# COMP 2804 — Solutions Assignment 2

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

**Solution:**

- Name: Sidney Crosby
- Student number: 87

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

$$\begin{array}{lll} f(0) & = & -18, \\ f(n) & = & 9(n-2)(n-3) + f(n-1) \quad \text{if } n \geq 1. \end{array}$$

Prove that

$$f(n) = 3(n-1)(n-2)(n-3)$$

for all $n \geq 0$.

**Solution:** The proof is by induction on $n$. The base case is when $n = 0$. Since $f(0) = -18$ and

$$3(n-1)(n-2)(n-3) = 3(-1)(-2)(-3) = -18,$$

the base case holds.

Let $n \geq 1$ and assume that the claim is true for $n - 1$. Thus, the induction hypothesis is that

$$f(n-1) = 3(n-2)(n-3)(n-4).$$

We have to show that

$$f(n) = 3(n-1)(n-2)(n-3).$$

Using the recurrence, the induction hypothesis, and some basic algebra, we get

$$\begin{array}{lll} f(n) & = & 9(n-2)(n-3) + f(n-1) \\ & = & 9(n-2)(n-3) + 3(n-2)(n-3)(n-4) \\ & = & 3(n-2)(n-3)\,(3 + (n-4)) \\ & = & 3(n-2)(n-3)(n-1) \\ & = & 3(n-1)(n-2)(n-3). \end{array}$$

**Question 3:** The functions $f : \mathbb{N} \rightarrow \mathbb{N}$ and $g : \mathbb{N}^2 \rightarrow \mathbb{N}$ are recursively defined as follows:

$$\begin{array}{llll} f(0) & = & 1, & \\ f(1) & = & 2, & \\ f(n) & = & g(f(n-2), f(n-1)) & \text{if } n \geq 2, \\ g(m, 0) & = & 2m & \text{if } m \geq 0, \\ g(m, n) & = & g(m, n-1) + 1 & \text{if } m \geq 0 \text{ and } n \geq 1. \end{array}$$

Solve these recurrences for $f$, i.e., express $f(n)$ in terms of $n$. Justify your answer.

**Solution:** If you stare long enough at the recurrence for $g$, it makes sense that

$$g(m,n) = 2m + n$$

for all $m \geq 0$ and $n \geq 0$. We prove by induction on $n$ that this is indeed the case.

The base case is when $n = 0$. Since $g(m,0) = 2m$ and $2m+n = 2m$, the base case holds.

Let $n \geq 1$ and assume that

$$g(m,n-1) = 2m + n - 1.$$

Then

$$g(m,n) = g(m,n-1) + 1 = (2m + n - 1) + 1 = 2m + n.$$

Now that we have solved the recurrence for $g$, we can rewrite the recurrence for $f$:

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

After determining some function values for small values of $n$, it looks like

$$f(n) = 2^n$$

for all $n \geq 0$. We prove by induction on $n$ that this is indeed the case. The base cases are $n = 0$ and $n = 1$:

- If $n = 0$: We have $f(0) = 1$ and $2^n = 2^0 = 1$. Thus, the claim is true for $n = 0$.
- If $n = 1$: We have $f(1) = 2$ and $2^n = 2^1 = 2$. Thus, the claim is true for $n = 1$.

Let $n \geq 2$ and assume that the claim is true for $n-1$ and $n-2$. Thus, we assume that

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

and

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

Then we get

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

**Question 4:** The set $S$ of binary strings is recursively defined in the following way:

- The string 00 is an element of the set $S$.
- The string 01 is an element of the set $S$.

- The string 10 is an element of the set $S$.
- If the string $s$ is an element of the set $S$, then the string $0s$ (i.e., the string obtained by adding the bit 0 at the front of $s$) is also an element of the set $S$.
- If the string $s$ is an element of the set $S$, then the string $10s$ (i.e., the string obtained by adding the bits 10 at the front of $s$) is also an element of the set $S$.

Let $s$ be an arbitrary string in the set $S$. Prove that $s$ does not contain the substring 11.

**Solution:** The strings in $S$ are defined in a recursive way: There are three base strings that are in $S$. These are 00, 01, and 10. Obviously, none of these three strings contains 11.

Then there is a recursive rule, telling us how to obtain more strings in $S$: You take a string $s$ for which you already know that it belongs to $S$. Then you produce two new strings, namely $0s$ and $10s$, which are also in $S$.

In the induction step, we take a string $s$ in $S$ and assume that it does not contain 11. Now we argue that the two new strings also do not contain 11:

- For the new string $0s$: Since $s$ does not contain 11, it is obvious that $0s$ does not contain 11.
- For the new string $10s$: Since $s$ does not contain 11, it is obvious that $10s$ does not contain 11.

**Question 5:** Let $n \geq 1$ be an integer and consider the set $S_n = \{1, 2, \ldots, n\}$. A *non-neighbor subset* of $S_n$ is any subset $T$ of $S$ having the following property: If $k$ is any element of $T$, then $k+1$ is not an element of $T$. (Observe that the empty set is a non-neighbor subset of $S_n$.)

For example, if $n = 3$, then $\{1, 3\}$ is a non-neighbor subset, whereas $\{2, 3\}$ is not a non-neighbor subset.

Let $N_n$ denote the number of non-neighbor subsets of the set $S_n$.

- Determine $N_1$, $N_2$, and $N_3$.
- Determine the value of $N_n$, i.e., express $N_n$ in terms of numbers that we have seen in class. Justify your answer. *Hint:* Derive a recurrence.

**First Solution:** In class, we have seen the following:

- The number of bitstrings of length $n$ that do not contain 00 is equal to $f_{n+2}$, the $(n+2)$-nd Fibonacci number.
- By changing the roles of 0 and 1, we see that the number of bitstrings of length $n$ that do not contain 11 is also equal to $f_{n+2}$.

- Each subset $T$ of $S_n$ can be encoded as a binary string of length $n$: The $i$-th bit of the string is 1 if $i \in T$, and it is 0 if $i \notin T$.

- If $T$ is a non-neighbor subset of $S_n$, then the corresponding bitstring does not contain 11. Conversely, any bitstring of length $n$ that does not contain 11 corresponds to a unique non-neighbor subset of $S_n$.

- We conclude that $N_n$ is equal to the number of bitstrings of length $n$ that do not contain 11. Thus, $N_n = f_{n+2}$. In particular, $N_1 = f_3 = 2$, $N_2 = f_4 = 3$, and $N_3 = f_5 = 5$.

**Second Solution:**

- To determine $N_1$, we list all non-neighbor subsets of $S_1 = \{1\}$: These are $\emptyset$ and $\{1\}$. Thus, $N_1 = 2$.

- To determine $N_2$, we list all non-neighbor subsets of $S_2 = \{1, 2\}$: These are $\emptyset$, $\{1\}$, and $\{2\}$. Thus, $N_2 = 3$.

- To determine $N_3$, we list all non-neighbor subsets of $S_3 = \{1, 2, 3\}$: These are $\emptyset$, $\{1\}$, $\{2\}$, $\{3\}$, and $\{1, 3\}$. Thus, $N_3 = 5$.

Let $n \geq 3$. Let us see how we can determine $N_n$. Each non-neighbor subset $T$ of $S_n$ is of exactly one of the following two types:

- $T$ does not contain $n$. Then $T$ is a non-neighbor subset of $S_{n-1}$. There are $N_{n-1}$ of these.

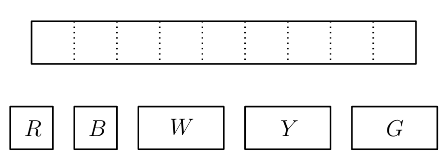
- $T$ contains $n$. Then $T$ does not contain $n - 1$ and $T \setminus \{n\}$ is a non-neighbor subset of $S_{n-2}$. There are $N_{n-2}$ of these.

Thus, the number of non-neighbor subsets of $S_n$ is equal to $N_{n-1} + N_{n-2}$. On the other hand, this number is also equal to $N_n$, because this is how we defined $N_n$. We conclude that

$$N_1 = 2, N_2 = 3, \text{ and for } n \geq 3, \; N_n = N_{n-1} + N_{n-2}.$$

This is a shifted Fibonacci sequence and it follows that $N_n = f_{n+2}$.

**Question 6:** Let $n$ be a positive integer and consider a $1 \times n$ board $B_n$ consisting of $n$ cells, each one having sides of length one. The top part of the figure below shows $B_9$.




You have an unlimited supply of *bricks*, which are of the following types (see the bottom part of the figure above):

- There are red ($R$) and blue ($B$) bricks, both of which are $1 \times 1$ cells.
- There are white ($W$), yellow ($Y$), and green ($G$) bricks, all of which are $1 \times 2$ cells.

A *tiling* of the board $B_n$ is a placement of bricks on the board such that

- the bricks exactly cover $B_n$ and
- no two bricks overlap.

In a tiling, a color can be used more than once and some colors may not be used at all. The figure below shows a tiling of $B_9$, in which each color is used and the color red is used twice.

| | | | | | |
|---|---|---|---|---|---|
| $B$ | $W$ | $R$ | $G$ | $R$ | $Y$ |

Let $T_n$ be the number of different tilings of the board $B_n$.

- Determine $T_1$ and $T_2$.
- Let $n \geq 3$ be an integer. Prove that

$$T_n = 2 \cdot T_{n-1} + 3 \cdot T_{n-2}.$$

- Prove that for any integer $n$,

$$2(-1)^{n-1} + 3(-1)^{n-2} = (-1)^n.$$

- Prove that for any integer $n \geq 1$,

$$T_n = \frac{3^{n+1} + (-1)^n}{4}.$$

**Solution:**

- To determine $T_1$: There are two ways to tile the board $B_1$: $R$ and $B$. Thus, $T_1 = 2$.
- To determine $T_2$: There are seven ways to tile the board $B_2$: $RR$, $RB$, $BR$, $BB$, $W$, $Y$, and $G$. Thus, $T_2 = 7$.

Next, we derive the recurrence: Let $n \geq 3$. Each of the $T_n$ many tilings of the board $B_n$ is of exactly one of the following five types:

- It starts with $R$ and is followed by a tiling of the board $B_{n-1}$. The number of such tilings is equal to $T_{n-1}$.

- It starts with $B$ and is followed by a tiling of the board $B_{n-1}$. The number of such tilings is equal to $T_{n-1}$.
- It starts with $W$ and is followed by a tiling of the board $B_{n-2}$. The number of such tilings is equal to $T_{n-2}$.
- It starts with $Y$ and is followed by a tiling of the board $B_{n-2}$. The number of such tilings is equal to $T_{n-2}$.
- It starts with $G$ and is followed by a tiling of the board $B_{n-2}$. The number of such tilings is equal to $T_{n-2}$.

By taking the sum of all these cases, it follows that

$$T_n = 2 \cdot T_{n-1} + 3 \cdot T_{n-2}.$$

Next we show that

$$2(-1)^{n-1} + 3(-1)^{n-2} = (-1)^n.$$

A fancy solution is as follows:

$$\begin{aligned} 2(-1)^{n-1} + 3(-1)^{n-2} &= -2(-1)^n + 3(-1)^n \\ &= (-2+3)\,(-1)^n \\ &= (-1)^n. \end{aligned}$$

In a less fancy solution, we consider two cases:

- If $n$ is even, then $(-1)^{n-1} = -1$ and $(-1)^{n-2} = 1$, and we get
$$2(-1)^{n-1} + 3(-1)^{n-2} = -2 + 3 = 1 = (-1)^n.$$
- If $n$ is odd, then $(-1)^{n-1} = 1$ and $(-1)^{n-2} = -1$, and we get
$$2(-1)^{n-1} + 3(-1)^{n-2} = 2 - 3 = -1 = (-1)^n.$$

The final step is to prove by induction that

$$T_n = \frac{3^{n+1} + (-1)^n}{4}.$$

There are two base cases:

- If $n = 1$: We have seen above that $T_1 = 2$. Since
$$\frac{3^{n+1} + (-1)^n}{4} = \frac{3^{1+1} + (-1)^1}{4} = \frac{9-1}{4} = \frac{8}{4} = 2,$$
the claim is true for $n = 1$.

- If $n = 2$: We have seen above that $T_2 = 7$. Since

$$\frac{3^{n+1} + (-1)^n}{4} = \frac{3^{2+1} + (-1)^2}{4} = \frac{27 + 1}{4} = \frac{28}{4} = 7,$$

  the claim is true for $n = 2$.

Now let $n \geq 3$ and assume the claim is true for $n - 1$ and $n - 2$, i.e., assume that

$$T_{n-1} = \frac{3^n + (-1)^{n-1}}{4}$$

and

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

Using the recurrence, basic algebra, and the equation in the third part, we get

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

**Question 7:** Those of you who come to class will remember that Jennifer loves to drink India Pale Ale (IPA). After a week of hard work, Jennifer goes to the pub and runs the following recursive algorithm, which takes as input an integer $n \geq 1$, which is a power of 4:

```
Algorithm JenniferDrinksIPA(n):

    if n = 1
    then place one order of chicken wings
    else for k = 1 to 4
           do JenniferDrinksIPA(n/4);
              drink n pints of IPA
           endfor
    endif
```

For $n$ a power of 4, let

- $P(n)$ be the number of pints of IPA that Jennifer drinks when running algorithm JenniferDrinksIPA($n$),

- $C(n)$ be the number of orders of chicken wings that Jennifer places when running algorithm JENNIFERDRINKSIPA$(n)$.

Determine the values of $P(n)$ and $C(n)$. Show your work.

**Solution:** We start with $C(n)$. It follows from the algorithm that

$$C(1) = 1.$$

Let $n \geq 4$. Algorithm JENNIFERDRINKSIPA$(n)$ makes 4 recursive calls to JENNIFERDRINKSIPA$(n/4)$. In each of these recursive calls, Jennifer places $C(n/4)$ orders of chicken wings. Thus,

$$C(n) = 4 \cdot C(n/4).$$

By determining some vales of $C(n)$ for small values of $n$, you will guess that

$$C(n) = n$$

for each $n \geq 1$ that is a power of 4. We prove by induction that this is indeed the case:

The base case is when $n = 1$. Since $C(1) = 1$ and $n = 1$, the base case holds.

Let $n \geq 4$ be a power of 4 and assume that

$$C(n/4) = n/4.$$

Then

$$C(n) = 4 \cdot C(n/4) = 4 \cdot n/4 = n.$$

Next we determine $P(n)$. It follows from the algorithm that

$$P(1) = 0.$$

Let $n \geq 4$. Algorithm JENNIFERDRINKSIPA$(n)$ makes 4 recursive calls to JENNIFERDRINKSIPA$(n/4)$. In each of these recursive calls, Jennifer drinks $n + P(n/4)$ pints of IPA. Thus,

$$P(n) = 4n + 4 \cdot P(n/4).$$

One way to solve this recurrence is to guess that

$$P(n) = 2n \log n$$

for each $n \geq 1$ that is a power of 4. We prove by induction that this guess is correct:

If $n = 1$, then $P(1) = 0$ and $2n \log n = 2 \cdot 1 \log 1 = 0$. Thus the base case holds.

Let $n \geq 4$ be a power of 4 and assume that

$$P(n/4) = 2 \cdot (n/4) \log(n/4).$$

Then

$$\begin{aligned} P(n) &= 4n + 4 \cdot P(n/4) \\ &= 4n + 4\,(2 \cdot (n/4) \log(n/4)) \\ &= 4n + 2n \log(n/4) \\ &= 4n + 2n\,(\log n - \log 4) \\ &= 4n + 2n\,(\log n - 2) \\ &= 2n \log n. \end{aligned}$$

Note that this is a correct solution to the question: We guessed the answer and then verified it. In order to get full marks, you do not have to explain how you got the answer, as long as you prove that your answer is correct.

In case you want to know how to get the answer, we use unfolding, just as we did in class for MergeSort. This gives the second way to get full marks:

$$\begin{aligned} P(n) &= 4n + 4 \cdot P(n/4) \\ &= 4n + 4\left(4 \cdot n/4 + 4 \cdot P(n/4^2)\right) \\ &= 2 \cdot 4n + 4^2 \cdot P(n/4^2) \\ &= 2 \cdot 4n + 4^2 \cdot \left(4 \cdot n/4^2 + 4 \cdot P(n/4^3)\right) \\ &= 3 \cdot 4n + 4^3 \cdot P(n/4^3) \\ &= 3 \cdot 4n + 4^3 \cdot \left(4 \cdot n/4^3 + 4 \cdot P(n/4^4)\right) \\ &= 4 \cdot 4n + 4^4 \cdot P(n/4^4). \end{aligned}$$

At this moment, you will see the pattern. After $k$ unfolding steps, we get

$$P(n) = k \cdot 4n + 4^k \cdot P(n/4^k).$$

Let $k$ be such that $n = 4^k$. Then $n = 2^{2k}$ and $\log n = 2k$. We get

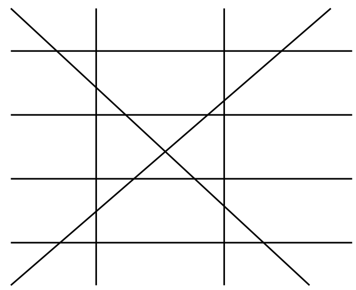
$$\begin{aligned} P(n) &= \frac{\log n}{2} \cdot 4n + n \cdot P(n/n) \\ &= \frac{\log n}{2} \cdot 4n + n \cdot P(1) \\ &= \frac{\log n}{2} \cdot 4n \\ &= 2n \log n. \end{aligned}$$

**Question 8:** A line is called *slanted* if it is neither horizontal nor vertical. Let $k \geq 1$, $m \geq 1$, and $n \geq 0$ be integers. Consider $k$ horizontal lines, $m$ vertical lines, and $n$ slanted lines, such that

- no two of the slanted lines are parallel,

- no three of the $k+m+n$ lines intersect in one single point.

These lines divide the plane into regions (some of which are bounded and some of which are unbounded). Denote the number of these regions by $R_{k,m,n}$. From the figure below, you can see that $R_{4,2,2} = 30$.



- Prove that
$$R_{k,m,0} = (k+1)(m+1).$$

  **Solution:** The $k$ horizontal lines divide the plane into $k+1$ horizontal slabs, whereas the $m$ vertical lines divide the plane into $m+1$ vertical slabs. If we overlay these $k+1$ horizontal slabs and $m+1$ vertical slabs, then we obtain exactly $(k+1)(m+1)$ regions. Thus, $R_{k,m,0} = (k+1)(m+1)$.

- Derive a recurrence for the numbers $R_{k,m,n}$ and use it to prove that
$$R_{k,m,n} = (k+1)(m+1) + (k+m)n + \binom{n+1}{2}.$$

**Solution:** We fix $k$ and $m$. Let $n \geq 1$ and consider $n$ slanted lines $L_1, L_2, \ldots, L_n$.

- We start by adding the lines $L_1, L_2, \ldots, L_{n-1}$ to the $k$ horizontal lines and $m$ vertical lines. At this moment, the number of regions is $R_{k,m,n-1}$.
- Now we add the line $L_n$. This line intersects
  - each of the $k$ horizontal lines,
  - each of the $m$ vertical lines, and
  - each of the lines $L_1, L_2, \ldots, L_{n-1}$.
  - Thus, there are $k+m+n-1$ intersections between $L_n$ and the lines we have already drawn. It follows that $L_n$ goes through $k+m+n$ regions (one more than the number of intersections), and each such region is cut into two. It follows that, when adding $L_n$, the number of regions increases by $k+m+n$.

Thus, we obtain the recurrence

$$R_{k,m,n} = R_{k,m,n-1} + k + m + n.$$

It remains to prove that

$$R_{k,m,n} = (k+1)(m+1) + (k+m)n + \binom{n+1}{2}.$$

We do this by induction on $n$. If $n = 0$, then $R_{k,m,n} = R_{k,m,0} = (k+1)(m+1)$ and

$$(k+1)(m+1) + (k+m)0 + \binom{0+1}{2} = (k+1)(m+1)$$

as well, proving the base case.

Let $n \geq 1$, and assume the claim is true for $n-1$, i.e., assume that

$$R_{k,m,n-1} = (k+1)(m+1) + (k+m)(n-1) + \binom{n}{2}.$$

We have to show that

$$R_{k,m,n} = (k+1)(m+1) + (k+m)n + \binom{n+1}{2}.$$

This follows by applying the recurrence and the assumption:

$$\begin{aligned} R_{k,m,n} &= R_{k,m,n-1} + k + m + n \\ &= (k+1)(m+1) + (k+m)(n-1) + \binom{n}{2} + k + m + n \\ &= (k+1)(m+1) + (k+m)n + \binom{n}{2} + n. \end{aligned}$$

Since

$$\begin{aligned} \binom{n}{2} + n &= \frac{n(n-1)}{2} + n \\ &= \frac{n(n-1) + 2n}{2} \\ &= \frac{n(n+1)}{2} \\ &= \binom{n+1}{2}, \end{aligned}$$

it follows that

$$R_{k,m,n} = (k+1)(m+1)(k+m)n + \binom{n+1}{2}.$$