Computer Science B36 | September 27, 2006
Scarborough Campus | University of Toronto

Homework Assignment #2
Due: October 11, 2006, by 12 noon
(in the course drop box)

*Appended to this document is a cover page for your assignment. Fill it out, staple your answers to it, and deposit the resulting document into the course drop box. Please do **not** enclose your assignment in an envelope.*

Question 1. (5 marks) Prove that if $m \in \mathbb{N}$ then the following program terminates.

```
FooBar(m)
    d := 2 * m
    if m mod 2 = 0 then i := m
    else i := m + 1 end if
    while i ≠ d do
        i := i + 2
    end while
```

Question 2. (10 marks) Prove that the program SecondMin(A) below is correct with respect to the following precondition/postcondition pair.

Precondition: A is an integer array, $length(A) \geq 2$, and all integers in A are distinct.

Postcondition: SecondMin(A) returns the second smallest integer in A.

```
SecondMin(A)
    m := min(A[1], A[2])
    s := max(A[1], A[2])
    i := 3
    while i ≤ length(A) do
        if A[i] < m then s := m; m := A[i]
        elsif A[i] < s then s := A[i] end if
        i := i + 1
    end while
    return s
```

Question 3. (10 marks) Assume that Occur(A, f, ℓ, x) is a program that returns the number of occurrences of x in $A[f..\ell]$. More precisely, it is correct with respect to the following precondition/postcondition pair:

Precondition: A is an array, $length(A) \geq 1$, and f, ℓ are integers such that $1 \leq f \leq \ell \leq length(A)$.

Postcondition: Occur(A, f, ℓ, x) returns $|\{i : f \leq i \leq \ell \text{ and } A[i] = x\}|$.

Recall that m is a ***majority element*** of an array A if m appears in more than half of the positions of A. Prove that the program FindMaj(A) below is correct with respect to the following precondition/postcondition pair:

Precondition: A is a nonempty array.

Postcondition: FindMaj(A) returns the majority element of A, if one exists; otherwise, it returns $\perp$.

continued

```
FindMaj(A)
    found := false
    i := 0
    m := (1 + length(A)) div 2
    while not found and i < m do
        i := i + 1
        if Occur(A, i, length(A), A[i]) > length(A)/2 then found := true end if
    end while
    if found then return A[i]
    else return ⊥ end if
```

(FindMaj(A) is the simple-minded majority-finding algorithm that we briefly considered in class. In class we discussed in detail two other, conceptually more complex but much more efficient, majority-finding algorithms.)

Question 4. (15 marks) Consider the recursive program Min(A, f, ℓ) below, where A is an array of integers, and f, ℓ are indices in A such that $f \leq \ell$.

```
Min(A, f, ℓ)
    if f = ℓ then
        return f
    else
        m := (f + ℓ) div 2
        x := Min(A, f, m)
        y := Min(A, m + 1, ℓ)
        if A[y] ≤ A[x] then
            return y
        else
            return x
        end if
    end if
```

a. (5 marks) Prove that Min(A, f, ℓ) is correct with respect to the following precondition/postcondition pair. (Note that A may contain several occurrences of the same integer in different positions.)

Precondition: A is a nonempty array of integers, and f, ℓ are integers such that $1 \leq f \leq \ell \leq \mathit{length}(A)$

Postcondition: Min(A, f, ℓ) returns an integer u such that (i) $f \leq u \leq \ell$, (ii) $A[u]$ is the smallest value in $A[f..\ell]$, and (iii) $A[u] < A[i]$ for every i such that $u < i \leq \ell$.

b. (3 marks) Let $C(n)$ be the number of comparisons between elements of the array that Min(A, f, ℓ) performs when $n = \ell - f + 1$ (i.e., n is the length of the subarray $A[f..\ell]$). Give a recurrence for $C(n)$.

c. (3 marks) Find an exact closed-form formula for $C(n)$ when n is a power of 2. Show your work.

d. (4 marks) Prove that there are positive constants a, b such that for *any* $n \in \mathbb{N}$ (not only values of n that are powers of 2), $a \cdot f(n) \leq C(n) \leq b \cdot f(n)$, where $f(n)$ is the closed-form formula for $C(n)$ that you obtained in part (c) of this question.

continued

Question 5. (15 marks) Define the function $f : \mathbb{N} \to \mathbb{N}$ as follows:

$$f(n) = \begin{cases} 1, & \text{if } 0 \leq n \leq 2 \\ 17f(\lfloor \frac{n}{3} \rfloor) + n^2, & \text{if } n \geq 3 \end{cases}$$

a. (7 marks) Give particular constants c and n_0 and prove that $f(n) \leq c \cdot n^{\log_3 17}$ for all $n \geq n_0$. (Hint: Find positive constants c and d such that you can prove $f(n) \leq c \cdot n^{\log_3 17} - d \cdot n^2$ for large enough n. It may be useful to prove that $\frac{n}{4} \leq \lfloor \frac{n}{3} \rfloor \leq \frac{n}{3}$ for all $n \geq 6$.)

b. (5 marks) Give a particular constant c and prove that $f(n) \geq c \cdot n^{\log_3 17}$ for every $n \geq 1$ that is a power of 3.

c. (3 marks) Give a particular constant c and prove that $f(n) \geq c \cdot n^{\log_3 17}$ for every $n \geq 0$. (Hint: Use part (b), and the fact (which you don't have to prove) that f is nondecreasing.)

Cover page for
CSCB36 Homework #2

Submitted by

(1) Family Name: ____________________

Given Name: ____________________

Student Number: ____________________

Tutorial: ____________________

(2) Family Name: ____________________

Given Name: ____________________

Student Number: ____________________

Tutorial: ____________________

By virtue of submitting this homework I/we acknowledge that I am/we are aware of the policy on homework collaboration for this course.