. *Algorithm Correctness, 25p.* Consider the (yet incomplete) algorithm

Precondition: x, y are integers with $x \geq 0$ and $y > 0$
Postcondition: returns q, r such that $x = q \cdot y + r$ and $y > r \geq 0$

$\textsc{Modulo}(x, y)$
  $q, r \leftarrow 0, \underline{\mathbf{x}}$
  // **Invariant:** $x = q \cdot y + r$ and $r \geq 0$
  while $\underline{\mathbf{y} \leq \mathbf{r}}$
    $q, r \leftarrow q + 1, \underline{\mathbf{r} - \mathbf{y}}$
  return q, r

Fill in the 3 missing expressions in the above algorithm, so that

1. the invariant is established by the initial assignments;
2. the invariant is maintained by the loop body;
3. the loop test eventually becomes false and then the invariant implies the postcondition.

You are *not* asked to prove that your answers make the 3 conditions hold.

3. *Analyzing Recursive Algorithms, 35p.* Consider the two programs

```
FUNSORT(A[1..n])
   if n > 1
      m ← ⌊n/2⌋
      for i ← 1 to m
         for j ← m + 1 to n
            if A[j] < A[i]
               A[i] ↔ A[j]
      FUNSORT(A[1..m])
      FUNSORT(A[m + 1..n])
```

```
WRONGSORT(A[1..n])
   if n > 1
      m ← ⌊n/2⌋
      for i ← 1 to m
         if A[i + m] < A[i]
            A[i] ↔ A[i + m]
      WRONGSORT(A[1..m])
      WRONGSORT(A[m + 1..n])
```

(a, 20p). For each program, write a recurrence for its running time. (You do not need to argue for it.)

Answer:
Since the nested loop obviously runs in time $\Theta(n^2)$, for the running time $T(n)$ we get the recurrence

$$T(n) \in 2T(\frac{n}{2}) + \Theta(n^2).$$

Since the loop obviously runs in time $\Theta(n)$, for the running time $T(n)$ we get the recurrence

$$T(n) \in 2T(\frac{n}{2}) + \Theta(n).$$

(b, 15p). Use the Master Theorem (listed on the bottom) to solve the above recurrences, so as to express the running time in the form $\Theta(f(n))$ with f as simple as possible.

The Master Theorem, with $a = b = 2$ and $r = \log_b(a) = 1 < 2 = q$, yields $T(n) \in \mathbf{\Theta(n^2)}$.

The Master Theorem, with $a = b = 2$ and $r = \log_b(a) = 1 = q$, yields $T(n) \in \mathbf{\Theta(n \lg(n))}$.

Master Theorem (one version). Given the recurrence

$$T(n) = aT(n/b) + f(n) \text{ for } n > n_0$$

with a an integer and with $b > 1$ a real number, $\lfloor\ \rfloor$ or $\lceil\ \rceil$ around n/b, and T eventually non-decreasing. With $r = \lg_b a$, solutions are as follows:

1. if $f(n) \in X(n^r)$ then $T(n) \in X(n^r \lg n)$ $(X \in \{O, \Omega, \Theta\})$.
2. if $f(n) \in X(n^q)$ with $q > r$ then $T \in X(n^q)$ $(X \in \{O, \Omega, \Theta\})$.
3. if $f(n) \in O(n^q)$ with $q < r$ then $T \in \Theta(n^r)$.

4. *Properties of asymptotic notation, 20p.* Below is a proof of the result:

if $f(n) \in O(n^3)$ and $g(n) \in \Omega(n)$ then $h(n) \in O(n^2)$
where $h(n) = \dfrac{f(n)}{g(n)}$ (whenever $g(n) > 0$).

Your task is to fill in the missing parts.

Begin Proof. By assumption, there exists $n_1 \geq 0$ and $c_1 > 0$ such that for $n \geq n_1$ we have

$$f(n) \leq \underline{\mathbf{c_1 n^3}}$$

and there exists $n_2 \geq 0$ and $c_2 > 0$ such that for $n \geq n_2$ we have

$$g(n) \underline{\geq \mathbf{c_2 n}}.$$

We infer that for $n \geq \max(\underline{\mathbf{n_1, n_2, 1}})$ and $c = \underline{\dfrac{\mathbf{c_1}}{\mathbf{c_2}}}$ we have the calculation

$$h(n) = \frac{f(n)}{g(n)} \underline{\leq} \frac{c_1 n^3}{c_2 n} = cn^2$$

which does show $h(n) \in O(n^2)$. *End Proof.*