COMP 2804 — Solutions Assignment 2

Question 1: On the first page of your assignment, write your name and student number.

Solution:

- Name: James Bond
- Student number: 007

Question 2: The function $f : \mathbb{N} \rightarrow \mathbb{N}$ is defined by

$$\begin{aligned} f(0) &= 0, \\ f(1) &= 0, \\ f(n) &= f(n-2) + 2^{n-1} \quad \text{if } n \geq 2. \end{aligned}$$

- Prove that for every even integer $n \geq 0$,
$$f(n) = \frac{2^{n+1} - 2}{3}.$$
- Prove that for every odd integer $n \geq 1$,
$$f(n) = \frac{2^{n+1} - 4}{3}.$$

Solution: For the first part, we do induction over all even integers $n \geq 0$. The base case is when $n = 0$. The LHS is $f(0)$, which is 0 by the definition of the function f. The RHS is

$$\frac{2^{0+1} - 2}{3} = \frac{2 - 2}{3} = 0.$$

Since LHS equals RHS, we are done with the base case.

For the induction step, let $n \geq 2$ be an even integer and assume that the claim is true for $n - 2$. Note that $n - 2$ is even as well. Thus, we assume that

$$f(n-2) = \frac{2^{n-1} - 2}{3}.$$

We have

$$\begin{aligned} f(n) &= f(n-2) + 2^{n-1} \\ &= \frac{2^{n-1} - 2}{3} + 2^{n-1} \\ &= \frac{2^{n-1} - 2 + 3 \cdot 2^{n-1}}{3} \\ &= \frac{4 \cdot 2^{n-1} - 2}{3} \\ &= \frac{2^{n+1} - 2}{3}. \end{aligned}$$

This proves the induction step.

For the second part, we do induction over all odd integers $n \geq 1$. The base case is when $n = 1$. The LHS is $f(1)$, which is 0 by the definition of the function f. The RHS is

$$\frac{2^{1+1} - 4}{3} = \frac{4 - 4}{3} = 0.$$

Since LHS equals RHS, we are done with the base case.

For the induction step, let $n \geq 3$ be an odd integer and assume that the claim is true for $n - 2$. Note that $n - 2$ is odd as well. Thus, we assume that

$$f(n-2) = \frac{2^{n-1} - 4}{3}.$$

We have

$$\begin{aligned} f(n) &= f(n-2) + 2^{n-1} \\ &= \frac{2^{n-1} - 4}{3} + 2^{n-1} \\ &= \frac{2^{n-1} - 4 + 3 \cdot 2^{n-1}}{3} \\ &= \frac{4 \cdot 2^{n-1} - 4}{3} \\ &= \frac{2^{n+1} - 4}{3}. \end{aligned}$$

This proves the induction step.

Question 3: The function $f : \mathbb{N}^2 \to \mathbb{N}$ is defined by

$$\begin{aligned} f(0, n) &= 2n \text{ if } n \geq 0, && (1) \\ f(m, 0) &= 0 \text{ if } m \geq 1, && (2) \\ f(m, 1) &= 2 \text{ if } m \geq 1, && (3) \\ f(m, n) &= f(m-1, f(m, n-1)) \text{ if } m \geq 1 \text{ and } n \geq 2. && (4) \end{aligned}$$

- Determine $f(2, 2)$.
- Determine $f(1, n)$ for $n \geq 1$.
- Determine $f(3, 3)$.

Solution: We start with $f(2, 2)$. From (4), we get

$$f(2, 2) = f(1, f(2, 1)).$$

From (3), we get $f(2,1) = 2$. Thus, we get

$$f(2,2) = f(1, f(2,1)) = f(1,2).$$

From (4), we get

$$f(2,2) = f(1,2) = f(0, f(1,1)).$$

From (3), we get $f(1,1) = 2$. Thus, we get

$$f(2,2) = f(0, f(1,1)) = f(0,2).$$

From (1), we get $f(0,2) = 4$. We conclude that

$$f(2,2) = f(0,2) = 4.$$

For the second part, if you determine $f(1,n)$ for some small values of n, you will see a pattern: It looks like

$$f(1,n) = 2^n \text{ for all } n \geq 1.$$

We prove by induction that this is correct: The base case is when $n = 1$. From (3), the LHS is equal to $f(1,1) = 2$. The RHS is $2^n = 2^1 = 2$. Since LHS equals RHS, we are done with the base case.

For the induction step, let $n \geq 2$ be an integer and assume that the claim is true for $n - 1$. Thus, we assume that

$$f(1, n-1) = 2^{n-1}.$$

From (4), (1) and this assumption, we get

$$f(1,n) = f(0, f(1, n-1)) = 2 \cdot f(1, n-1) = 2 \cdot 2^{n-1} = 2^n.$$

This proves the induction step.

Finally, we determine $f(3,3)$: From (4), we get

$$f(3,3) = f(2, f(3,2)).$$

Before we continue, we determine $f(3,2)$: From (4), (3), and the first part of this question, we get

$$f(3,2) = f(2, f(3,1)) = f(2,2) = 4.$$

Now we go back to $f(3,3)$:

$$f(3,3) = f(2, f(3,2)) = f(2,4).$$

From (4), we get

$$f(3,3) = f(2,4) = f(1, f(2,3)).$$

Before we continue, we determine $f(2,3)$: From (4), the first part of the question, and the second part of the question, we get

$$f(2,3) = f(1, f(2,2)) = f(1,4) = 2^4 = 16.$$

Now we go back to $f(3,3)$:

$$f(3,3) = f(1, f(2,3)) = f(1,16).$$

From the second part of the question, we get

$$f(3,3) = f(1,16) = 2^{16}.$$

Question 4: A *block* in a bitstring s is a maximal sequence of consecutive 1's. For example, the bitstring $s = 11011101000$ contains the three blocks

$$\underbrace{11}\,0\,\underbrace{111}\,0\,\underbrace{1}\,000.$$

A bitstring is called *awesome*, if each of its blocks has an even length. Thus, the bitstring above is not awesome, whereas both bitstrings 00011011110 and 0000000 are awesome.

For any integer $n \geq 1$, let A_n denote the number of awesome bitstrings of length n.

- Determine A_1, A_2, A_3, and A_4.
- Determine the value of A_n, i.e., express A_n in terms of numbers that we have seen in class.

  Hint: Derive a recurrence relation.

Solution: For $n = 1$, there are 2 bitstrings of length 1. The bitstring 0 is awesome, whereas the bitstring 1 is not awesome. Thus,

$$A_1 = 1.$$

For $n = 2$, there are 4 bitstrings of length 2. The bitstrings 00 and 11 are awesome, whereas the bitstrings 01 and 10 are not awesome. Thus,

$$A_2 = 2.$$

For $n = 3$, there are 8 bitstrings of length 3. The bitstrings 000, 011, and 110 are awesome. Each other bitstring either has exactly one 1 or is equal to 101; neither of them is awesome. Thus,

$$A_3 = 3.$$

For $n = 4$, there are 16 bitstrings of length 4. The bitstrings 0000, 0011, 0110, 1100, and 1111 are awesome. None of the other bitstrings is awesome. Thus,

$$A_4 = 5.$$

Let $n \geq 3$ be an integer. We are going to express A_n in terms of A_{n-1} and A_{n-2}:

- Make a table consisting of all awesome bitstrings of length n, one bitstring per row. The number of rows in this table is equal to A_n.
- Re-arrange the rows such that the top part contains all awesome bitstrings that start with 0, and the bottom part contains all awesome bitstrings that start with 1.
- Consider the top part. If we remove the first bit (which is 0), then we see all awesome bitstrings of length $n-1$. Thus, the top part consists of A_{n-1} many bitstrings.
- Consider the bottom part. Each row starts with 1. The second bit must be 1 as well (otherwise, the first bit would form a block of length one, which is not even). If we remove the first two bits (which are 11), then we see all awesome bitstrings of length $n-2$. Thus, the top part consists of A_{n-2} many bitstrings.
- We conclude that for $n \geq 3$,
$$A_n = A_{n-1} + A_{n-2}.$$

Hey! This is the Fibonacci recurrence! Look at the following table:

f_0	f_1	f_2	f_3	f_4	f_5	f_6	...
0	1	1	2	3	5	8	...
		A_1	A_2	A_3	A_4	A_5	...

From this table, you see that
$$A_n = f_{n+1} \text{ for all } n \geq 1.$$

Question 5: The Fibonacci numbers are defined as follows: $f_0 = 0$, $f_1 = 1$, and $f_n = f_{n-1} + f_{n-2}$ for $n \geq 2$.

In class, we have seen that for any $m \geq 1$, the number of 00-free bitstrings of length m is equal to f_{m+2}.

- Let $n \geq 2$ be an integer. What is the number of 00-free bitstrings of length $2n-1$ for which the bit in the middle position is equal to 1?
- Let $n \geq 3$ be an integer. What is the number of 00-free bitstrings of length $2n-1$ for which the bit in the middle position is equal to 0?
- Use the previous results to prove that for any integer $n \geq 3$,
$$f_{2n+1} = f_n^2 + f_{n+1}^2.$$

Solution: Let $n \geq 2$ be an integer. Any 00-free bitstring of length $2n-1$ that has 1 in the middle has the following form:

- The first $n-1$ bits form a 00-free bitstring.

- The bit in the middle is 1.
- The last $n-1$ bits form a 00-free bitstring.

There are f_{n+1} many choices for the first $n-1$ bits, and there are f_{n+1} many choices for the last $n-1$ bits. By the Product rule, the number of 00-free bitstrings of length $2n-1$ that have 1 in the middle is equal to

$$f_{n+1} \cdot f_{n+1} = f_{n+1}^2.$$

Let $n \geq 3$ be an integer. Any 00-free bitstring of length $2n-1$ that has 0 in the middle has the following form:

- The first $n-2$ bits form a 00-free bitstring.
- The three bits in the middle are 101.
- The last $n-2$ bits form a 00-free bitstring.

There are f_n many choices for the first $n-2$ bits, and there are f_n many choices for the last $n-2$ bits. By the Product rule, the number of 00-free bitstrings of length $2n-1$ that have 0 in the middle is equal to

$$f_n \cdot f_n = f_n^2.$$

For the last part of this question, we know that the total number of 00-free bitstrings of length $2n-1$ is equal to f_{2n+1}. Out of these, there are f_{n+1}^2 that have 1 in the middle, and there are f_n^2 that have 0 in the middle. We conclude that

$$f_{2n+1} = f_n^2 + f_{n+1}^2.$$

Question 6: Elisa Kazan[1] loves to drink cider. During the weekend, Elisa goes to the pub and runs the following recursive algorithm, which takes as input an integer $n \geq 0$:

```
Algorithm ElisaGoesToThePub(n):

    if n = 0
    then order Fibonachos
    else if n is even
         then ElisaGoesToThePub(n/2);
               drink n^2/2 bottles of cider;
               ElisaGoesToThePub(n/2)
         else for i = 1 to 4
              do ElisaGoesToThePub((n − 1)/2);
                 drink (n − 1)/2 bottles of cider
              endfor;
              drink 1 bottle of cider
         endif
    endif
```

[1] President of the Carleton Computer Science Society

For $n \geq 0$, let $C(n)$ be the number of bottles of cider that Elisa drinks when running algorithm ElisaGoesToThePub(n). Determine the value of $C(n)$.

Solution: Since the algorithm is recursive, we are going to set up a recurrence for $C(n)$:

- If $n = 0$, Elisa orders Fibonachos. Since these are not a type of cider, we have

$$C(0) = 0. \tag{5}$$

- Assume that $n \geq 2$ is even. When we run ElisaGoesToThePub(n), Elisa does the following:
  - During the first recursive call, she drinks $C(n/2)$ bottles of cider.
  - Next, she drinks $n^2/2$ bottles of cider.
  - During the second recursive call, she drinks $C(n/2)$ bottles of cider.

  We conclude that

$$C(n) = 2 \cdot C(n/2) + n^2/2. \tag{6}$$

- Assume that $n \geq 1$ is odd. When we run ElisaGoesToThePub(n), Elisa does the following:
  - During each of the 4 iterations of the for-loop,
    * there is one recursive call, during which she drinks $C((n-1)/2)$ bottles of cider,
    * she drinks $(n-1)/2$ bottles of cider.
  - At the end, she drinks 1 bottle of cider.

  We conclude that

$$C(n) = 4\,(C((n-1)/2) + (n-1)/2) + 1,$$

  which we rewrite as

$$C(n) = 4 \cdot C((n-1)/2) + 2n - 1. \tag{7}$$

The complete recurrence is given by (5), (6), and (7).

Our task is to solve this recurrence. If you determine $C(n)$ for some small values of n, you will see a pattern: It looks like

$$C(n) = n^2 \text{ for all } n \geq 0.$$

We prove by induction that this is correct.

The base case is when $n = 0$. The LHS is $C(0)$, which is 0 by (5). The RHS is $0^2 = 0$. Since LHS equals RHS, we are done with the base case.

For the induction step, let $n \geq 1$ be an integer and assume that the claim is true for all values less than n. Thus, we assume that

$$C(m) = m^2 \text{ for all } m \text{ with } 0 \leq m < n.$$

First we assume that n is even. We know from (6) that

$$C(n) = 2 \cdot C(n/2) + n^2/2.$$

Since $0 \leq n/2 < n$, we can apply the induction hypothesis with $m = n/2$, and we get

$$\begin{aligned} C(n) &= 2 \cdot C(n/2) + n^2/2 \\ &= 2 \cdot (n/2)^2 + n^2/2 \\ &= 2 \cdot \left(n^2/4\right) + n^2/2 \\ &= n^2. \end{aligned}$$

Now assume that n is odd. We know from (7) that

$$C(n) = 4 \cdot C((n-1)/2) + 2n - 1.$$

Since $0 \leq (n-1)/2 < n$, we can apply the induction hypothesis with $m = (n-1)/2$, and we get

$$\begin{aligned} C(n) &= 4 \cdot C((n-1)/2) + 2n - 1 \\ &= 4\left(\frac{n-1}{2}\right)^2 + 2n - 1 \\ &= (n-1)^2 + 2n - 1 \\ &= \left(n^2 - 2n + 1\right) + 2n - 1 \\ &= n^2. \end{aligned}$$

This proves the induction step.

Question 7: In the fall term of 2015, Nick[2] took COMP 2804. Nick was always sitting in the back of the classroom and spent most of his time eating bananas. Nick uses the following banana-buying-scheme:

- At the start of week 0, there are 2 bananas in Nick's fridge.
- For any integer $n \geq 0$, Nick does the following during week n:
  - At the start of week n, Nick determines the number of bananas in his fridge and stores this number in a variable x.
  - Nick goes to Jim's Banana Empire, buys x bananas, and puts them in his fridge.

[2] your friendly TA

– Nick takes $n + 1$ bananas out of his fridge and eats them during week n.

For any integer $n \geq 0$, let $B(n)$ be the number of bananas in Nick's fridge at the start of week n. Determine the value of $B(n)$.

Solution: Since the banana-buying-scheme is recursive, we are going to set up a recurrence for $B(n)$:

- We are given that
$$B(0) = 2. \tag{8}$$
- Assume that $n \geq 1$. We are going to express $B(n)$ in terms of $B(n-1)$. Note that $B(n)$ is equal to the number of bananas in Nick's fridge at the start of week n. This is the same as the number of bananas in Nick's fridge at the end of week $n-1$. Let us see what happens during week $n-1$:
  - At the start of week $n-1$, there are $B(n-1)$ bananas in Nick's fridge.
  - At the start of week $n-1$, the variable x has value $B(n-1)$.
  - Nick buys $x = B(n-1)$ bananas and puts them in his fridge. At this moment, the fridge contains $2 \cdot B(n-1)$ bananas.
  - During week $n-1$, Nick eats n bananas. Thus, at the end of week $n-1$, there are $2 \cdot B(n-1) - n$ bananas in the fridge.

  We conclude that
$$B(n) = 2 \cdot B(n-1) - n. \tag{9}$$

The complete recurrence is given by (8) and (9).

Our task is to solve this recurrence. If you determine $B(n)$ for some small values of n, you will see a pattern: It looks like

$$B(n) = n + 2 \text{ for all } n \geq 0.$$

We prove by induction that this is correct.

The base case is when $n = 0$. The LHS is $B(0)$, which is 2 by (8). The RHS is $0 + 2 = 2$. Since LHS equals RHS, we are done with the base case.

For the induction step, let $n \geq 1$ be an integer and assume that the claim is true for $n - 1$. Thus, we assume that

$$B(n-1) = n + 1.$$

From (9) and this assumption, we get

$$\begin{aligned} B(n) &= 2 \cdot B(n-1) - n \\ &= 2(n+1) - n \\ &= n + 2. \end{aligned}$$

This proves the induction step.