Drexel University, Department of Mathematics

# Math 235: Mathematical Problem Solving, Fall 2020: Homework 1

Darij Grinberg

September 29, 2020

## 1 Exercise 1

### 1.1 Problem

Let $k$ be a positive integer. A set $S$ of integers is said to be *$k$-lacunar* if every two distinct elements $u, v \in S$ satisfy $|u - v| > k$. (Thus, a 1-lacunar set is the same as a lacunar set as defined in the notes.)

Let $n \in \mathbb{N}$. Let $S$ be a $k$-lacunar subset of $\{1, 2, \ldots, n\}$. Prove that

$$|S| \leq \frac{n + k - 1}{k}.$$

(Thus, in particular, if $S$ is a lacunar subset of $\{1, 2, \ldots, n\}$, then $|S| \leq \dfrac{n+1}{2}$.)

### 1.2 Solution

...

# 2 Exercise 2

## 2.1 Problem

Let $(f_0, f_1, f_2, \ldots)$ be the Fibonacci sequence. Prove that any $n, m \in \mathbb{N}$ satisfy

$$\gcd(f_n, f_m) = f_{\gcd(n,m)}.$$

## 2.2 Solution

...

---

# 3 Exercise 3

## 3.1 Problem

Let $(F_0, F_1, F_2, \ldots)$ be the *Fermat sequence* – that is, the sequence of integers defined by

$$F_n = 2^{2^n} + 1 \qquad \text{for each } n \in \mathbb{N}.$$

(Keep in mind that nested powers are to be read top-to-bottom: That is, the expression "$a^{b^c}$" means $a^{(b^c)}$ rather than $\left(a^b\right)^c$.)

**(a)** Prove that

$$F_n = F_0 F_1 \cdots F_{n-1} + 2 \qquad \text{for every integer } n > 0.$$

**(b)** Prove that $\gcd(F_n, F_m) = 1$ for any two distinct positive integers $n$ and $m$.

## 3.2 Solution

...

---

# 4 Exercise 4

## 4.1 Problem

Prove that there exist **infinitely many odd** positive integers $n$ for which

$$\frac{1! \cdot 2! \cdot \cdots \cdot (2n)!}{(n+1)!} \text{ is a perfect square.}$$

## 4.2 Solution

...

---

# 5 Exercise 5

## 5.1 Problem

Let $x \in \mathbb{R}$. Let $n$ and $m$ be positive integers. Prove that

$$\sum_{k=0}^{mn-1} \left\lfloor x + \frac{k}{n} \right\rfloor = m \left( \lfloor nx \rfloor + \frac{n(m-1)}{2} \right).$$

## 5.2 Solution

...

---

# 6 Exercise 6

## 6.1 Problem

Let $(f_0, f_1, f_2, \ldots)$ be the Fibonacci sequence. Prove that each nonnegative integer $n$ satisfies

$$f_n \equiv f_{n\%5} \cdot 3^{n//5} \mod 5.$$

(Here, $n//5$ and $n\%5$ denote the quotient and the remainder obtained when dividing $n$ by 5.)

## 6.2 Solution

...

---

# 7 Exercise 7

## 7.1 Problem

Let $a$ and $b$ be two **positive** integers. Prove that there exist **positive** integers $x$ and $y$ such that

$$\gcd(a, b) = xa - yb.$$

## 7.2 Solution

...

---

# 8 Exercise 8

## 8.1 Problem

Prove that any positive integer $a$ can be uniquely expressed in the form

$$a = 3^m + b_{m-1}3^{m-1} + b_{m-2}3^{m-2} + \cdots + b_0 3^0$$

where $m$ is a nonnegative integer, and where $b_0, b_1, \ldots, b_{m-1} \in \{0, 1, -1\}$. (This is called the *balanced ternary representation.*)

## 8.2 Solution

...

---

# 9 Exercise 9

## 9.1 Problem

Let $p, q, m, n \in \mathbb{N}$ with $p \leq m$ and $q \leq n$. Consider an $m \times n$-table $T$ of integers, with all entries distinct. In each column of $T$, we mark the $p$ largest entries with a *cyan* marker. In each row of $T$, we mark the $q$ largest entries with a *red* marker. Prove that at least $pq$ entries of $T$ are marked twice (i.e., with both colors).

[**Example:** Let $p = 2$ and $q = 2$ and $m = 3$ and $n = 3$ and

$$T = \begin{pmatrix} 1 & 2 & 9 \\ 4 & 3 & 8 \\ 5 & 6 & 7 \end{pmatrix}.$$

Then,

$$\text{the cyan entries are } 4, 5, 3, 6, 8, 9, \text{ while}$$
$$\text{the red entries are } 2, 9, 4, 8, 6, 7.$$

Thus, the entries $4, 6, 8, 9$ are marked twice. This is exactly the $pq$ entries claimed in the exercise. You can easily find situations in which there are more than $pq$ doubly-marked entries.]

## 9.2 Solution

...

---

# 10 Exercise 10

## 10.1 Problem

A *bitstring* shall mean a finite sequence consisting of 0's and 1's. (This is what we called an "$n$-bitstring" in the Gray code example in the notes, except that the length is no longer fixed.) We shall write our bitstrings without commas and parentheses – i.e., we shall simply write $a_1 a_2 \cdots a_n$ for the bitstring $(a_1, a_2, \ldots, a_n)$.

Bitstrings can be transformed by *moves*. In each *move*, you pick two consecutive entries 01 in the bitstring (appearing in this order), and replace them by three consecutive entries 100 (in this order). For example, here is a sequence of moves:

$$01\underline{01} \to \underline{01}100 \to 10\underline{01}00 \to 1\underline{01}0000 \to 11000000$$

(where we use underscores to mark the places where the moves are happening). Note that the last bitstring in this sequence has no two consecutive entries 01 any more, and thus no more moves can be applied to it.

**(a)** Prove that there are no infinite sequences of moves. That is, if you start with a bitstring $a$, then any sequence of moves that can be applied successively must have an end.

[**Hint:** Given a bitstring $a$, let $n_a$ denote the number of 0s in $a$, and let $k_a$ be the number of 1s in $a$. Induct on $k_a$. Within the induction step, induct on $n_a$.]

**(b)** **(optional; to be discussed later)** A bitstring shall be called *immovable* if no move can be applied to it. Part **(a)** shows that, starting with any bitstring $a$, we can always reach an immovable bitstring by performing moves until no more moves are possible. Prove that this immovable bitstring is uniquely determined by $a$ – that is, no matter how you perform the moves, the immovable bitstring that results at the end will be the same. Moreover, the number of moves needed to reach the immovable bitstring will be the same.

## 10.2 Solution

...

---

# References