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

February 27, 2013, 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 notation, 20p.* Let f and g be functions that map non-negative integers into positive real numbers; hence f/g is well-defined and given by:

$$(f/g)(n) = \frac{f(n)}{g(n)} \text{ for } n \geq 0.$$

Prove that if $f \in O(n^3)$ and $g \in \Omega(n)$ then $f/g \in O(n^2)$.

Answer: Our assumption is that there exists $n_1, n_2 \geq 0$ and $c_1, c_2 > 0$ such that $f(n) \leq c_1 n^3$ for all $n \geq n_1$, and $g(n) \geq c_2 n$ for all $n \geq n_2$. For $n \geq \max(n_1, n_2, 1)$ we then have

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

which shows that $f/g \in O(n^2)$.

2. *Algorithm correctness, 30p.* Prove that the below algorithm meets its specification. To do so, you must argue:

- that the loop invariant is established before the loop and is maintained after each iteration, and
- that the loop test eventually becomes false and then the invariant implies the postcondition.

Precondition: x, y are integers with $x > 0$
Postcondition: returns $z = x \cdot y$

Mult(x, y)
  $q, z \leftarrow x, 0$
  // **Invariant:** $z = (x - q) \cdot y$
  while $q \neq 0$
    $q, z \leftarrow q - 1, z + y$
  return z

Answer: After the code preceding the loop, the invariant amounts to $0 = (x - x) \cdot y$ which trivially holds.

To see that the invariant is maintained by a loop iteration, observe that its left hand side as well at its right hand side will be increased by y.

By the precondition, q is initially a positive integer; as q is decremented by 1 in each iteration, q will eventually become 0. Then the loop will terminate, and the invariant will tell $z = (x - 0) \cdot y$ which gives the postcondition.

3. *Running Time Analysis, 25p.* Analyze the worst-case running time of the algorithm below, and express your answer as simply as possible using Θ-notation in terms of n. You should first state a recurrence, and then solve that recurrence using (one of the variants of) the Master Theorem.

```
OCCURS(A[1..n], x)
   if n < 1
      return false
   else
      p ← (n + 1) div 2
      if A[p] = x
         return true
      else if A[p] < x
         return OCCURS(A[p + 1..n], x)
      else
         return OCCURS(A[1..p − 1], x)
```

Answer: We get the recurrence (for the worst case when x is not in A)

$$T(n) = T(\frac{n}{2}) + f(n)$$

with $f(n) \in \Theta(1)$. We can apply the Master Theorem with $a = 1, b = 2$ and thus $r = \log_2(1) = 0$; as $f(n) \in \Theta(n^0)$ we get $T(n) \in \Theta(n^0 \log(n)) = \Theta(\log(n))$.

4. *Amortized Analysis, 25p.* As in a previous homework assignment, we consider stacks implemented by tables, and shall use n to denote the current number of stack elements and k to denote the current table size (thus $n \leq k$ will always hold). In this question, we assume that the *only* operation is Pop; this operation will decrement n by 1 and will usually have actual cost 1. However, if the new value of n equals $\lfloor k/2 \rfloor$ then our implementation will (in order to free space) copy the stack elements into a new table of size $\lfloor k/2 \rfloor$; the actual cost of this operation is $\lfloor k/2 \rfloor$.

Your task is to prove that in either case, the *amortized cost* of a Pop operation is bounded by a constant. For that purpose, use the potential function $\mathbf{\Phi(n, k) = k - n}$ (which is always ≥ 0 and can be assumed to be initially 0).

Answer: For an ordinary pop, the actual cost is 1 and Φ increases by 1 (as n decreases by 1 while k stays put); hence the amortized cost is $1 + 1 = 2$.

Now assume $n' = \lfloor k/2 \rfloor$ where $n' = n - 1$ is the new value of n. Then Φ becomes 0, whereas before it was $k-n = k-n'-1 = k-\lfloor k/2 \rfloor-1 \geq \lfloor k/2 \rfloor-1$. The amortized cost $\lfloor k/2 \rfloor + \Delta\Phi$ is thus bounded by $\lfloor k/2 \rfloor - (\lfloor k/2 \rfloor - 1) = 1$.