Computer Science CSC 238F — October 21 1998
St. George Campus — University of Toronto

Solutions to Homework Assignment #2

1. $f := 1$
$l := n$
loop
  exit when $f = l$
  $m := \left\lfloor \frac{f+l+1}{2} \right\rfloor$
  if $A(m) \leq x$ then $l := m - 1$
  else $f := m$
end loop
if $A(f) > x$ then print f
else print "failure"

The notation v_k will be used to denote the value of the variable v after k iterations of the loop.

Invariant:
(a) $1 \leq f_k \leq l_k \leq n$, and
(b) if there is an index i with $1 \leq i \leq n$ and $A(i) > x$, then the largest such value of i is between f_k and l_k (inclusive).

For $k \in \mathbb{N}$, let $P(k)$ be the statement "If there are k complete iterations of the loop, then the invariant is true after k iterations". We shall prove by induction that $P(k)$ holds for all $k \in \mathbb{N}$.

Basis: When $k = 0$, the invariant is true, since $f_0 = 1$ and $l_0 = n$.

Inductive Step: Let k be any positive integer. Suppose $P(k-1)$ is true. We must show that $P(k)$ is also true. Assume there is a kth iteration of the loop.

By the induction hypothesis, $l_{k-1} \geq f_{k-1}$. Because there is a kth iteration, we know that $f_{k-1} \neq l_{k-1}$. So, $l_{k-1} \geq f_{k-1} + 1$. Hence,

$$\begin{aligned}
m_k &= \left\lfloor \frac{f_{k-1} + l_{k-1} + 1}{2} \right\rfloor \\
&\geq \left\lfloor \frac{f_{k-1} + (f_{k-1} + 1) + 1}{2} \right\rfloor \\
&= \left\lfloor \frac{2(f_{k-1} + 1)}{2} \right\rfloor \\
&= f_{k-1} + 1 \\
&\geq 1 \quad \text{(by the induction hypothesis).}
\end{aligned}$$

We also have

$$\begin{aligned}
m_k &= \left\lfloor \frac{f_{k-1} + l_{k-1} + 1}{2} \right\rfloor \\
&\leq \left\lfloor \frac{2l_{k-1} + 1}{2} \right\rfloor \quad \text{(by the induction hypothesis)} \\
&= l_{k-1} \quad \text{(since } l_{k-1} \text{ is an integer)} \\
&\leq n \quad \text{(by the induction hypothesis).}
\end{aligned}$$

Case I: $A(m_k) \leq x$.
Then, $f_k = f_{k-1}$ and $l_k = m_k - 1$. But it was shown above that $1 \leq f_{k-1} \leq m_k - 1 \leq n$, so part (a) of the invariant is true. To prove part (b), suppose there is an index i for which $A(i) > x$. Then the largest such i is between f_{k-1} and l_{k-1} (inclusive), by the induction hypothesis. However, we know that $A(m_k) \leq x$. Hence, all elements of $A(i)$ with $i > m_k$ are also less than or equal to x, since the elements of A are sorted in reverse order. Thus, the largest index i such that $A(i) > x$ must lie between $f_{k-1} = f_k$ and $m_k - 1 = l_k$ (inclusive).

Case II: $A(m_k) > x$.
Then, $f_k = m_k$ and $l_k = l_{k-1}$. It was shown above that $1 \leq m_k \leq l_{k-1} \leq n$, so part (a) of the invariant is true. To prove part (b), suppose there is an index i for which $A(i) > x$. Then the largest such i is between f_{k-1} and l_{k-1} (inclusive), by the induction hypothesis. However, we know that $A(m_k) > x$, so the largest index i such that $A(i) > x$ must lie between $m_k = f_k$ and $l_{k-1} = l_k$ (inclusive).

This completes the proof by induction that $P(k)$ is true for all $k \in \mathbb{N}$.

Now we shall prove that the loop terminates. Suppose the loop never terminates in order to derive a contradiction. Let $x_k = l_k - f_k$. This is always non-negative, by the invariant we proved above. For $k > 0$, the value of x_k is either $m_k - 1 - f_{k-1} \leq l_{k-1} - f_{k-1} - 1 < x_{k-1}$ or $l_{k-1} - m_k \leq l_{k-1} - (f_{k-1} + 1) < x_{k-1}$. So, the x_k's form a strictly decreasing, infinite sequence of natural numbers, which contradicts the well-ordering principle. Thus, the loop must terminate.

We shall now show that the postconditions are satisfied when the programme terminates. Clearly, n, x and A are never changed by the programme. When the programme terminates, $f = l$. If there is an index i such that $A(i) > x$, then $f = l$ is the largest such index (by the invariant (b)), and the programme will output the correct value, f. If there is no such index, then the condition $A(f) > x$ will be false, and the programme will output "failure".

2. Invariant: $s_k \in \mathbb{Z}^+$, $r_k \in \mathbb{N}$ and $r_k^2 \leq x < (r_k + s_k)^2$.

For $k \in \mathbb{N}$, let $P(k)$ be the statement "If there are k complete iterations of the loop, then the invariant is true after k iterations". We shall prove by induction that $P(k)$ holds for all $k \in \mathbb{N}$.

Basis: When $k = 0$, the invariant is true, since $r_0 = 0$, $s_0 \in \mathbb{N}$ and $s_0^2 > x \geq 0$ (by the preconditions).

Inductive Step: Let k be any positive integer. Suppose $P(k-1)$ is true. We must show that $P(k)$ is also true. Assume there is a kth iteration of the loop.

By the induction hypothesis, $s_{k-1} \in \mathbb{Z}^+$, so $s_k = \lceil s_{k-1}/2 \rceil$ is also in $\mathbb{Z}^+$. Clearly, $r_k \in \mathbb{N}$.

Case I: $(r_{k-1} + s_k)^2 \leq x$.
Then, $r_k = r_{k-1} + s_k$. So, $r_k^2 \leq x$ and

$$\begin{aligned}
(r_k + s_k)^2 &= (r_{k-1} + 2s_k)^2 \\
&\geq (r_{k-1} + s_{k-1})^2 \quad (\text{since } 2s_k = 2\lceil s_{k-1}/2 \rceil \geq s_{k-1}) \\
&> x \quad (\text{by induction hypothesis}).
\end{aligned}$$

Case II: $(r_{k-1} + s_k)^2 > x$.
Then, $r_k = r_{k-1}$. So, $r_k^2 = r_{k-1}^2 \leq x$ and $(r_k + s_k)^2 = (r_{k-1} + s_k)^2 > x$.

This completes the proof by induction that $P(k)$ is true for all $k \in \mathbb{N}$.

Now we shall prove that the loop terminates. Suppose the loop never terminates in order to derive a contradiction. Thus, $s_k \geq 2$ for all natural numbers k. For $k > 0$, $s_k = \lceil s_{k-1}/2 \rceil \leq (s_{k-1}+1)/2 \leq s_{k-1} - 1$ (since $s_{k-1} \geq 2$). Thus, the s_k's form a strictly decreasing, infinite sequence of positive integers, which contradicts the well-ordering principle. Thus, the loop must terminate.

We shall now show that the postconditions are satisfied when the programme terminates. When the programme terminates, $s = 1$, so we have $r \in \mathbb{N}$ and $r^2 \le x < (r+1)^2$. I.e. $r = \lfloor \sqrt{x} \rfloor$.

3. First, let's get an upper bound on the values of the variables that are used during the programme. We know that $s_0 \le 2^n$, and it was shown above that the s_k's form a strictly decreasing sequence. Thus, during all iterations, the value of s has at most $n+1$ bits. By the invariant above, $r_k \le r_k^2 \le x$, so the value of r always has at most n bits. Thus, each of the additions and comparisons can be executed in $O(n)$ time, and the multiplication statement can be executed in $O(n^2)$ time. Thus, each iteration of the loop can be executed in $O(n^2)$ time.

Next, we show that there are at most n iterations of the loop. It is given that s is initially a power of 2, say 2^l, where $l \le n$. It is easy to see that after k iterations of the loop, $s_k = 2^{l-k}$. Hence, after l iterations, the loop terminates.

Therefore the entire programme runs in time $O(n^3)$.

4.

(a) Suppose $f \in O(g)$ and $g \in O(h)$. Then, by definition, there exist $c, c' \in \mathbb{R}^+$ and $n_0, n_0' \in \mathbb{N}$ such that
$f(n) \le cg(n)$ for all $n \ge n_0$ and
$g(n) \le c'h(n)$ for all $n \ge n_0'$.
Thus, for all $n \ge \max(n_0, n_0')$,
$f(n) \le cc'h(n)$.
Thus, $f \in O(h)$.

(b) ($\Rightarrow$) Suppose $g \in O(f)$. Then, there exist $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that for all $n \ge n_0$, $g(n) \le cf(n)$. Let $d = 1/c \in \mathbb{R}^+$. Then for all $n \ge n_0$, $f(n) \ge dg(n)$, so $f \in \Omega(g)$.

($\Leftarrow$) Suppose $f \in \Omega(g)$. Then, there exist $d \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that for all $n \ge n_0$, $f(n) \ge df(n)$. Let $c = 1/d \in \mathbb{R}^+$. Then for all $n \ge n_0$, $g(n) \le cg(n)$, so $g \in O(f)$.

(c) Suppose that $f \in O(g)$ in order to derive a contradiction. Then there exist $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that for all $n \ge n_0$,

$$
\begin{aligned}
& f(n) \le cg(n) \\
\Leftrightarrow \quad & n^5/30 \le c \cdot 45n^4 \\
\Leftrightarrow \quad & n \le 1350c.
\end{aligned}
$$

But, when $n = \max(1350c + 1, n_0)$, $n \ge n_0$ and $f(n) > cg(n)$. From this contradiction, we conclude that $f \notin O(g)$.