Computer Science B36 October 11, 2006
Scarborough Campus University of Toronto

Solutions for Homework Assignment #2

Answer to Question 1. Let $P(k)$ be the predicate: "If the loop is executed at least k times, then i_k is an even natural number and $i_k \leq d$." Using induction we prove that $P(k)$ holds for all $k \in \mathbb{N}$.

BASIS: There are two cases.

CASE 1. m is an even natural number. Then by the program $i_0 = m$ and so i_0 is an even natural number. By the program $d = 2m$. Since, $m \leq 2m$ for every $m \in \mathbb{N}$, it follows that $i_0 \leq d$.

CASE 2. m is an odd natural number. Then by the program $i_0 = m + 1$ and so i_0 is an even natural number. By the program $d = 2m$. Since m is an odd natural number, $m \geq 1 \Rightarrow m+1 \leq 2m$. Thus, $i_0 \leq d$.

INDUCTION STEP: Let ℓ be an arbitrary natural number, and suppose that $P(\ell)$ holds. We must prove that $P(\ell + 1)$ also holds. Assume that the loop is executed at least $\ell + 1$ times (otherwise $P(\ell + 1)$ holds trivially). By the induction hypothesis i_ℓ is even; by the program $i_{\ell+1} = i_\ell + 2$, so $i_{\ell+1}$ is even. Since the loop is executed $\ell + 1$ times, by the loop exit condition, $i_\ell \neq d$; by the induction hypothesis, $i_\ell \leq d$. Thus, $i_\ell < d$. Since i_ℓ and d are both even (the former by the induction hypothesis, and the latter because $d = 2m$), and since $i_\ell < d$, it follows that $i_\ell \leq d - 2$. Thus, $i_{\ell+1} = i_\ell + 2 \leq d$. Therefore, $P(\ell + 1)$ holds.

Thus, for each $k \in \mathbb{N}$ such that the loop is executed at least k times, the quantity $d - i_k$ takes on nonnegative integer values. Furthermore, this quantity decreases in each iteration (because $d - i_{k+1} = d - (i_k + 2) < d - i_k$). By the Well-Ordering Principle, the loop terminates and therefore so does the program.

Answer to Question 2.

Loop Invariant Lemma. For each $k \in \mathbb{N}$, if the loop is executed at least k times then

(a) $i_k \leq \mathit{length}(A) + 1$,

(b) m_k is the minimum of $A[1..i_k - 1]$, and

(c) s_k is the second minimum of $A[1..i_k - 1]$.

PROOF THAT LI $\Rightarrow$ PARTIAL CORRECTNESS. Suppose the program terminates. Thus, the loop terminates after some number, say t, of iterations. By the loop exit condition and part (a) of the LI, $i_t = \mathit{length}(A) + 1$. By part (c) of the LI, s_t is the second minimum of $A[1..i_t - 1] = A[1..\mathit{length}(A)]$, i.e., of the entire array A. Since s_t is the value returned by SECONDMIN, the postcondition holds.

[Note that part (b) of the LI isn't needed here; it is needed to carry out the induction proof of the Loop Invariant Lemma.]

PROOF OF THE LOOP INVARIANT LEMMA. Let $P(k)$ be the predicate: "If the loop is executed at least k times then

(a) $i_k \leq \mathit{length}(A) + 1$,

(b) m_k is the minimum of $A[1..i_k - 1]$, and

(c) s_k is the second minimum of $A[1..i_k - 1]$."

We'll use induction to prove that $P(k)$ holds for all $k \in \mathbb{N}$.

BASIS: By the precondition, $\mathit{length}(A) \geq 2$. So the positions $A[1]$ and $A[2]$ of the array exist. By the program $m_0 = \min(A[1], A[2])$, $s_0 = \max(A[1], A[2])$, and $i_0 = 3$. Thus $i_0 = 2 + 1 \leq \mathit{length}(A) + 1$, m_0 is the minimum of $A[1..i_0 - 1]$, and s_0 is the second minimum of $A[1..i_0 - 1]$. So, $P(0)$ holds.

INDUCTION STEP: Let ℓ be an arbitrary natural number. Suppose $P(\ell)$ holds. We'll prove that $P(\ell + 1)$ also holds. Assume that the loop is executed at least $\ell + 1$ times (otherwise, $P(\ell+1)$ holds trivially). Thus, by the loop exit condition, $i_\ell \leq \mathit{length}(A)$. By the program $i_{\ell+1} = i_\ell + 1 \leq \mathit{length}(A) + 1$. Thus, part (a) of $P(\ell + 1)$ holds. To prove that parts (b) and (c) of $P(\ell + 1)$ hold we consider three cases:

CASE 1. $A[i_\ell] < m_\ell$. By part (b) of the induction hypothesis, m_ℓ is the min of $A[1..i_\ell - 1]$. Since $A[i_\ell] < m_\ell$, it follows that $A[i_\ell]$ is the minimum of $A[1..i_\ell]$; and since all elements of A are distinct, the minimum of $A[1..i_\ell - 1]$ is the second minimum of $A[1..i_\ell]$. By the program, $m_{\ell+1} = A[i_\ell]$, $s_{\ell+1} = m_\ell$, and $i_{\ell+1} = i_\ell + 1$. Thus, $m_{\ell+1}$ and $s_{\ell+1}$ are, respectively, the minimum and second minimum of $A[1..i_{\ell+1} - 1]$, proving that parts (b) and (c) of $P(\ell + 1)$ hold in this case.

CASE 2. $m_\ell \leq A[i_\ell] < s_\ell$. By parts (b) and (c) of the induction hypothesis, the hypothesis of this case, and the fact that all elements of A are distinct (by the precondition), we have that

$$\text{minimum of } A[1..i_\ell - 1] < A[i_\ell] < \text{second minimum of } A[1..i_\ell - 1].$$

This implies that the minimum of $A[1..i_\ell - 1]$ (which, by part (b) of the induction hypothesis, is m_ℓ) is the same as the minimum of $A[1..i_\ell]$; and that $A[i_\ell]$ is the second minimumm of $A[1..i_\ell]$. By the program, in this case, $m_{\ell+1} = m_\ell$, $s_{\ell+1} = A[i_\ell]$, and $i_{\ell+1} = i_\ell + 1$. Therefore, $m_{\ell+1}$ and is$_{\ell+1}$ are, respectively, the minimum and second minimum of $A[1..i_{\ell+1} - 1]$. Thus, parts (b) and (c) of $P(\ell + 1)$ hold in this case.

CASE 3. $A[i_\ell] \geq m_\ell$ and $A[i_\ell] \geq s_\ell$. By parts (b) and (c) of the induction hypothesis, the hypothesis of this case, and the fact that all elements of A are distinct (by the precondition), we have that

$$\text{minimum of } A[1..i_\ell - 1] < \text{second minimum of } A[1..i_\ell - 1] < A[i_\ell].$$

This implies that the minimum and second minimum of $A[1..i_\ell - 1]$ (which, by parts (b) and (c) of the induction hypothesis are m_ℓ and s_ℓ) are, respectively, the same as the minimum and second minimum of $A[1..i_\ell]$. By the program, in this case, $m_{\ell+1} = m_\ell$, $s_{\ell+1} = s_\ell$, and $i_{\ell+1} = i_\ell + 1$. Therefore, $m_{\ell+1}$ and $s_{\ell+1}$ are, respectively, the minimum and second minimum of $A[1..i_{\ell+1} - 1]$. Thus, parts (b) and (c) of $P(\ell + 1)$ hold in this case.

[Note that in the induction step we need part (b) of $P(\ell)$ to prove part (c) of $P(\ell + 1)$. This is why we had to include part (b) as part of our loop invariant.]

PROOF OF TERMINATION: By part (a) of the LI, the quantity $\mathit{length}(A) + 1 - i_k$ is a nonnegative integer. Since $i_{k+1} = i_k + 1$, the value of this quantity strictly decreases in each iteration. By the Well-Ordering Principle the loop termiantes and so does the program.

Answer to Question 3.

Loop Invariant Lemma. For any $k \in \mathbb{N}$, if the loop is executed at least k times then

(a) $i_k = k$ and $i_k \leq m$, and

(b) if $\mathit{found}_k = \text{true}$ then $A[i_k]$ is the majority element of A.

PROOF THAT LI $\Rightarrow$ PARTIAL CORRECTNESS. Suppose the program terminates. There are two cases.

CASE 1. A has a majority element, say a. Let ℓ be the minimum index such that $A[\ell] = a$. Thus $\ell \leq m$ (otherwise, the number of positions of A that contain a could not exceed $length(A)/2$ and so a would not be the majority element of A). Furthermore, by the minimality of ℓ, for all j such that $1 \leq j < \ell$, $A[j]$ is not the majority element of A. Therefore, by part (b) of the LI, for all j such that $1 \leq j < \ell$, $found_j$ = false. By part (a) of the LI, for all j such that $1 \leq j < \ell$, $i_j = j$; and since $\ell \leq m$, for all such j, $i_j < m$. Thus, by the loop exit condition the loop is executed at least ℓ times. On the ℓth iteration, $\text{OCCUR}(A, i_\ell, length(A), A[i_\ell])$ returns the number of positions of $A[i_\ell..length(A)]$ that contain a. By part (a) of the LI, $i_\ell = \ell$; and so, by the minimality of ℓ, the number of positions of $A[\ell..length(A)]$ that contain a is the number of positions of the entire array A that contain a; since a is the majority element of A, this number exceeds $length(A)/2$. Thus $\text{OCCUR}(A, i_\ell, length(A), A[i_\ell])$ returns a number greater than $length(A)/2$, and so $found_\ell$ = true. By the loop exit condition the loop is not executed $\ell + 1$ times. Since $found_\ell$ = true, the program returns $A[i_\ell] = A[\ell] = a$. Since a is the majority element of A, the postcondition holds.

CASE 2. A has no majority element. By part (b) of the loop invariant, for each $\ell \in \mathbb{N}$ such that the loop is executed at least ℓ times, $found_\ell$ = false. Thus, if the program terminates, it returns $\bot$. Since A contains no majority element, the postcondition holds.

PROOF OF THE LOOP INVARIANT LEMMA. Let $P(k)$ be the predicate: "If the loop is executed at least k times then

(a) $i_k = k$ and $i \leq m$, and

(b) if $found_k$ = true then $A[i_k]$ is the majority element of A."

We'll use induction to prove that $P(k)$ holds for all $k \in \mathbb{N}$.

BASIS: By the program, $i_0 = 0$. By the precondition, $length(A) \geq 1$ and so $i_0 \leq \big(1 + length(A)\big) \,\mathbf{div}\, 2 = m$. So, part (a) of $P(0)$ holds. By the program, $found_0$ = false and so part (b) of $P(0)$ holds trivially.

INDUCTION STEP: Let ℓ be an arbitrary natural number. Suppose $P(\ell)$ holds. We'll prove that $P(\ell + 1)$ also holds. Suppose the loop is executed at least $\ell + 1$ times (otherwise, $P(\ell + 1)$ holds trivially). We have

$$\begin{aligned} i_{\ell+1} &= i_\ell + 1 && \text{[by the program]} \\ &= \ell + 1 && \text{[by the induction hypothesis]} \end{aligned}$$

Since the loop is executed at least $\ell + 1$ times, by the exit condition, $i_\ell < m$, so $i_{\ell+1} \leq m$. Thus, part (a) of $P(\ell + 1)$ holds.

To prove that part (b) of $P(\ell + 1)$ holds, we must show that if $found_{\ell+1}$ = true then $A[i_{\ell+1}]$ is the majority element of A. So, assume that $found_{\ell+1}$ = true. Since the loop is executed at least $\ell + 1$ times, by the loop exit condition, $found_\ell$ = false. Since $found_\ell$ = false and $found_{\ell+1}$ = true, it follows that during iteration $\ell + 1$ the condition of the if statement was true; i.e., $\text{OCCUR}(A, i_{\ell+1}, length(A), A[i_{\ell+1}]) >$ $/length(A)/2$. Since the loop is executed at least $\ell + 1$ times, again by the loop exit condition, $i_\ell < m$. Thus, $i_{\ell+1} \leq m = \big(1 + length(A)\big)/2 \leq length(A)$ (since, by the precondition, $length(A) \geq 1$. Furthermore, $1 \leq \ell + 1 = i_{\ell+1}$. Thus, $1 \leq i_{\ell+1} \leq length(A)$. Therefore, the call $\text{OCCUR}(A, i_{\ell+1}, length(A), A[i_{\ell+1}])$ returns the number of occurrences of $A[i_{\ell+1}]$ in $A[i_{\ell+1}..length(A)]$. Since this exceeds $length(A)/2$, the number of occurrences of $A[i_{\ell+1}]$ in the entire array A also exceeds $length(A)/2$. In other words, $A[i_{\ell+1}]$ is the majority element of A.

PROOF OF TERMINATION: By part (a) of the LI, the quantity $m - i_k$ is a nonnegative integer. Since $i_{k+1} = i_k + 1$, the value of this quantity strictly decreases in each iteration. By the Well-Ordering Principle the loop termiantes and so does the program.

Answer to Question 4.

a. Let A be an integer array. Let $P(n)$ be the following predicate:

$P(n)$: for all integers f and ℓ such that $1 \le f \le \ell \le \mathit{length}(A)$ and $\ell - f + 1 = n$, $\textsc{Min}(A, f, \ell)$ returns an integer u such that $f \le u \le \ell$, and $A[u]$ is the min of $A[f..\ell]$, and $A[u] < A[v]$ for every v such that $u < v \le \ell$.

We will prove by complete induction that $P(n)$ holds for all integers $n \ge 1$. The correctness of Min follows directly from this fact.

Let i be an arbitrary natural number and assume that $P(j)$ holds for all j such that $1 \le j < i$. We'll show that $P(i)$ also holds.

Case 1. $i = 1$. Let f and ℓ be integers such that $1 \le f \le \ell \le \mathit{length}(A)$ and $\ell - f + 1 = i$. Thus, $f = \ell$. The program terminates immediately, returning the value f; clearly $f \le f \le f$, $A[f]$ is the smallest value in $A[f..f]$, and $A[f] < A[v]$ for every v such that $f < v \le f$ (since there is no such v).

Case 2. $i > 1$. Let f and ℓ be integers such that $1 \le f \le \ell \le \mathit{length}(A)$ and $\ell - f + 1 = i$. So $f < \ell$. Let $m = (f + \ell)$ **div** 2. By Lemma 2.2 in the notes, $f \le m < \ell$. Therefore, $1 \le m - f + 1 < i$ and $1 \le \ell - m < i$. So the induction hypothesis tells us that $\textsc{min}(A, f, m)$ returns a value x and $\textsc{min}(A, m+1, \ell)$ returns a value y such that

$f \le x \le m$, $A[x]$ is the min of $A[f..m]$, and $A[x] < A[v]$ for every v s.t. $x < v \le m$.
$m + 1 \le y \le \ell$, $A[y]$ is the min of $A[m+1..\ell]$, and $A[y] < A[v]$ for every v s.t. $y < v \le \ell$.

There are two cases to consider.

Subcase 2(a). $A[y] \le A[x]$. In this case, the program halts and returns the value y. Since $f < m+1 \le y \le \ell$, we have $f \le y \le \ell$. Since $A[x]$ is the min of $A[f..m]$, $A[y] \le A[x]$, and $A[y]$ is the min of $A[m+1..\ell]$, it follows that $A[y]$ is the min of $A[f..\ell]$. Lastly, from the induction hypothesis, $A[y] < A[v]$ for every v such that $y < v \le \ell$. Thus, $P(i)$ holds in this case.

Subcase 2(b). $A[x] < A[y]$. In this case, the program halts and returns the value x. Since $f \le x \le m < \ell$, we have $f \le x \le \ell$. Since $A[x]$ is the min of $A[f..m]$, $A[x] < A[y]$, and $A[y]$ is the min of $A[m+1..\ell]$, it follows that $A[x]$ is the min of $A[f..\ell]$. Lastly, since $A[x] < A[y]$, we have that $A[x] < A[v]$ for every v such that $m + 1 \le v \le \ell$; in addition (from the induction hypothesis) $A[x] < A[v]$ for every v such that $x < v \le m$. Therefore, $A[x] < A[v]$ for every v such that $x < v \le \ell$. Thus, $P(i)$ holds in this case.

b.

$$C(n) = \begin{cases} 0, & \text{if } n = 1 \\ C(\lceil n/2 \rceil) + C(\lfloor n/2 \rfloor) + 1, & \text{if } n > 1 \end{cases}$$

This reflects the fact that when $\textsc{Min}(A, f, \ell)$ operates on a subarray of length 1 (i.e., when $f = \ell$), it performs no comparisons between array elements; and that when it operates on subarrays of length $n > 1$, it performs one comparison between array elements (namely, the comparison between $A[x]$ and $A[y]$) in addition to those performed by each of the two recursive calls. In the latter case, the two recursive calls operate on subarrays of length $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$.

c. If n is a power of 2, the above recurrence simplifies to

$$C(n) = \begin{cases} 0, & \text{if } n = 1 \\ 2C(n/2) + 1, & \text{if } n > 1 \end{cases}$$

By repeated substitution we get

$$\begin{aligned}
C(n) &= 2C(n/2) + 1 \\
&= 2\big(2C(n/2^2) + 1)\big) + 1 \\
&= 2^2C(n/2^2) + 2 + 1 \\
&= 2^2\big(2C(n/2^3) + 1)\big) + 2 + 1 \\
&= 2^3C(n/2^3) + 2^2 + 2 + 1 \\
&= \cdots \\
&= 2^iC(n/2^i) + 2^{i-1} + \cdots + 2^2 + 2 + 1 \\
&= 2^iC(n/2^i) + (2^i - 1)
\end{aligned}$$

for any positive integer $i \leq \log_2 n$. Taking $i = \log_2 n$, we have

$$C(n) = 2^{\log_2 n}C(1) + (2^{\log_2 n} - 1) = n - 1.$$

d. First we prove that $C(n)$ is nondecreasing. Let $P(n)$ be the predicate: "for every positive integer $m < n$, $C(m) \leq C(n)$". We'll use complete induction to prove that $C(n)$ holds for every positive integer n.

CASE 1. $n \in \{1, 2\}$. $P(1)$ is trivially true (since there is no positive ineger $m < 1$). $C(1) = 0$ and $C(2) = 1$, so $P(2)$ is also true.

CASE 2. Consider an arbitrary positive integer $n > 2$. Assume that for every integer j such that $1 \leq j < n$, $P(j)$ holds. We will prove that $P(n)$ also holds. By $P(n-1)$, it suffices to prove that $C(n-1) \leq C(n)$. Since $n > 2$, we have that $1 \leq \lfloor n/2 \rfloor \leq \lceil n/2 \rceil < n$, and therefore $P(\lfloor n/2 \rfloor)$ and $P(\lceil n/2 \rceil)$ both hold.

$$\begin{aligned}
C(n-1) &= C(\lceil (n-1)/2 \rceil) + C(\lfloor (n-1)/2 \rfloor) + 1 && \text{[by def of } C(n-1)\text{, since } n-1 > 1] \\
&\leq C(\lceil n/2 \rceil) + C(\lfloor n/2 \rfloor) + 1 && \text{[by the I.H. } P(\lceil n/2 \rceil) \text{ and } P(\lfloor n/2 \rfloor)] \\
&= C(n) && \text{[by def of } C(n)\text{, since } n > 1]
\end{aligned}$$

For any positive integer n, let $\hat{n}$ be the smallest power of 2 that is greater than or equal to n. That is, $\hat{n}$ is the unique power of 2 such that

$$\hat{n}/2 < n \leq \hat{n} \tag{1}$$

Since $C(n)$ is nondecreasing,

$$C(\hat{n}/2) \leq C(n) \leq C(\hat{n}) \tag{2}$$

Since $\hat{n}$ and therefore $\hat{n}/2$ are powers of 2, by part (c) and (2) we have

$$\hat{n}/2 - 1 \leq C(n) \leq \hat{n} - 1 \tag{3}$$

Consider any positive integer $n \geq 2$. By the first inequality of (1), $\hat{n}/2 < n$ and so

$$\hat{n} < 2n \;\Rightarrow \hat{n} - 1 < 2n - 1 \leq 2n - 2.$$

Therefore, by the second inequality of (3), $C(n) \leq 2(n-1)$. For $n = 1$ we have $C(1) = 0 \leq 2 \cdot (1-1)$. Thus, for every positive integer n, $C(n) \leq 2(n-1)$.

Consider any positive integer $n \geq 3$. By the second inequality of (1), $\hat{n} \geq n \geq 3$, and so $(\hat{n}-2)/2 \geq (\hat{n}-1)/4 \geq (n-1)/4$. Therefore, by the first inequality of (3), $C(n) \geq (n-1)/4$. For $n = 1, 2$ we have $C(1) = 0 \geq (1-1)/4$ and $C(2) = 1 \geq (2-1)/4$. Thus, for every positive integer n, $C(n) \geq (n-1)/4$.

We have therefore shown that for every positive integer n, $(n-1)/4 \le C(n) \le 2(n-1)$. Thus, choosing $a = 1/4$ and $b = 2$ we get positive constants with the desired properties.

Answer to Question 5.

a. It will be useful to establish that for all integers $n \ge 6$, $\frac{n}{4} \le \lfloor \frac{n}{3} \rfloor \le \frac{n}{3}$. Clearly $\lfloor \frac{n}{3} \rfloor \le \frac{n}{3}$. It is also easy to see that $\frac{n-2}{3} \le \lfloor \frac{n}{3} \rfloor$, and that for $n \ge 8$, $\frac{n}{4} \le \frac{n-2}{3}$. It remains to prove that for $n \in \{6, 7\}$, $\frac{n}{4} \le \lfloor \frac{n}{3} \rfloor$; this can be checked directly.

Let $c = 11$ and $d = 16$ and $n_0 = 2$. We will show that for all integers $n \ge n_0$, $f(n) \le c \cdot n^{\log_3 17} - d \cdot n^2$ for all $n \ge n_0$. It follows that for all $n \ge 2$, $f(n) \le 11 \cdot n^{\log_3 17}$.
(Note: We figured out these constants c, d, n_0 by *first* doing the proof, and seeing what constants would work.)
Consider the predicate

$P(n): \quad f(n) \le c \cdot n^{\log_3 17} - d \cdot n^2$

We will use complete induction to show that $P(n)$ holds for all integers $n \ge n_0$. Let i be an integer $\ge n_0$, and assume that $P(j)$ holds for every integer j, $n_0 \le j < i$. We want to show $P(i)$.

CASE 1. $i \ge 6$. (These are the induction cases.)
Since $i \ge 3$ we have

$$f(i) = 17 f(\lfloor \frac{i}{3} \rfloor) + i^2 \tag{4}$$

Since $i > 0$, $\frac{i}{3} < i$, so $\lfloor \frac{i}{3} \rfloor < i$; since $i \ge 6$, $\lfloor \frac{i}{3} \rfloor \ge 2 = n_0$. Since we have $n_0 \le \lfloor \frac{i}{3} \rfloor < i$, the induction hypothesis tells us that $P(\lfloor \frac{i}{3} \rfloor)$ holds, so

$$f(\lfloor \frac{i}{3} \rfloor) \le c \cdot \lfloor \frac{i}{3} \rfloor^{\log_3 17} - d \cdot \lfloor \frac{i}{3} \rfloor^2 \tag{5}$$

Since $i \ge 6$, we have $\frac{i}{4} \le \lfloor \frac{i}{3} \rfloor \le \frac{i}{3}$. It therefore follows from (5) that

$$f(\lfloor \frac{i}{3} \rfloor) \le c \cdot (\frac{i}{3})^{\log_3 17} - d \cdot (\frac{i}{4})^2 \tag{6}$$

and simplifying (6) we get

$$f(\lfloor \frac{i}{3} \rfloor) \le c \cdot \frac{i^{\log_3 17}}{17} - d \cdot \frac{i^2}{16} \tag{7}$$

Combining (4) and (7) we get

$f(i) \le 17(c \cdot \frac{i^{\log_3 17}}{17} - d \cdot \frac{i^2}{16}) + i^2 = c \cdot i^{\log_3 17} - (\frac{17}{16} \cdot d - 1)i^2$. Since our goal is to show $P(i)$, namely $f(i) \le c \cdot i^{\log_3 17} - d \cdot i^2$, it suffices to show $-(\frac{17}{16} \cdot d - 1)i^2 \le -d \cdot i^2$, so it suffices to show $(\frac{17}{16} \cdot d - 1)i^2 \ge d \cdot i^2$, which follows from $\frac{17}{16} \cdot d - 1 \ge d$, which follows from $\frac{d}{16} \ge 1$, which is true since $d = 16$.

Remark: We decided to do the induction cases first. It was convenient to be able to assume that $i/4 \le \lfloor i/3 \rfloor$, and for this reason we chose $i \ge 6$ to be the induction cases. In order to be able to apply the induction hypothesis, we needed $\lfloor i/3 \rfloor \ge n_0$, and for this reason (since i could be as low as 6) we chose $n_0 = 2$. After doing the proof we saw that we needed $d/16 \ge 1$, and for this reason we chose $d = 16$. It now remains to do the base cases: $i = 2, 3, 4, 5$. We do these below, and the value $c = 11$ was chosen to make these base cases all work out.

CASE 2. $i \in \{2,3,4,5\}$. It is easy to check each of the following:

$f(2) = 1 \leq 11 \cdot 2^{\log_3 17} - 16 \cdot 2^2$
$f(3) = 17f(1) + 3^2 = 17 \cdot 1 + 3^2 = 26 \leq 11 \cdot 3^{\log_3 17} - 16 \cdot 3^2$
$f(4) = 17f(1) + 4^2 = 17 \cdot 1 + 4^2 = 33 \leq 11 \cdot 4^{\log_3 17} - 16 \cdot 4^2$
$f(5) = 17f(1) + 5^2 = 17 \cdot 1 + 5^2 = 42 \leq 11 \cdot 5^{\log_3 17} - 16 \cdot 5^2$

b. In the proof below, we see that it is the base case that determines what constant c we should use; it turns out we can use $c = 1$.

Consider the predicate:

$P(n)$: If n is a power of 3, then $f(n) \geq n^{\log_3 17}$

We will use complete induction to show that $P(n)$ holds for all integers $n \geq 1$. Let $i \geq 1$ be an integer, and assume that $P(j)$ holds for every integer j, $1 \leq j < i$. We want to show $P(i)$. Assume that i is a power of 3 (otherwise $P(i)$ is trivially true).

CASE 1. $i = 1$.
Since $f(1) = 1$ and $1^{\log_3 17} = 1$, $P(1)$ holds.

CASE 2. $i > 1$.
Since i is a power of 3, $i \geq 3$ and divisible by 3 and so
$f(i) = 17f(i/3) + i^2$. $1 \leq i/3 < i$, and so $P(i/3)$ holds. Since $i/3$ is a power of 3,
$f(i/3) \geq (i/3)^{\log_3 17} = (1/17)i^{\log_3 17}$, and so
$f(i) \geq 17(1/17)i^{\log_3 17} + i^2 \geq i^{\log_3 17}$.

c. We will show that $c = \frac{1}{17}$ works. That is, we will show that $f(n) \geq (1/17)n^{\log_3 17}$ for every $n \in \mathbb{N}$.

Let $n \in \mathbb{N}$. If $n = 0$, since $f(0) = 1$, it is easy to check that $f(0) \geq (1/17)0^{\log_3 17}$. So assume that n is positive. Then there is a positive integer m such that m is a power of 3 and $n/3 \leq m \leq n$. Since f is nondecreasing, $f(n) \geq f(m)$. By part (b), $f(m) \geq m^{\log_3 17}$. Since $m \geq n/3$, $m^{\log_3 17} \geq (n/3)^{\log_3 17} = (1/17)n^{\log_3 17}$.

Putting all this together, we have $f(n) \geq (1/17)n^{\log_3 17}$.