Homework 2 sample solution

Due 08/31/16

August 25, 2016

1. Prove that $sum = i^2$ after every iteration of the for loop below:

```
Input: n: nonnegative integer
Output: n^2
sum = 0
for i = 1 to n do
    sum = sum + 2i − 1
end
return sum
```

Answer:

Proof. (1st iteration) Before the loop, $sum = 0$. In the first iteration, $i = 1$, so sum becomes $sum + 2i - 1 = 0 + 2(1) - 1 = 1$. Since $i^2 = 1$, $sum = i^2$ after the first iteration of the for loop.

(k^{th} iteration) Suppose that $sum = k^2$ after the k^{th} iteration of the for loop. In iteration $k + 1$, $i = k + 1$, so sum becomes $sum + 2i - 1 = k^2 + 2(k + 1) - 1 = k^2 + 2k + 1 = (k + 1)^2$.

Therefore, by induction, $sum = i^2$ after every iteration of the for loop. $\square$

2. Use strong induction on n to prove that the algorithm below computes 2^n for all $n \geq 0$.

Answer:

Proof. (*Base case*) When $n = 0$, QuickPow returns 1, and $2^n = 1$, so QuickPow is correct for $n = 0$.

(*Inductive step*) Suppose that QuickPow is correct for all n from 0 up to k, for some $k \geq 0$. We prove that QuickPow is correct when $n = k + 1$ in two cases: when n is even, and when n is odd.

(*n is even*) If n is even, $n/2$ is an integer. Moreover, $n/2 > 0$ since $n \geq 1$ and $n/2 < k + 1$, so $0 < n/2 < k + 1$. Therefore, QuickPow$(n/2)$ will return $2^{n/2}$ by the inductive hypothesis, which will become t. $t^2 =$

```
Input: n: nonnegative integer
Output: 2^n
Algorithm: QuickPow
if n = 0 then
    return 1
else if n is even then
    t = QuickPow(n/2)
    return t^2
else
    t = QuickPow((n − 1)/2)
    return 2t^2
end
```

$(2^{n/2})^2 = 2^{2(n/2)} = 2^n$, so QuickPow will return 2^n for $n = k + 1$ when n is even.

(n *is odd*) If n is odd, $n - 1$ is even, so $(n - 1)/2$ is odd. Moreover, $n \geq 1$, so $(n - 1)/2 \geq 0$ and $(n - 1)/2 < n$, so $0 \leq (n - 1)/2 < n$. Therefore, QuickPow($(n - 1)/2$) will return $2^{(n-1)/2}$ by the inductive hypothesis, which will become t. $2t^2 = 2(2^{(n-1)/2})^2 = 2(2^{2(n-1)/2}) = 2(2^{n-1}) = 2^n$, so QuickPow will return 2^n for $n = k + 1$ when n is odd.

As every integer must be either even or odd, QuickPow returns 2^n for all values of $n = k + 1$. Hence, by induction, QuickPow returns 2^n for all $n \geq 0$. $\square$