CSci 243 Homework 8

Due: 10:00 am, Monday, November 2
Alexander Powell

1. (10 points) Suppose that a store offers gift certificates in denominations of \$25 and \$40. Determine the possible total amounts you can form using these gift certificates. Prove your answer using strong induction.

   Solution:

   Proof. Let's begin by looking at possible combinations of gift certificates to see if we can spot a pattern. This will be our inductive basis. We have:

   $25 \times 1 = 25$

   $40 \times 1 = 40$

   $25 \times 2 = 50$

   $40 + 25 = 65$

   $25 \times 3 = 75$

   $40 \times 2 = 80$

   $40 + 25 \times 2 = 90$

   $25 \times 4 = 100$

   $40 \times 2 + 25 = 105$

   $40 \times 3 = 120$

   $25 \times 5 = 125$

   $40 \times 2 + 25 \times 2 = 130$

   $25 \times 4 + 40 = 140$

   $25 + 40 \times 3 = 145$

   $25 \times 6 = 150$

   $25 \times 3 + 40 \times 2 = 155$

   $40 \times 4 = 160$

   Now, at this point we can make the claim that we can create $140 + 5n, n \geq 0$ using the gift certificates. Inductive Hypothesis: We can form any number that's a multiple of 5 and above 160 with only \$25 and \$40 gift certificates. Inductive Step: We already have a string of 5 consecutive multiples of 5 (140 through 160). If n is a multiple of 5 greater than 160 then the inductive hypothesis says that it is possible to create $n - 25$ with the gift certificates. Therefore, It is also possible to create n by just adding another \$25 gift certificate. Therefore, we have proven that we can form all amounts in the form $140 + 5n, n \geq 0$ as well as those shown in the base case. □

2. (8 points) Consider the dot game we saw in class. Again there are two rows of dots, with n_1 and n_2 dots respectively, and players can remove any number of dots during their turn, but only from one row. However, now, the player who removes the last dot loses.

   If $n_1 = n_2 > 1$, prove that there is a winning strategy for the second player. What happens if $n_1 \neq n_2$?

   Solution:

   Proof. We use induction on n (or equivalently n_1 and n_2).

   Base Case: $n = 2$. When $n_1 = n_2 = 2$ then there are 2 rows of 2 dots for a total of 4 dots.

   There are two cases:

   Case 1:

   If player 1 removes one dot from n_1 then player 2 should remove both dots from n_2 forcing player 1 to remove the last dot on n_1 and lose.

   Case 2:

   If player 2 removes both dots from n_1 then player 2 should only remove one dot from n_2 forcing player 1 to remove the last dot on n_2. Again, player 2 wins so the base step holds.

   Inductive Hypothesis: Assume true for any number of dots $j \leq k$, player 2 can choose a winning strategy.

   Inductive Step: Prove that for some games with rows of dots $k_1 = k_2 > 1$ and $k_1 = n_1 + 1$ there is a winning strategy for player 2. From here there are three cases:

   Case 1: If player 1 removes all dots except one from a row, then player 2 removes the entire other row, forcing player 1 to remove the last dot and lose.

   Case 2: If player 1 removes all dots from a row then player 2 should remove all but one dot from the other row, again forcing player 1 to remove the last dot and lose.

   Case 3: If player 1 leaves more than one dot on the row they play on, then player 2 should remove an equal number from the other row. This will result in a smaller game that will eventually come to a point where we can use the inductive hypothesis to prove player 2 will have a winning strategy.

   Therefore, we have proven that if $n_1 = n_2 > 1$, there is a winning strategy for the second player.

   Also, if $n_1 \neq n_2$ there is not any guaranteed strategy for a player 2 to win. Consider the game where there is a row of 3 dots and a row of 2 dots. Then for the case where player 1 removes one dots from the row of three dots, either way the second player will lose (depending of course on the intelligence of the first player). $\square$

3. (6 points) Give a recursive definition for each of the following sequences $\{a_n\}$ for $n = 1, 2, 3, \ldots$.

   (a) $a_n = 4n - 2$

   Solution: We can see the pattern by evaluating the first few terms of the sequence:

$$a_1 = 4(1) - 2 = 2$$

$$a_2 = 4(2) - 2 = 6$$

$$a_3 = 4(3) - 2 = 10$$

$$a_4 = 4(4) - 2 = 14$$

So, we can see the the sequence $\{a_n\}$ is the sequence of starting at 2, each successive element being 4 more than the one before it. Therefore, we can represent this sequence with the recursive definition:

$$a_1 = 2,\ a_{n+1} = a_n + 4$$

(b) $a_n = 1 + (-1)^n$

Solution: We can see the pattern by evaluating the first few terms of the sequence:

$$a_1 = 1 + (-1)^1 = 0$$
$$a_2 = 1 + (-1)^2 = 2$$
$$a_3 = 1 + (-1)^3 = 0$$
$$a_4 = 1 + (-1)^4 = 2$$

So, we can see that the sequence is made up of alternating $0s$ and $2s$. This can be represented with the recursive definition

$$a_1 = 0,\ a_{n+1} = a_n + 2(-1)^{n+1}$$

(c) $a_n = (\frac{1}{2})^n$

Solution: Again, by examining the pattern from the first few terms of the sequence we get:

$$a_1 = \left(\frac{1}{2}\right)^1 = \frac{1}{2}$$
$$a_2 = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$$
$$a_3 = \left(\frac{1}{2}\right)^3 = \frac{1}{8}$$
$$a_4 = \left(\frac{1}{2}\right)^4 = \frac{1}{16}$$

So, the pattern is each element of the sequence is one half of the one before it. This can be defined in the recursive definition:

$$a_1 = \frac{1}{2},\ a_{n+1} = \frac{a_n}{2}$$

4. For string $w = a_1 a_2 \cdots a_n$, the reversal of the string is defined as $w^R = a_n \cdots a_2 a_1$.

(a) (2 points) What is ε^R? What is $(10110)^R$?

Solution: $\varepsilon^R = \varepsilon$ and $(10110)^R = (01101)$

(b) (4 points) Give a recursive definition of the reversal of a string.

Solution: A recursive definition for the reversal of a string can be given with the basis of $S^R = S$ and the recursive step: if $w = ua$ for $u \in \Sigma^*$ and $a \in \Sigma$, then $w^R = au^R$.

(c) (6 points) Use structural induction to prove that $(w_1w_2)^R = w_2^R w_1^R$.

Solution: Let $P(w)$ be the property where $(w_1w_2)^R = w_2^R w_1^R$, when $w_1 \in \Sigma^*$. We must show that $\forall w \in \Sigma^*,\ P(w)$.

Base case: When $|w_2| = 0$, then $(w_1w_2)^R = w_1^R = w_2 w_1^R = w_2^R w_1^R$, so the base case holds.

Inductive Hypothesis: Assume $P(w)$ is true for any w.

Inductive Step: We must show that $P(wa)$ is true $\forall a \in \Sigma$. From our assumption, $(w_1w_2)^R = w_2^R w_1^R\ \forall w \in \Sigma^*$. So, $(w_1(w_2a))^R = ((w_1w_2)a)^R = a(w_1w_2)^R = aw_2^R w_1^R = (w_2a)^R w_1^R$. Therefore, $P(wa)$ holds. Thus, $(w_1w_2)^R = w_2^R w_1^R$.

5. (8 points) A palindrome is a string that reads the same forward and backward, i.e., $w = w^R$. Give a recursive algorithm in pseudocode that checks whether a given string w is a palindrome. What is the time complexity of your algorithm?

   Solution:

```
Algorithm IsPalindrome(string word)
{
    // We consider 1 element strings to be palindromes
    if (length(word) <= 1) { return true }
    first = word[0]
    last = word[length(word) - 1]
    if (first != last) { return false }
    else
    {
        return IsPalindrome(word.substring(1, length(word) - 1))
    }
}
```

   If we denote n to be the length of the input string then in the worst case the function would have to be recursively called $\frac{n}{2}$ times. So we can say the order is $O(\frac{n}{2})$, which is equivalent to taking out the constant and saying the complexity is $O(n)$.

6. (8 points) Give a recursive algorithm in pseudocode that finds the maximum number among n integers. What is the time complexity of your algorithm?

   Solution:

```
Algorithm GetMaximum(Array array, int n)
{
    if (n == 1) { return array[0] }
    else
    {
        max = GetMaximum(array, n-1)
        if (array[n-1] > max) { return array[n-1] }
        else { return max }
    }
}
```

   Since at the worst case this algorithm will have to be called as many times as the length of the given array, the time complexity can be described as $O(n)$.

7. (8 points) Assume $n = 4^k$ (i.e., $k = \log_4 n$) for some k. Solve the following recurrence relation by iteration.

$$f(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3f(\frac{n}{4}) + n & \text{if } n \geq 2 \end{cases}$$

Solution: By evaluating the first few iterations of the recurrence relation we get:

$$f(1) = 1$$

$$f(4) = 3 \times f(1) + 4 = 7$$

$$f(16) = 3 \times f(4) + 16 = 37$$

$$f(64) = 3 \times f(16) + 64 = 175$$

$$f(256) = 3 \times f(64) + 256 = 781$$

$$f(1024) = 3 \times f(256) + 1024 = 3367$$

From this, we can hypothesize the recurrence relation can be represented using the following formula:

$$4^{log_4(n \times 4)} - 3^{log_4(n \times 4)} = 4^{1+\log_4 n} - 3^{1+\log_4 n}$$

We need to prove this with induction:

Base Case: $f(1) = 1 = 4^{1+log_4 1} - 3^{1+log_4 1} = 4^1 - 3^1 = 1$, so the base case holds true.

Inductive Hypothesis: Assume for all k that $f(k) = 4^{1+\log_4 k} - 3^{1+\log_4 k}$.

Inductive Step: We must prove that for $4k$ the property holds true. We can show that $4^{1+log_4 4k} - 3^{1+log_4 4k} = 3 \times f(\frac{4k}{k}) + 4k$. Additionally,

$$f(4k) = 4f(\frac{4k}{k}) + 4k = 3(4^{1+\log_4 k} - 3^{1+\log_4 k}) + k = 3 \times 4^{1+\log_4 k} - 3^{2+\log_4 k} + 4k$$

$$= 3 \times 4^{1+\log_4 k} - 3^{1+\log_4 4k} + 4k = 4k + 4k - 3^{1+\log_4 4k}$$

$$= 12k + 4k - 3^{1+\log_4 4k} = 16k - 3^{1+\log_4 4k} = 4^{1+\log_4 4k} - 3^{1+\log_4 4k},$$

Therefore, we have proven that the recurrence relation can be solved with the above closed formula.