CS 331: Algorithms and Complexity (Spring 2024)
Unique numbers: 50930, 50935 50940, 50945

Classwork 1

(a) Prove that $n^3 - n$ is divisible by 3 for all positive integers.
Sol.
Base case: $n = 1$, $1^3 - 1$ is divisible by 3, 0 divisible by 3.

Induction hypothesis: Assume that the formula holds for $n = k$, i.e., $k^3 - k$ is divisible by 3

Induction Step: show that the formula holds for $n = k + 1$, i.e., $(k + 1)^3 - (k + 1)$ is divisible by 3

$$
\begin{aligned}
(k+1)^3 - (k+1) &= (k^3 + 3k^2 + 3k + 1) - (k + 1) \\
&= (k^3 - k) + 3k^2 + 3k \\
&= (k^3 - k) + 3(k^2 + k)
\end{aligned}
$$

But we know that$(k^3 - k)$ is divisible by 3 by induction hypothesis, and
$3(k^2 + k)$ is divisible by 3
Conclusion:$(k + 1)^3 - (k + 1)$ is divisible by 3
Since if a is divisisble by b and c is divisible by b, then $a + c$ is divisible by b

(b) The Fibonacci numbers $F_n \geq 0$ are the numbers of a famous sequence defined by
$F_0 = 0$
$F_1 = 1$
...
$F_n = F_{n-1} + F_{n-2}$.

Prove that for all $n \geq 0$, $F_n < 2^n$.
Sol.
Define $P(n) = F_n < 2^n$.

Base case: We need two base cases since the definition of F_n requires two values.
Base Case 1: $n = 0$. $F_0 = 0$, which is less than $2^0 = 1$.
Base Case 2: $n = 1$. $F_1 = 0$, which is less than $2^1 = 2$.

Induction hypothesis: Assume that the formula holds for $n = k$, i.e., $P(0)$, $P(1)$, ..., $P(k)$ are all true.

Induction Step: show that the formula holds for $n = k+1$, i.e., $P(k+1)$, or $F_{k+1} < 2^{k+1}$.
$F_{k+1} = F_k + F_{k-1} < 2^k + 2^{k-1} < 2^k + 2^k = 2(2^k) = 2^{k+1}$.

By strong induction, for all $n \geq 0$, $F_n < 2^n$.
(c) You were asked to provide an algorithm that finds the maximum value in a given array, A, with n elements. The algorithm is supposed to scan through each location in the array and report the maximum value seen so far. Algorithm **??** is purported to solve this problem. Show that this is indeed the case by providing a mathematical proof of correctness for Algorithm **??**. I.e., show that it finds the maximum value in A. Note: the array index starts from 0.

Algorithm 1: Find maximum in array

```
Initially index = 0 and max = A[index];
while index < n do
    if A[index] > max then
        max = A[index];
    Increase index by 1;
report max;
```

Sol: Base case: If $n = 1$, then the body of the while loop doesn't get executed and we report the maximum value as $A[0]$.

Induction hypothesis: Assume that max stores the maximum value of $A[0..n-2]$ (which covers n-1 elements of the array A).

Induction step: Show that max stores the maximum value of $A[0..n-1]$ (which covers n

elements of the array A).

We know from the induction hypothesis that at the $(n-1)$-th iteration, max stores the maximum value of $A[0..n-2]$. In the n-th and final iteration, if $A[n-1] > \text{max}$, then max will store $A[n-1]$. Otherwise, max will keep the maximum of $A[0..n-2]$ since $A[n-1]$ is not bigger. Therefore, upon termination of the final iteration, max will store the maximum of $A[0..n-1]$.