Homework 6

Name: Aina Azman

Net ID: 457 464 051

Problem 1

***25.** Prove that if $h > -1$, then $1 + nh \leq (1 + h)^n$ for all nonnegative integers n. This is called **Bernoulli's inequality**.

- Predicate:
  - P(n) = $1 + nh \leq (1 + h)^n$ for $h > -1$.
- Base case:
  - P(0) = $1 + 0 \leq (1 + h)^0, 1 \leq 1$
  - The base case holds.
- Inductive step:
  - P(k) = $1 + kh \leq (1 + h)^k$.
  - Since $h > -1$, then $(1 + h) > 0$.
  - P(k+1): $(1 + h)^{k+1} = (1 + h)^k(1 + h) \geq (1 + h)(1 + kh) = 1 + kh + h + kh^2 = 1 + h(k + 1) + kh^2$.
  - Hence, $1 + h(k + 1) \leq 1 + h(k + 1) + kh^2$.
  - Further simplify, we get $0 \leq kh^2$ which is true given that k is a non-negative integers.
- Therefore, the Bernoulli's inequality holds.

Problem 2

***36.** Prove that 21 divides $4^{n+1} + 5^{2n-1}$ whenever n is a positive integer.

- Predicate:
  - P(n) = : $21|(4^{n+1} + 5^{2n-1})$ for n is a positive integers.
- Base case:
  - P(1)= $21|(4^2 + 5) = 1$
  - The base case holds.
- Inductive step:
  - P(k) = $(4^{k+1} + 5^{2k-1}) = 21k$
  - P(k+1): $(4^{k+2} + 5^{2(k+1)-1})$

$$
\begin{gathered}
(4^{k+2} + 5^{2(k+1)-1}) \\
(4^{k+2} + 5^{2k+1}) \\
(4(4^{k+1}) + 5^2(5^{2k-1})) \\
4(4^{k+1}) + 25(21k - 4^{k+1}) \\
4(4^{k+1}) + 25(21k) - 25(4^{k+1}) \\
21(25k) - 21(4^{k+1}) \\
21(25k - 4^{k+1})
\end{gathered}
$$

- Hence, the equation provided is indeed divisible by 21 given n is a positive integer.

Problem 3

70. Use mathematical induction to prove that $G(n) \leq 2n - 4$ for $n \geq 4$. [*Hint:* In the inductive step, have a new person call a particular person at the start and at the end.]

- Predicate
  - P(n) = $G(n) \leq 2n - 4$ for $n \geq 4$.

- Base case:
  - P(4) = $G(4) \leq 2(4) - 4 = 4.$
  - The base case holds.
- Inductive Step:
  - $G(k) \leq 2k - 4 = 2(k-2)$
  - $G(k+1) \leq 2(k+1) - 4$
  - A new person is added who calls a particular person at the start and at the end.
  - This new person can be the last caller in the sequence, calling a person at the k-th position at the start, and calling a person at the (k+1)-th position at the end.
  - Now, there are two parts:
    - 1st - group is up to the k-th position, denoted by G(k)
    - 2nd - where the new person is called, denoted by 1 (since it takes one more move to call this new person).

$$\text{Given that } G(k) \leq 2k - 4$$
$$G(k+1) = G(k) + 1 \leq (2k - 4) + 1 = 2k - 3$$
$$2k - 3 = 2(k+1) - 5$$
$$\text{Since k} \geq 4, \text{then } k + 1 \geq 5$$
$$\text{Therefore, } 2(k+1) - 5 \leq 2(k+1) - 4$$
$$\text{Thus, } G(k+1) \leq 2(k+1) - 4$$

- Thus, by the principle of mathematical induction, $G(n) \leq 2n - 4$ for $n \geq 4$.

Problem 4

14. Suppose you begin with a pile of n stones and split this pile into n piles of one stone each by successively splitting a pile of stones into two smaller piles. Each time you split a pile you multiply the number of stones in each of the two smaller piles you form, so that if these piles have r and s stones in them, respectively, you compute rs. Show that no matter how you split the piles, the sum of the products computed at each step equals $n(n-1)/2$.

- Predicate:
  - P(n) = n(n-1)/2
- Base case:
  - P(1) =1(1-1)/2 = 0
  - The base case holds.
- Inductive Hypothesis:
  - Assume that for some positive integer k, P(k) = k(k-1)/2
- Inductive step:
  - P(k) = k(k-1)/2
  - When a pile of (k+1) stones is split, we get two piles of stones with size r and s, where $r+s=k+1$.
  - The product of this step produces rs.
  - $P(k+1)=P(r)+P(s)+rs$.
  - By inductive hypothesis, $P(r)=r(r-1)/2$ and $P(s)=s(s-1)/2$.
  - Substituting the above expressions in P(k+1), we get $P(k+1)=r(r-1)/2+s(s-1)/2+rs$.
  - Expand the expression and we get $(r^2-r+s^2-s+2rs)/2=((r+s)(r+s-1))/2$
  - Since we also know that $r+s=k+1$, we get $((k+1)(k+1-1))/2=k(k+1)/2$.

- Hence, it is proven that $P(k+1) = k(k+1)/2$.

- Therefore, it is proven that no matter how the piles are split, the sum of the products computed at each steps will always equals to $n(n-1)/2$.

Problem 5

25. Suppose that $P(n)$ is a propositional function. Determine for which positive integers n the statement $P(n)$ must be true, and justify your answer, if

a) $P(1)$ is true; for all positive integers n, if $P(n)$ is true, then $P(n+2)$ is true.

b) $P(1)$ and $P(2)$ are true; for all positive integers n, if $P(n)$ and $P(n+1)$ are true, then $P(n+2)$ is true.

c) $P(1)$ is true; for all positive integers n, if $P(n)$ is true, then $P(2n)$ is true.

d) $P(1)$ is true; for all positive integers n, if $P(n)$ is true, then $P(n+1)$ is true.

a)

- Given that P(1) is true, then P(1 + 2) is true based on the statement.
- If P(3) is true, then P(3+2) is also true.
- If P(5) is true, then P(5+2) is also true.
- If P(7) is true, then P(7+2) is also true.
- Therefore, P(n) is true for n = 1, 3, 5, 7, 9, ...

b)

- Given that P(1) and P(2) are true, then P(1 + 2) is true based on the statement.
- If P(2) and P(3) is true, then P(2 + 2) is also true.
- If P(3) and P(4) is true, then P(3+2) is also true.

- Therefore, P(n) and P(n+1) are true for n = 1, 2, 3, 4, 5, ... which is any positive n integer.

c)

- Given that P(1) is true, then P(2 (1)) is true based on the statement.
- If P(2) is true, then P(2 (2)) is also true.
- If P(4) is true, then P(2 (4)) is also true.
- If P(8) is true, then P(2 (8)) is also true.
- If P(16) id true, then P(2 (16)) is also true.
- Therefore, P(n) is true for n = 1, 2, 4, 8, 16, ... which is any positive n integer that is the power of 2.

d)

- Given that P(1) is true, then P(1 + 1) is true based on the statement.
- If P(2) is true, then P(2+1) is also true.
- If P(3) is true, then P(3+1) is also true.
- If P(4) is true, then P(4+1) is also true.
- Therefore, P(n) is true for n = 1, 2, 3, 4, 5, ... which is any positive n integer.

Problem 6

6. **(10 points)** Devise a recursive algorithm for finding $x^n \pmod{m}$ whenever n, x and m are positive integers. Also prove that your algorithm is correct. You should use the fact that $x^n \pmod{m} = (x^{n-1} \pmod{m} \cdot x \pmod{m}) \pmod{m}$.

- Recursive algorithm to find $x^n (\text{mod m})$

```
function modExponentiation(int n,int x, int m){

if(n == 0) //base case
    return 1%m;


else //inductive step
    int temp = modExponentiation (n - 1, x, m) * (x % m);

    return temp % m;

}
```

- Prove by mathematical induction:
  - Base case:
    - When n = 0, the algorithm returns 1 mod m, which is 1 as any number mod m is itself.
  - Inductive step:
    - Assume that for some positive integer k, the algorithm correctly computes x^k mod m using recursive calls.
    - Consider $n = k + 1$:
      - $x^{k+1} \bmod m = (x^k \bmod m \times x \bmod m) \bmod m$
    - By the induction hypothesis, we assume that x^k mod m is computed correctly.
    - Therefore, the recursive call modExponentiation(k, x, m) returns x^k mod m correctly.

- Thus, the algorithm computes $(x^k \bmod m \times x \bmod m)$ mod m correctly.
- Hence, the algorithm correctly computes x^{k+1} mod m.
- Therefore, by mathematical induction, the algorithm correctly computes x^n mod m for all positive integers n.