CS 506, Online Foundations of CS	**Homework 4** **Induction**	Dr. David Nassimi

Study Module 3: Proofs
Proof by Induction and Strong Induction

1. Prove by **simple induction** that every postage amount of 8 cents or more can be achieved by using only 5-cent stamps and 3-cent stamps. That is, prove that for every integer $n \geq 8$, there exist non-negative integers A and B such that:

$$n = 5A + 3B.$$

2. The following "proof by induction" attemps to prove that all horses in universe have the same color. That is, for every $n \geq 1$, any set of n horses in universe have the same color. Obviously, there is something wrong with this proof. State clearly which step is wrong and why.

   Proof:

   (a) For $n = 1$, the set contains a single horse, which has the same color by itself, so the base is obviously correct.

   (b) For the hypothesis, suppose the claim is correct for some $n \geq 1$.

   (c) Then, we will prove the claim is also correct for $n + 1$. Consider any set of $n + 1$ horses. Let us number them as $1, 2, \cdots, n + 1$.

      i. By the hypothesis, horses $\{1, 2, \cdots, n\}$ have the same color.

      ii. By the hypothesis, horses $\{2, 3, \cdots, n + 1\}$ have the same color.

      iii. These two sets have in common horses $\{2, 3, \cdots, n\}$.

      iv. Therefore, by transitivity, these $n + 1$ horses all have the same color.

3. Prove by **induction** that every postage of 24 cents or more can be achieved using only 7-cent stamps and 5-cent stamps. That is, prove that for every integer $n \geq 24$, there exist some **non-negative** integers A and B such that

$$P(n): \quad n = 7A + 5B.$$

   Let $P(n)$ be the predicate "postage of n cents can be achieved." Provide the inductive proof by using each of the following methods.

   (a) **(Simple Induction)** Prove the induction base, $n = 24$. Then, for any $n \geq 24$, prove that if $P(n)$ is true, then $P(n + 1)$ will be true.

   (b) **(Strong Induction)** Prove the induction base cases $(24, 25, \cdots, 30)$. Then for $n \geq 31$, prove that:

$$\text{If } P(m) \text{ is true for all } 24 \leq m < n, \text{ then } P(n) \text{ is also true.}$$

4. Prove each formula by induction.

   (a)

$$S(n) = 1 + 3 + 5 + \cdots + (2n - 1) = n^2$$

   (b)

$$\sum_{i=1}^{n} (\frac{i}{2^i}) = 2 - \frac{n+2}{2^n}$$

5. Prove by induction that for all intergers $n \geq 1$,

$$2^n > n.$$

6. Prove by induction that the following pseudocode computes X^n. (This algorithm computes power by repeated multiplications.)

```
T = 1;
for i = 1 to n {
   T = T * X }
```

Hint: Use induction to prove that after iteration i of the for loop, where $i = 1, 2, 3, \cdots, n$, the result is $T = X^i$. (This is called a loop invariant.) Therefore, after the last iteration, $i = n$ and so $T = X^n$.

7. Prove by induction that the following pseudocode computes X^n, where $n = 2^k$.
(This algorithm is much more efficient than the earlier one. In the earlier agorithm, the power goes up by 1 after each iteration of the for loop, but in this algorithm the power doubles after each iteration.)

```
real Power (real X, int k) {
T = X;
for i = 1 to k {
   T = T * T }
return (T)
}
```

Hint: Use induction to prove that after iteration $i = m$ of the for loop, $T = X^{2^m}$.

8. Consider the following recurrence equation.
Note: The symbol $\lfloor\rfloor$ is the "floor" function. For any real x, $\lfloor x \rfloor$ rounds down x to its nearest integer. For example, $\lfloor 3.1415 \rfloor = 3$. And $\lfloor 3 \rfloor = 3$.

$$f(n) = \begin{cases} 1, & n = 1 \\ f(\lfloor n/2 \rfloor) + n, & n \geq 2 \end{cases}$$

(a) Compute and tabulate $f(n)$ for $n = 1$ to 8.

(b) Prove by induction that the solution has the following bound.

$$f(n) < 2n.$$

Hint: A strong form of induction is needed here. (Don't try to increment n by 1 in your induction step.) To prove the bound for any n, you have to assume the bound is true for all smaller values of n. That is, assume $f(m) < 2m$ for all $m < n$. This strong hypothesis in particular will mean

$$f(\lfloor n/2 \rfloor) < 2\lfloor n/2 \rfloor.$$

Additional Exercises (Not to be handed-in)

9. Consider the following recurrence equation, where n is an integer power of 2. That is $n = 2^k$ for some integer k.

$$f(n) = \begin{cases} 1, & n = 1 \\ f(n/2) + n, & n \geq 2 \end{cases}$$

(a) Compute and tabulate $f(n)$ for $n = 1, 2, 4, 8, 16, 32$.

(b) Prove by induction that the solution is

$$f(n) = 2n - 1.$$

10. Consider the following recurrence equation, where n is an integer power of 2. That is $n = 2^k$ for some integer k.

$$f(n) = \begin{cases} 1, & n = 1 \\ 1, & n = 2 \\ f(n/2) + f(n/4), & n \geq 4 \end{cases}$$

(a) Compute and tabulate $f(n)$ for $n = 1, 2, 4, 8, 16, 32, 64$.

(b) Prove by induction that the solution has the following upper bound.

$$f(n) \leq n, \quad n \geq 1$$

(c) Prove by induction that the solution has the following lower bound.

$$f(n) > \sqrt{n}, \quad n \geq 4$$

11. Prove by Induction:

$$S(n) = \sum_{i=1}^{n} (i^3) = \frac{(n)^2(n+1)^2}{4}$$

12. Prove by induction that an integer is divisible by 3 if and only if sum of its digits is divisible by 3.

Hint: For induction step, add 3 to a given number. What happens if no carry occurs? What happens if one or more positions produce a carry? (The net effect of a carry on sum of the digits is -10+1, which is -9.) For example, consider what happens when you add 3 to each of the following numbers: 384, 348, 198.

13. Prove by induction that an integer is divisible by 9 if and only if sum of its digits is divisible by 9.

14. Let n be the number of nodes in a *tree* and e be the number of edges. Prove by induction on n that

$$e = n - 1$$

.

15. A directed graph (digraph) is *acyclic* if it contains no directed cycle. The *indegree* of a vertex in a digraph is the number of edges coming into the vertex, and the *outdegree* of the vertex is the number of edges going out of it. Prove by **contradiction** each of the following.

(a) An acyclic digraph has at least one vertex with outdegree 0.

(b) An acyclic digraph has at least one vertex with indegree 0.

16. Consider the mergesort algorithm. This is an efficient sorting algorithm bases on divide-and-conquer strategy. This algorithm sorts n elements, where $n > 1$, as follows:

- Divide the array into two halves. (If n is even, then each half has exactly $n/2$ elements. Otherwise, the two halves are of size $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$.)
- Sort each half recursively.
- Merge the two sorted halves.

Let $f(n)$ be the *worst-case number of key-comparisons* for mergesort algorithm to sort n elements.

(a) Consider the special case when n is a power of 2. That is, $n = 2^k$ for some integer k. We can describe $f(n)$ as follows. (This recursive definition of $f(n)$ is called a recurrence relation.)

$$f(n) = \begin{cases} 2f(n/2) + n - 1, & n > 1 \\ 0, & n = 1. \end{cases} \tag{1}$$

Prove by induction the solution has the following upper bound. (Note: All logorithms in this course are assumed to be in base 2, unless stated otherwise.)

$$f(n) \leq n \log n.$$

(b) Next, consider the general case when n is any integer. Then,

$$f(n) = \begin{cases} f(\lceil n/2 \rceil) + f(\lfloor n/2 \rfloor) + n - 1, & n > 1 \\ 0, & n = 1. \end{cases} \tag{2}$$

Prove by induction that:

$$f(n) \leq n \lceil \log n \rceil.$$

Hint: Observe that

$$\lceil \log \lfloor n/2 \rfloor \rceil \leq \lceil \log \lceil n/2 \rceil \rceil.$$

Use the fact that integer n falls between two consecutive powers of 2. That is, let

$$2^{k-1} < n \leq 2^k.$$

Then, $\lceil \log n \rceil = k$. And,

$$2^{k-2} < \lceil n/2 \rceil \leq 2^{k-1}.$$

So,

$$\lceil \log \lceil n/2 \rceil \rceil = k - 1 = \lceil \log n \rceil - 1.$$