CSC 236 October 22, 2004
St George Campus University of Toronto

Solutions for Assignment #2

Answer to Question 1.

a. Let A, i, j, k be as stated in the question.

To prove the first part, assume that u is such that $FREQ(A, i, j, u) = (j - i + 1)/2$. Since $j \geq i - 1$, the number of integers ℓ such that $i \leq \ell \leq j$ is $j - i + 1$. So for every $u' \neq u$, $FREQ(A, i, j, u') \leq (j - i + 1)/2$. So for every u'', $FREQ(A, i, j, u'') \leq (j - i + 1)/2$. So $A[i..j]$ does not have a majority element.

To prove the second part, assume that u is the majority element of $A[i..k]$, and assume that $A[i..j]$ does not have a majority element. We have
$FREQ(A, i, j, u) + FREQ(A, j + 1, k, u) = FREQ(A, i, k, u) > (k - i + 1)/2$, and
$(k - i + 1)/2 = (j - i + 1)/2 + (k - (j + 1) + 1)/2$, so
$FREQ(A, i, j, u) + FREQ(A, j + 1, k, u) > (j - i + 1)/2 + (k - (j + 1) + 1)/2$. We also have
$FREQ(A, i, j, u) \leq (j - i + 1)/2$ (since A[i..j] doesn't have a majority element), so
$FREQ(A, j + 1, k, u) > (k - (j + 1) - 1)/2$, so u is the majority element of $A[j + 1..k]$.

b. Assume the Precondition holds: $m \geq 1$ and $A[1..m]$ has a majority element.
We first state a loop invariant lemma.

Loop Invariant Lemma: At every visit to the **while** test:

- $0 \leq j < x \leq m$
- $A[1..j]$ does not have a majority element.
- $c = FREQ(A, j + 1, x, u) > (x - (j + 1) + 1)/2$.

Before proving the Lemma, we show how to use it to prove partial correctness and termination.

To prove termination: By the LIL, $m - x$ is always ≥ 0 after every completion of the loop. By inspecting the code, we see that x increases (by either 1 or 2) each time through the loop, and so $m - x$ decreases each time through the loop. So by the Termination Principle, the program halts.

To prove Partial Correctness: Assume the program halts. Then $x = m$. By the LIL we have $m > j \geq 0$, and $A[1..j]$ does not have a majority element. The second part of Part **a** tells us that the majority element of $A[1..m]$ (which, by the Precondition exists) must be the majority element of $A[j + 1..m]$, and by the LIL, this is u. So the program returns u, the majority element of $A[1..m]$.

Proof of Loop Invariant Lemma: We assume the usual notation, namely that if visit n to the **while** test occurs (that is, if the loop body has been executed n times), then we define j_n, x_n, u_n, c_n to be the values of j, x, u, c at that visit. We define:

$P(n)$: If visit n to the test occurs (that is, if the loop body is executed at least n times) then

- $0 \leq j_n < x_n \leq m$
- $A[1..j_n]$ does not have a majority element.
- $c_n = FREQ(A, j_n + 1, x_n, u_n) > (x_n - (j_n + 1) + 1)/2$.

We will use simple induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$.

Basis: We want to show $P(0)$. By inspecting the program we see that:
$j_0 = 0$, $x_0 = 1$, $u_0 = A[1]$, and $c_0 = 1$. By the precondition we have $m \geq 1$. So we have

- $0 \leq 0 < 1 \leq m$
- $A[1..0]$ does not have a majority element.
- $1 = FREQ(A, 1, 1, A[1]) = 1 > (1 - 1 + 1)/2$

Induction Step: Let $i \geq 0$ and assume $P(i)$. We want to show $P(i+1)$. Assume visit $i+1$ to the test occurs, that is, the loop body is executed at least $i+1$ times. So $x_i \neq m$. So from $P(i)$ we have:

- $0 \leq j_i < x_i < m$
- $A[1..j_i]$ does not have a majority element.
- $c_i = FREQ(A, j_i + 1, x_i, u_i) > (x_i - (j_i + 1) + 1)/2$.

¿From examining the program, we see that there are three cases to consider.

Case 1. $A[x_i + 1] = u_i$. Then we have:
$j_{i+1} = j_i$, $x_{i+1} = x_i + 1$, $u_{i+1} = u_i$, and $c_{i+1} = c_i + 1$. So it follows that:

- $0 \leq j_{i+1} < x_{i+1} \leq m$
- $A[1..j_{i+1}]$ does not have a majority element.
- $c_{i+1} = c_i + 1 =$
  $FREQ(A, j_i + 1, x_i, u_i) + 1 = FREQ(A, j_i + 1, x_i + 1, u_i) = FREQ(A, j_{i+1} + 1, x_{i+1}, u_{i+1})$.
  The next to last equality is because $A[x_i + 1] = u_i$. We also have
  $c_{i+1} = c_i + 1 > (x_i - (j_i + 1) + 1)/2 + 1 > (x_i + 1 - (j_i + 1) + 1)/2 = (x_{i+1} - (j_{i+1} + 1) + 1)/2$.

Case 2. $A[x_i + 1] \neq u_i$ and $2c_i \neq x_i + 1 - j_i$. Then we have:
$j_{i+1} = j_i$, $x_{i+1} = x_i + 1$, $u_{i+1} = u_i$, and $c_{i+1} = c_i$. So it follows that:

- $0 \leq j_{i+1} < x_{i+1} \leq m$
- $A[1..j_{i+1}]$ does not have a majority element.
- $c_{i+1} = c_i = FREQ(A, j_i + 1, x_i, u_i) = FREQ(A, j_i + 1, x_i + 1, u_i) = FREQ(A, j_{i+1} + 1, x_{i+1}, u_{i+1})$.
  The next to last equality is because $A[x_i + 1] \neq u_i$. We also have
  $2c_i > x_i - (j_i + 1) + 1$, so $2c_i \geq x_i + 1 - j_i$. Since $2c_i \neq x_i + 1 - j_i$, we have $2c_i > x_i + 1 - j_i$, and so $c_{i+1} > (x_i + 1 - j_i)/2 = (x_{i+1} - (j_{i+1} + 1) + 1)/2$.

Case 3. $A[x_i + 1] \neq u_i$ and $2c_i = x_i + 1 - j_i$. Then we have:
$j_{i+1} = x_i + 1$, $x_{i+1} = x_i + 2$, $u_{i+1} = A[x_i + 2]$, and $c_{i+1} = 1$.

- $0 \leq x_i + 1 < x_i + 2 \leq m + 1$. This isn't quite good enough since we want to show that $x_i + 2 \leq m$. $FREQ(A, j_i + 1, x_i + 1, u_i) = FREQ(A, j_i + 1, x_i, u_i) = c_i = ((x_i + 1) - (j_i + 1) + 1)/2$, so the first part of Part **a** tells us that $A[j_i + 1..x_i + 1]$ does not have a majority element. Since we know $A[1..j_i]$ also does not have a majority element, the second part of Part **a** tells us that $A[1..x_i + 1]$ does not have a majority element. So $x_i + 1 \neq m$, so $x_i + 2 \neq m + 1$, so $x_i + 2 \leq m$.
- We've just shown that $A[1..x_i + 1]$ does not have a majority element, so $A[1..j_{i+1}]$ does not have a majority element.
- $1 = FREQ(A, (x_i + 1) + 1, x_i + 2, A[x_i + 2]) > ((x_i + 2) - (x_i + 2) + 1)/2$.

Answer to Question 2.

a.

Precondition: k, F, L are integers, $0 \le F \le \sqrt{k} < L$.

Postcondition: $\lfloor \sqrt{k} \rfloor$ is returned.

Define $P(n)$ as follows.
$P(n)$: For all integers k, F, L such that $0 \le F \le \sqrt{k} < L$ and $L - F = n$, HELPER(k, F, L) halts and returns $\lfloor \sqrt{k} \rfloor$.

We will prove that $P(n)$ holds for every integer $n \ge 1$ by complete induction.
(This implies correctness of HELPER since the Precondition implies that $L - F \ge 1$.)
Let $i \ge 1$ be an integer such that for all integers j, if $1 \le j < i$ then $P(j)$ holds. We will show $P(i)$ holds.

CASE 1. $i = 1$. Let k, F, L be such that $0 \le F \le \sqrt{k} < L = F + 1$.
Then $F = \lfloor \sqrt{k} \rfloor$, and since $L = F + 1$, the program returns F.

CASE 2. $i \ge 2$. Let k, F, L be such that $0 \le F \le \sqrt{k} < L$ and $L - F = i$.
Since $L \ne F + 1$, the program assigns to m the value $(F + L)$ div 2. Since $L - F > 1$, one can prove that $F < m < L$. We have two subcases.

CASE 2.1 $m \le \sqrt{k}$.
Then $1 \le L - m < L - F = i$ and $0 \le m \le \sqrt{k} < L$. So by the induction hypothesis, the call to HELPER(k, m, L) halts and returns $\lfloor \sqrt{k} \rfloor$.

CASE 2.2 $m > \sqrt{k}$.
Then $1 \le m - F < L - F = i$ and $0 \le F \le \sqrt{k} < m$. So by the induction hypothesis, the call to HELPER(k, F, m) halts and returns $\lfloor \sqrt{k} \rfloor$.

b. Let $k \in \mathbb{N}$. Then $0 \le 0 \le \sqrt{k} < k + 1$. So the call to HELPER$(k, 0, k + 1)$ returns $\lfloor \sqrt{k} \rfloor$.

Answer to Question 3. Joe is wrong. His program does not meet the specification.

Consider the following example: Mergesort$(A, 1, 4)$ where $A[1] = A[2] = A[3] = A[4] = 1$. The first recursive call to Mergesort$(A, 1, 2)$ doesn't change the array at all. We will consider the second recursive call in the next paragraph. Note that the last statement we added has no effect since $f = 1$.

Now consider the second recursive call Mergesort$(A, 3, 4)$. The recursive calls to Mergesort$(A, 3, 3)$ and Mergesort$(A, 4, 4)$ do not change the array at all. But the last statement causes $A[1]$ to get assigned 0.

So the net effect is that $A[1]$ is changed to 0.

Answer to Question 4.

a. (NOTE: By going through the proof in advance, we were able to determine how big n would have to be in order for the induction step to go through, and hence what the base cases should be, and hence what value we should choose for c.)

Let $c = 13$. Define:

$P(n): \quad f(n) \leq cn^3 - 5n^2$.

We will use complete induction to prove that $P(n)$ holds for all integers $n \geq 1$.
Let $i \geq 1$ be an integer; assume that $P(j)$ holds for all integers j such that $1 \leq j < i$.
We want to show $P(i)$.

Case 1. (induction cases) $i \geq 4$.
Since $1 \leq \lfloor i/2 \rfloor < i$, $P(\lfloor i/2 \rfloor)$ holds, so $f(\lfloor i/2 \rfloor) \leq c\lfloor i/2 \rfloor^3 - 5\lfloor i/2 \rfloor^2$.
Since $(i-1)/2 \leq \lfloor i/2 \rfloor \leq i/2$, we have $f(\lfloor i/2 \rfloor) \leq c(i/2)^3 - 5((i-1)/2)^2 = ci^3/8 - (5/4)(i-1)^2$.
Since $i \geq 1$, we have $f(i) = 8f(\lfloor i/2 \rfloor) + (i-1)^2 \leq 8(ci^3/8 - (5/4)(i-1)^2) + (i-1)^2 = ci^3 - 9(i-1)^2$.
In order to show that $f(i) \leq ci^3 - 5i^2$, it suffices to show that $ci^3 - 9(i-1)^2 \leq ci^3 - 5i^2$. So it suffices to show that $((i-1)/i)^2 \geq 5/9$. Since $(i-1)/i$ increases as i increases (for $i \geq 1$), it suffices to show that $((4-1)/4)^2 \geq 5/9$, and the reader can check that this is true.

Case 2. (base cases) $1 \leq i \leq 3$. We compute
$f(1) = 8f(0)+0 = 8 \leq 13\cdot 1 - 5$; $\quad f(2) = 8f(1)+1 = 65 \leq 13\cdot 8 - 20$; $\quad f(3) = 8f(1)+4 = 68 \leq 13\cdot 27 - 45$.

b. We will show that $f(2^k) > 8 \cdot (2^k)^3 = 8^{k+1}$ for all integers $k \geq 1$. Define

$P(k): \quad f(2^k) > 8^{k+1}$.

We will use simple induction to prove that $P(k)$ holds for every integer $k \geq 1$.

Basis: $k = 1$.
$f(2^1) = 65 > 8^2$.

Induction Step: Let $k \geq 1$ and assume $P(k)$ holds. We will show $P(k+1)$.
$f(2^{k+1}) = 8f(2^k) + (2^{k+1} - 1)^2 \geq 8f(2^k) > 8(8^{k+1}) = 8^{k+2}$.

c. We will show that $f(3 \cdot 2^k) < 3 \cdot (3 \cdot 2^k)^3 - (3 \cdot 2^k)^2 = 81 \cdot 8^k - 9 \cdot 4^k$ for all integers $k \geq 1$. Define

$P(k): \quad f(3 \cdot 2^k) < 81 \cdot 8^k - 9 \cdot 4^k$.

We will use simple induction to prove that $P(k)$ holds for every integer $k \geq 0$.

Basis: $k = 0$.
$f(3 \cdot 2^0) = 68 < 81 \cdot 8^0 - 9 \cdot 4^0$.

Induction Step: Let $k \geq 0$ and assume $P(k)$ holds. We will show $P(k+1)$.
$f(3 \cdot 2^{k+1}) = 8f(3 \cdot 2^k) + (3 \cdot 2^{k+1} - 1)^2 < 8(81 \cdot 8^k - 9 \cdot 4^k) + 9 \cdot 4^{k+1} = 81 \cdot 8^{k+1} - 9 \cdot 4^{k+1}$.