Ryan Lo (101117765)
COMP 2804 — Assignment 2

Due: Sunday October 17, 11:59 pm.

Assignment Policy:

- Your assignment must be submitted as a single .pdf file. Typesetting (using Latex, Word, Google docs, etc) is recommended but not required. Marks will be deducted for illegible or messy solutions. This includes but is not limited to excessive scribbling, shadows, blurry photos, messy handwriting, etc.
- **No late assignments will be accepted.**
- You are encouraged to collaborate on assignments, but at the level of discussion only. When writing your solutions, you must do so in your own words.
- Past experience has shown conclusively that those who do not put adequate effort into the assignments do not learn the material and have a probability near 1 of doing poorly on the exams (which is where most of the marks are).
- When writing your solutions, you must follow the guidelines below.
  - You must justify your answers.
  - The answers should be concise, clear and neat.
  - When presenting proofs, every step should be justified.

Question 1:

- Write your name and student number.

Question 2: You have cultivated a unique strain of tomato that is delicious in your backyard, so you decide to grow more. You have 19 plants to start (year 0), and you let them germinate naturally, which means each plant turns into 3 plants the following year. Starting in year 1 you will harvest 10 plants per year of operation. So year 1 is 10 plants, year 2, 20 plants, etc, to sell at the farmer's market. A local squirrel population finds your tomato garden in the first year and eat 21 plants a year thereafter. That means we can express the growth of our tomato garden year by year with the following recursive function.

$$f(0) = 19,$$
$$f(n) = 3 \cdot f(n-1) - 10n - 21, n \geq 1.$$

Prove that the closed form of this recursion is $f(n) = 3^n + 5n + 18, \forall n \geq 0$.

Base Case:
$f(0) = 3^0 + 5(0) + 18 = 19$
Inductive Step: Let $n \geq 1$
Assume $n - 1$ is true
$f(n-1) = 3^{n-1} + 5(n-1) + 18$
$f(n) = 3 * f(n-1) - 10n - 21$
$= 3 * (3^{n-1} + 5(n-1) + 18) - 10n - 21$
$= 3^{n-1+1} + 15(n-1) + 54 - 10n - 21$
$= 3^n + 15n - 15 + 54 - 10n - 21$
$= 3^n + 5n + 18$

Question 3: Consider the following recursive algorithm and let n be a power of 3. The subroutine PROCESS($List[i]$) takes a single character from position i in the list $List$ and does some operation to it. Determine the number of times PROCESS($List[i]$) is called as a function of n. Be sure to justify (prove) your answer.

```
Algorithm THIRDS(List, n):

    if n = 1:
        return;
    for i in range(n):
        PROCESS(List[i]);
    THIRDS(List, n/3)
```

Everytime THIRDS(List,n) is called, Process happens n times.
Let T(n) be the number of times process is called.
Base Case:
$T(1) = 0$, it just returns, process isn't called.
$T(n) = T(n/3) + n$
$T(n/3) = T(n/9) + n/3$
$T(n) = T(n/9) + n/3 + n$
$T(n/9) = T(n/27) + n/9$
$T(n) = T(n/27) + n/9 + n/3 + n$
$T(n) = T(n/n) + T(3) + T(9) + ... + n/9 + n/3 + n$
$T(n) = T(1) + T(3) + T(9) + ... + n/9 + n/3 + n$
$T(n) = 0 + 3 + 9 + ... + n/9 + n/3 + n$
After k steps, $n = 3^k$
$k = \log_3 n$
$T(n) = 0 + 3 + 9 + ... + 3^{k-2}/9 + 3^{k-1}/3 + 3^k$
$T(n) = \sum_{i=1}^{k} 3^i$

Closed form of exponential sum:

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

Changing the expression we have above into closed form:

$$T(n) = \frac{3(3^k - 1)}{3 - 1} = 3/2 * (n - 1)$$

$$T(n) = 3/2 * (n - 1)$$

Question 4:

(a) Prove that there cannot be a 00-free bitstring of length n with less than $\lfloor \frac{n}{2} \rfloor$ 1's.

There are n boxes, for a bitstring to be of 00-free, two 0's cannot be next to each other. That means there can only be $n/2$ of 0's in n boxes. To minimize the number of 1's, we must maximize the number of 0's. At most there could be $\lceil \frac{n}{2} \rceil$ 0's if the n is odd. If we fill at most $\lceil \frac{n}{2} \rceil$ 0's then the rest must be filled with 1's which is $\lfloor \frac{n}{2} \rfloor$ 1's. So therefore, the amount of 1's in a 00-free bitstring cannot be less than $\lfloor \frac{n}{2} \rfloor$.

(b) Explain why $\sum_{i=\lfloor \frac{n}{2} \rfloor}^{n} \binom{n}{i}$ is an upper bound on the number of 00-free bitstrings of length n.

This summation is an upper bound on the number of 00-free bitstrings of length n because the number of 00-free bitstrings if length n is f_{n+2}.

When n=1, $\sum_{i=\lfloor \frac{1}{2} \rfloor}^{1} \binom{1}{i} = \binom{1}{0} + \binom{1}{1} = 2$

When n=2 $\sum_{i=\lfloor \frac{2}{2} \rfloor}^{2} \binom{2}{i} = \binom{2}{1} + \binom{2}{2} = 3$

When n=3 $\sum_{i=\lfloor \frac{3}{2} \rfloor}^{3} \binom{3}{i} = \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7$

When n=4 $\sum_{i=\lfloor \frac{4}{2} \rfloor}^{4} \binom{4}{i} = \binom{4}{2} + \binom{4}{3} + \binom{4}{4} = 11$

$\sum_{i=\lfloor \frac{3}{2} \rfloor}^{3} \binom{3}{i} = 7$ but $f_{3+2} = 5$

$\sum_{i=\lfloor \frac{4}{2} \rfloor}^{4} \binom{4}{i} = 11$ but $f_{4+2} = 8$

As n increases, both $\sum_{i=\lfloor \frac{n}{2} \rfloor}^{n} \binom{n}{i}$ and f_{n+2} will increase but $\sum_{i=\lfloor \frac{n}{2} \rfloor}^{n} \binom{n}{i}$ will never surpass f_{n+2}. Therefore, the number of bitstrings is always going to be equal to or less than $\sum_{i=\lfloor \frac{n}{2} \rfloor}^{n} \binom{n}{i}$, so it can be considered an upper bound.

(c) Assume n is even. How many 00-free bitstring of length n have exactly $\frac{n}{2}$ 1's?.

If n is even, and you have exactly $\frac{n}{2}$ 1's then you must have exactly $\frac{n}{2}$ 0's. And every 0 must be separated by a 1. Either the bitstrings start with 1 or starts with 0. Therefore, there are 2 ways for this to happen.

Task 1: Write down the $n/2$ 1's, since n is even there is only one way to do this.

Task 2: Place the $n/2$ 0's between the 1's such that no two 0's are next to each other. There are $n/2-1$ places in between the 1's and 2 more places at the beginning and at the end of the 00-free bitstring. For a total of $n/2+1$ possible locations.

Therefore the number of 00-free bitstrings of length n that have exactly $n/2$ 1's is $\binom{n/2+1}{n/2}$

(d) Given that n is even, how many 00-free bitstring of length n where n is even have exactly $(\frac{n}{2}+1)$ 1's? $(\frac{n}{2}+2)$ 1's?

Consider an example where n is 8. Having exactly $(\frac{n}{2}+1)$ 1's means that there are 5 1's and 3 0's.

Task 1: Write down the 5 1's, there is 1 way to do this.

Task 2: Place the 3 0's between the 1's such that no two 0's are next to one another. There are 6 possible locations and 3 0's to place so $\binom{6}{3}$

The number of 00-free bitstrings of length n where n is even and has exactly $(\frac{n}{2}+1)$ 1's is $\binom{\frac{n}{2}+2}{\frac{n}{2}-1}$

Consider an example where n is 8. Having exactly $(\frac{n}{2}+2)$ 1's means that there are 6 1's and 2 0's.

Task 1: Write down the 6 1's, there is 1 way to do this

Task 2: Place the 2 0's between the 1's such that no two 0's are bext to one another. There are 7 possible locations and 2 0's to place so $\binom{7}{2}$

The number of 00-free bitstrings of length n where n is even and has exactly $(\frac{n}{2}+2)$ 1's is $\binom{\frac{n}{2}+3}{\frac{n}{2}-2}$ where $n \geq 4$

(e) Generalizing on your answer from (d), explain why the following is true for an even value n:

$$\sum_{i=0}^{\frac{n}{2}} \binom{i+\frac{n}{2}+1}{2i+1} = f_{n+2}$$

where f_{n+2} is the $(n+2)th$ Fibonnacci number.

Where n = 2:

$$\sum_{i=0}^{\frac{2}{2}} \binom{i+\frac{2}{2}+1}{2i+1} = f_{n+2}$$

$$\sum_{i=0}^{1} \binom{i+1+1}{2i+1} = f_{n+2}$$

$$\sum_{i=0}^{1} \binom{i+2}{2i+1} = f_{n+2}$$

$$\binom{0+2}{2(0)+1} + \binom{1+2}{2(1)+1} = f_{n+2}$$

$$\binom{2}{1} + \binom{3}{3} = f_{2+2}$$

$$2 + 1 = f_4$$

$$3 = f_4$$

Question 5: Consider a string of characters of length n where each character is chosen from the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. We will call these *decimal strings*.

(a) We call a decimal string 00-*free* if it does not contain two consecutive 0's. Let D_n be the number of 00-free decimal strings of length n. Determine D_1 and D_2, then express D_n in terms of D_{n-1} and D_{n-2} for $n \geq 3$.

$D_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$D_2 = \{01, 02, 03, 04, 05, 06, 07, 08, 09,$
$10, 11, 12, 13, 14, 15, 16, 17, 18, 19,$
$20, 21, 22, 23, 24, 25, 26, 27, 28, 29,$
$30, 31, 32, 33, 34, 35, 36, 37, 38, 39,$
$40, 41, 42, 43, 44, 45, 46, 47, 48, 49,$
$50, 51, 52, 53, 54, 55, 56, 57, 58, 59,$
$60, 61, 62, 63, 64, 65, 66, 67, 68, 69,$
$70, 71, 72, 73, 74, 75, 76, 77, 78, 79,$
$80, 81, 82, 83, 84, 85, 86, 87, 88, 89,$
$90, 91, 92, 93, 94, 95, 96, 97, 98, 99\}$

$D_1 = 10^1$, $D_2 = 10^2 - 1 = 99$

A decimal string can be formed from one of two ways:

1: X, All 00-free decimal strings of length n-1

Where X ∈ {1,2,3,...,9}

2: X, Y, All 00-free decimal strings of length n-2

Where X $\in$ {1,2,3,...,9} and Y $\in$ {0,1,2,3,...,9}

Therefore $D_n = 9 * D_{n-1} + 9 * 10 * D_{n-2}$

$D_n = 9 * D_{n-1} + 90 * D_{n-2}$

(b) Find an expression for the number of 00-free decimal strings of length n with precisely i 0's. As in Question 4 there can be at most $n - \lfloor \frac{n}{2} \rfloor$ 0's in the string, so you may assume $i \leq n - \lfloor \frac{n}{2} \rfloor$. Hint: Look at your answer for Question 4.

At most there can be $n - \lfloor \frac{n}{2} \rfloor$ 0's

Let's assume that the 0's are going to be in the even slots, now we can choose i of $n - \lfloor \frac{n}{2} \rfloor$ places to put these zeros in. The remaining spots would be filled with {1,2,3,...,9}. And there would be $n - i$ spots left from the subtracted amount of 0's already places.

$$\binom{n - \lfloor \frac{n}{2} \rfloor}{i} * 9^{n-i}$$

Since there are two possible ways for this to happen (0's in even & 0's in odd),

$$\binom{n - \lfloor \frac{n}{2} \rfloor}{i} * 9^{n-i} * 2$$