COMP 2804 — Solutions Assignment 2

Question 1:

- Write your name and student number.

Solution: Johan Cruijff, 14

Question 2: Consider a finite set S of points in the two-dimensional plane. A point p of S is called a *covidiot*, if p is within two meters of some other point of S.

Let R be a rectangle whose horizontal sides have a length of 20 meters and whose vertical sides have a length of 30 meters. Assume that all points of S are contained in R and that S contains at least 601 points.

- Use the Pigeonhole Principle to prove that S contains at least two covidiots.

Solution: To apply the Pigeonhole Principle, we are going to divide the rectangle R into 600 "pieces". Each point of S belongs to a unique "piece" and, therefore, there is a "piece" that contains at least two points of S, say p and q. The "pieces" must be chosen such that both p and q are covidiots.

What are these "pieces"? The fact that $600 = 20 \cdot 30$ suggests that we divide the rectangle R into *cells*, each one being a square with sides of length 1 meter. The number of cells is equal to $20 \cdot 30 = 600$. Each point of S is contained in a unique cell. (If a point is on the boundaries of several cells, then we assign it to exactly one cell.) Thus, we have 600 "boxes" (i.e., the cells), and we have placed at least 601 points in these boxes. By the Pigeonhole Principle, there is a cell that contains at least two points of S. Let p and q be two such points that are in the same cell. The distance between p and q is at most the diagonal of the cell, which, by Pythagoras, is $\sqrt{2} < 2$ meters. Thus, the distance between p and q is less than two meters and, therefore, both p and q are covidiots.

Note: The claim is still true if we replace 601 by a much smaller number. Using 601 leads to a very simple proof, as shown above.

Question 3:

- Let $\alpha_1, \alpha_2, \ldots, \alpha_7$ be real numbers that are all contained in the interval $[-\pi/2, \pi/2]$.

  Use the Pigeonhole Principle to prove that there are two distinct indices i and j such that $0 \leq \alpha_i - \alpha_j \leq \pi/6$.

  Solution: The interval $[-\pi/2, \pi/2]$ has length π. We divide it into six intervals, each having length $\pi/6$. Each of the seven real numbers is contained in a unique interval. (If a number is on the endpoints of two intervals, then we assign it to, say, the leftmost of these intervals.) By the Pigeonhole Principle, there is an interval that contains at least two numbers. Let i and j, with $i \neq j$, be such that α_i and α_j are in the same interval. We may assume that $\alpha_j \leq \alpha_i$; otherwise, we swap i and j. Then, $0 \leq \alpha_i - \alpha_j$. Since these numbers are in the same interval, we have $\alpha_i - \alpha_j \leq \pi/6$.

- Let $a_1, a_2, \ldots, a_7$ be real numbers such that $a_i a_j \neq -1$ for all $i \neq j$.

  Prove that there are two distinct indices i and j such that

  $$0 \leq \frac{a_i - a_j}{1 + a_i a_j} \leq \frac{1}{\sqrt{3}}.$$

  Hint: For each i, let p_i be the point with coordinates $(1, a_i)$, and consider the angle between the x-axis and the vector from the origin to p_i. You learned in high school that

  $$\tan(\alpha - \beta) = \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta}.$$

  Solution: Let α_i be the angle between the x-axis and the vector from the origin to p_i. Then $-\pi/2 < \alpha_i < \pi/2$ and $\tan \alpha_i = a_i$.

  Thus we have seven real numbers $\alpha_1, \ldots, \alpha_7$ that are all contained in the interval $(-\pi/2, \pi/2)$. From the first part, there exist two distinct indices i and j such that $0 \leq \alpha_i - \alpha_j \leq \pi/6$. Then

  $$0 \leq \tan(\alpha_i - \alpha_j) \leq \tan(\pi/6) = 1/\sqrt{3}.$$

  Since

  $$\tan(\alpha_i - \alpha_j) = \frac{\tan \alpha_i - \tan \alpha_j}{1 + \tan \alpha_i \tan \alpha_j} = \frac{a_i - a_j}{1 + a_i a_j},$$

  the proof is complete.

Question 4: The function $f : \{0, 1, 2, \ldots\} \to \mathbb{R}$ is defined by

$$\begin{array}{rcll} f(0) & = & 7, & \\ f(n) & = & 5 \cdot f(n-1) + 12n^2 - 30n + 15 & \text{if } n \geq 1. \end{array}$$

- Prove that for every integer $n \geq 0$,

  $$f(n) = 7 \cdot 5^n - 3n^2.$$

Solution: The proof is by induction on n.

The base case is when $n = 0$. The left-hand side is $f(0) = 7$ and the right-hand side is $7 \cdot 5^0 - 3 \cdot 0^2 = 7$.

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

$$f(n-1) = 7 \cdot 5^{n-1} - 3(n-1)^2.$$

We have to prove that the claim is true for n. Here we go:

$$
\begin{aligned}
f(n) &= 5 \cdot f(n-1) + 12n^2 - 30n + 15 \\
&= 5 \cdot \left(7 \cdot 5^{n-1} - 3(n-1)^2\right) + 12n^2 - 30n + 15 \\
&= 7 \cdot 5^n - 15(n-1)^2 + 12n^2 - 30n + 15 \\
&= 7 \cdot 5^n - 15\left(n^2 - 2n + 1\right) + 12n^2 - 30n + 15 \\
&= 7 \cdot 5^n - 3n^2.
\end{aligned}
$$

Thus, the claim is true for n.

Question 5: Consider the following recursive algorithm, which takes as input an integer $n \geq 1$ that is a power of two:

```
Algorithm MYSTERY(n):

    if n = 1
    then return 1
    else x = MYSTERY(n/2);
         return n + x
    endif
```

- Determine the output of algorithm MYSTERY(n) as a function of n. As always, justify your answer.

Solution: After running the algorithm "by hand" for some small powers of n, we "guess" that

for every $n \geq 1$ that is a power of 2, MYSTERY(n) returns $2n - 1$.

We prove by induction that our guess is correct.

Base case: The smallest power of two is $n = 2^0 = 1$. In this case, the algorithm returns 1, which is $2n - 1$.

Induction step: Let $n \geq 2$ be a power of two and assume that MYSTERY($n/2$) returns $2 \cdot (n/2) - 1 = n - 1$. (Note that $n/2$ is a power of two.)

Now we run MYSTERY(n). It first runs MYSTERY($n/2$), which, by assumption, returns $n - 1$. Thus, $x = n - 1$. Then, MYSTERY(n) returns $n + x$, which is $n + (n - 1) = 2n - 1$.

Question 6: In class, we have seen that for any integer $m \geq 1$, the number of 00-free bitstrings of length m is equal to f_{m+2}, which is the $(m + 2)$-th Fibonacci number.

Let $n \geq 4$ be an integer and consider the set

$$B = \{(x, y) : \text{both } x \text{ and } y \text{ are 00-free bitstrings of length } n - 1\}.$$

- Explain, in a few sentences, why $|B| = f_{n+1}^2$.

  Note that f_{n+1}^2 is the square of f_{n+1}, i.e., it is to be read as $(f_{n+1})^2$.

  Solution: There are f_{n+1} choices for x and f_{n+1} choices for y. By the Product Rule, $|B| = f_{n+1} \cdot f_{n+1} = f_{n+1}^2$.

- Determine the number of elements (x, y) in B for which the concatenation xy is 00-free.

  Solution: If the concatenation xy is 00-free, then both x and y are 00-free as well; thus, (x, y) is in B.

  Since the length of xy is $2(n-1) = 2n-2$, the number of such xy is equal to the number of 00-free strings of length $2n-2$, which is f_{2n}.

- Determine the number of elements (x, y) in B for which the concatenation xy is not 00-free.

  Solution: The only way that xy is not 00-free is when x ends with 0 and y starts with 0.

  1. Since x is 00-free and ends with 0, its second bit from the right is 1. Thus, we can write $x = x'10$, where x' has length $n-3$. This x' can be an arbitrary 00-free string of length $n-3$. Thus, there are f_{n-1} choices for x'.
  2. Since y is 00-free and starts with 0, its second bit from the left is 1. Thus, we can write $y = 01y'$, where y' has length $n-3$. This y' can be an arbitrary 00-free string of length $n-3$. Thus, there are f_{n-1} choices for y'.
  3. By the Product Rule, the number of choices for (x', y') is $f_{n-1} \cdot f_{n-1} = f_{n-1}^2$.

- Use the previous results to prove that

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

  Solution: We have seen that $|B| = f_{n+1}^2$.

  We divide B into two groups:

  1. Those (x, y) in B for which xy is 00-free. We have seen above that there are f_{2n} many of these.
  2. Those (x, y) in B for which xy is not 00-free. We have seen above that there are f_{n-1}^2 many of these.

  By the Sum Rule, we get $|B| = f_{2n} + f_{n-1}^2$.

  Thus, we have shown that

  $$f_{n+1}^2 = f_{2n} + f_{n-1}^2,$$

  which is equivalent to what we are asked to prove.

Question 7: The sequence $s_0, s_1, s_2, \ldots$ of bitstrings is recursively defined as follows:

1. $s_0 = 1$, i.e., the bitstring of length one consisting of one 1.
2. For $n \geq 1$, the bitstring s_n is obtained as follows: In the bitstring s_{n-1}, replace each 1 by 10 and replace each 0 by 1.

The first few strings in this sequence are

$$s_0 = 1; s_1 = 10; s_2 = 101; s_3 = 10110; s_4 = 10110101.$$

- Prove that for each $n \geq 0$, the bitstring s_n is 00-free.

  Solution: For $n = 0$, we have $s_0 = 1$, which is 00-free. Assume that $n \geq 1$. By construction, each 0 in s_n has 1 to its left. Therefore, s_n does not contain 00 and, thus, s_n is 00-free.

For each $n \geq 0$, let L_n be the length of s_n and let O_n be the number of 1's in s_n.

- Determine L_0, L_1, O_0, and O_1.

  Solution: Since $s_0 = 1$, we have $L_0 = 1$ and $O_0 = 1$. Since $s_1 = 10$, we have $L_1 = 2$ and $O_1 = 1$.

- Let $n \geq 2$. Prove that $O_n = L_{n-1}$.

  Solution: How do we obtain the string s_n: We take s_{n-1} and replace each 1 by 10 and replace each 0 by 1. Thus, each bit in s_{n-1} leads to one 1 in s_n. As a result, the number of bits in s_{n-1} is equal to the number of 1's in s_n. That is, $L_{n-1} = O_n$.

- Let $n \geq 2$. Prove that $L_n = L_{n-1} + O_{n-1}$.

  Solution: How do we obtain the string s_n: We take s_{n-1} and replace each 1 by 10 and replace each 0 by 1. Thus, each 1 in s_{n-1} increases the number of bits in s_n by one, and each 0 in s_{n-1} does not change the number of bits in s_n.

  As a result, the length of s_n is equal to the length of s_{n-1} plus the number of 1's in s_{n-1}. That is, $L_n = L_{n-1} + O_{n-1}$.

- Express O_n in terms of numbers that we have seen in class.

  Solution: We do some algebra. For $n \geq 3$,

$$\begin{aligned} O_{n-1} &= L_{n-2} && \text{(in part 3, replace } n \text{ by } n-1) \\ &= L_{n-1} - O_{n-2} && \text{(in part 4, replace } n \text{ by } n-1) \\ &= O_n - O_{n-2} && \text{(from part 3)} \end{aligned}$$

  Thus, we have

$$O_n = O_{n-1} + O_{n-2}.$$

  You can verify that this is also true for $n = 2$. Thus, O_n satisfies the Fibonacci recurrence. Look at the following table:

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	1	1	2	3	5	8	13
	O_0	O_1	O_2	O_3	O_4	O_5	O_6

From this table, we see that $O_n = f_{n+1}$ for each $n \geq 0$.

For each $n \geq 0$, let Z_n be the number of 0's in s_n.

- Express Z_n in terms of numbers that we have seen in class.

  Solution: For $n \geq 1$, from part 3 (with n replaced by $n+1$), we have $L_n = O_{n+1}$. You can verify that this is also true for $n = 0$. Thus, we have $L_n = f_{n+2}$ for each $n \geq 0$.

  Observe that $L_n = O_n + Z_n$. Therefore, for $n \geq 0$,

$$Z_n = L_n - O_n = f_{n+2} - f_{n+1} = f_n.$$

Question 8: Consider a bitstring $b_1 b_2 \dots b_n$ of length n, where $n \geq 2$ is an integer. For any integer i, the bit b_i is called *lonely*, if $b_i = 1$ and its neighboring bits are 0. More formally, the bit b_i is lonely if

1. $i = 1$, $b_1 = 1$, and $b_2 = 0$, or
2. $i = n$, $b_n = 1$, and $b_{n-1} = 0$, or
3. $2 \leq i \leq n-1$, $b_i = 1$, and $b_{i-1} = b_{i+1} = 0$.

For example, in the following bitstring, the three bits in boldface are lonely:

10001110 0**1**0**1**000

A bitstring of length at least two is called *happy* if none of its bits is lonely.

For any integer $n \geq 2$, let A_n denote the number of happy bitstrings of length n, and let B_n denote the number of happy bitstrings of length n that start with 11.

- Determine A_2, A_3, A_4, and A_5.

  Solution: We list all happy bitstrings:

  1. Length 2: Out of the 4 strings of length 2, the following are happy: 00 and 11. Thus, $A_2 = 2$.
  2. Length 3: Out of the 8 strings of length 3, the following are happy: 000; 011; 110; 111. Thus, $A_3 = 4$.

3. Length 4: Out of the 16 strings of length 4, the following are happy:

$$\begin{array}{c} 0000 \\ 0011 \\ 0110 \\ 1100 \\ 0111 \\ 1110 \\ 1111 \end{array}$$

Thus, $A_4 = 7$.

4. Length 5: Out of the 32 strings of length 5, the following are happy:

$$\begin{array}{c} 00000 \\ 00011 \\ 00110 \\ 01100 \\ 11000 \\ 00111 \\ 01110 \\ 11100 \\ 01111 \\ 11110 \\ 11011 \\ 11111 \end{array}$$

Thus, $A_5 = 12$.

- Let $n \geq 3$ be an integer. Express A_n in terms of A_{n-1} and B_n.

  Solution: The number of happy strings of length n is equal to A_n. We divide them into two groups:

  1. Those that start with 0. If we remove the first bit from these strings, then we obtain all happy strings of length $n-1$. Thus, there are A_{n-1} many strings in this group.
  2. Those that start with 1. In each such string, the second bit must be 1. The number of strings in this group is equal to the number of happy strings of length n that start with 11, which is B_n.

  By the Sum Rule, we have
  $$A_n = A_{n-1} + B_n. \tag{1}$$

- Let $n \geq 4$ be an integer. Express A_{n-1} in terms of A_{n-2} and B_{n-1}.

  Solution: In the previous part, we replace n by $n-1$. This gives
  $$A_{n-1} = A_{n-2} + B_{n-1}. \tag{2}$$

- Let $n \geq 5$ be an integer. Express B_n in terms of A_{n-3} and B_{n-1}.

  Solution: The number of happy strings of length n that start with 11 is equal to B_n. We divide them into two groups:

  1. Those that start with 110. If we remove the first three bits from these strings, then we obtain all happy strings of length $n - 3$. Thus, there are A_{n-3} many strings in this group.
  2. Those that start with 111. If we remove the first bit from these strings, then we obtain all happy strings of length $n - 1$ that start with 11. Thus, there are B_{n-1} many strings in this group.

  By the Sum Rule, we have

$$B_n = A_{n-3} + B_{n-1}. \tag{3}$$

- Let $n \geq 5$ be an integer. Prove that

$$A_n = 2 \cdot A_{n-1} - A_{n-2} + A_{n-3}.$$

  Solution: If we apply (1) and (3), we get

$$\begin{aligned} A_n &= A_{n-1} + B_n \\ &= A_{n-1} + A_{n-3} + B_{n-1}. \end{aligned} \tag{4}$$

  If we subtract (4) from (2), we get

$$A_n - A_{n-1} = A_{n-1} + A_{n-3} - A_{n-2}.$$

  By re-arranging, we get

$$A_n = 2 \cdot A_{n-1} - A_{n-2} + A_{n-3}.$$