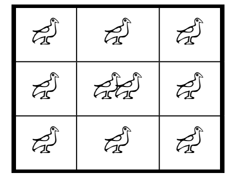
The pigeonhole principle

ICTS-RRI Maths Circle, Bengaluru

13, 27 April 2024, 10:00 am to 1:00 pm



The principle

> Let n be a positive integer. If $n + 1$ balls are placed in n bins, then some bin must have at least two balls.

The principle is simple, yet it often leads to very pleasing conclusions[1].

[1] https://archive.org/details/mathematical-circles-russian-experience/page/31/mode/1up

Practice problems

Attempt the problems in the book

> Dmitri Fomin, Sergey Genkin, Ilia Itenberg: *Mathematical Circles (Russian Experience)*, (click here).

Example. There are 20 guests at a party. Some guests shake hands with other guests, but no pair of guests shake hands twice. Show that there are two guests who shake hands exactly the same number of times. (To use the above pigeonhole principle, ask yourself the following: what are the balls, what are the bins?) 

It might help to visualize the pigeonhole principle using a picture. Draw a graph with vertices and edges as follows. On the left, keep a vertex for each ball; call the vertices $a_1, a_2, \ldots, a_{n+1}$. On the right, keep a vertex for each bin; call these vertices $b_1, b_2, \ldots, b_n$. If ball i is placed in bin j, draw a directed edge (an arrow) from a_i to b_j. The point is the following. The total number of arrows leaving the vertices on the left is exactly $n + 1$; so the total number of arrows landing on the vertices on the right must also be $n + 1$. Let d_j be the number of arrows that land on b_j. Then,

$$n + 1 = d_1 + d_2 + \cdots d_n \leq n \cdot \max_j d_j.$$

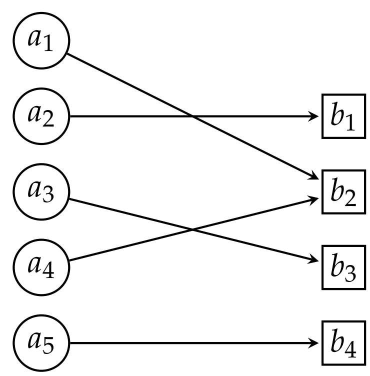


Figure 1: Five balls placed in four bins

So $\max_j d_j \geq \frac{n+1}{n} = 1 + \frac{1}{n}$. But $\max_j d_j$ is an integer, so it must be at least 2. Notice that the quantity $1 + \frac{1}{n}$ is the average number of balls per bin. We simply combined two elementary facts: (i) in any collection of numbers, the maximum is always at least their average; (ii) an integer that is more than one must be at least two. In general, following this approach we arrive at a somewhat more general version of the pigeonhole principle.

> Suppose m balls are placed in n bins, then
>
> 1. there is a bin containing at least m/n balls; and
> 2. there is a bin containing at most m/n balls.

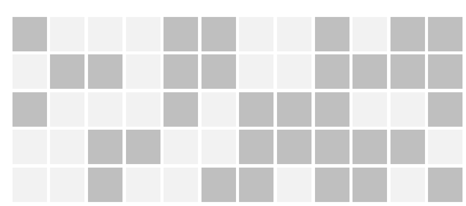


Figure 2: A 5×12 table

Example. Each cell of a 5×41 table is coloured either *black* or *white*. Show that there are three rows and three columns so that all nine cells in their intersection have the same colour. This is a little tricky, but please give it a try. (I don't want to spoil the challenge for you, but here are some imprecise hints, if you need them. First, focus on the columns. In this column, some three of the cells must be of the same colour (why?). Now, think of the columns as balls and put them in 10 bins. Why 10 bins? What are the bins for? When is a column placed in a bin? Conclude that some bin has at least . . . balls; then . . . .)

Exploration I

Consider a sequence of distinct integers, e.g.,

$$s = 1, 5, 19, 4, 7, 22, 6, 8, 15, 11\,.$$

A subsequence of this sequence is obtained by removing some of the numbers in the sequence and retaining the rest. For example,

$$s_1 = 1, 4, 6, 8, 11$$
$$s_2 = 19, 15, 11$$

are subsequences of the sequence s. A sequence is *monotone increasing* if each element in the sequence is greater than the previous element (as in s_1); it is *monotone decreasing* if each element in the sequence is less than the previous element (as in s_2).

Explore: Construct a sequence t by reordering the numbers 1,2 . . . , 16, so that t contains neither a monotone increasing nor a monotone decreasing subsequence of five elements. Do you think every such sequence obtained by reordering 1,2 . . . , 17 always has a monotone subsequence with 5 elements? Why? Is it true that every infinite sequence of distinct integers has an infinite monotone subsequence.

Exploration II

Consider a graph where the vertices represent locations in a city and the edges represent two-way streets connecting adjacent locations. There is a robber in the city. We wish to determine the minimum number of cops needed to catch the robber. Here are the rules. First, the cops occupy some vertices in the graph. Then, the robber places herself on some vertex of her choice.

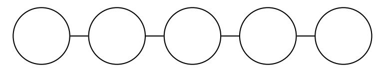


Figure 3: The path graph P_5

The cops and robber make their moves in alternate time-steps. The cops traverse one edge at a time; the robber can run through several edges in one step. If it is the cops' turn, one of the cops moves to an adjacent vertex (we allow dummy moves, where no cop moves; we also allow multiple cops to occupy the same vertex of the graph). If it is the robber's turn, she can move to another vertex along a path on which *none of the intermediate vertices is occupied by a cop*, or she can stay put. If at any time a cop and a robber are at the same vertex, the robber is caught.

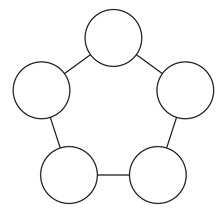


Figure 4: The cycle graph C_5

The game is played with perfect information, the cops know where the robber is at each point, and the robber knows where the cops are.

(a) How many cops does one need to catch a robber on the path P_5? How many does one need for the cycle C_8? What about paths and cycles with more vertices?

(b) Consider the 8×8 *grid graph* $G_{8\times 8}$. Show that with eight cops there is a strategy for the cops to catch the robber no matter how the robber tries to evade them. What if we have only seven cops? What if we play the game on an $n \times n$ grid graph $G_{n\times n}$?

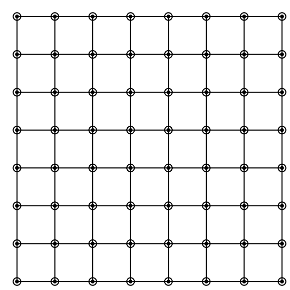


Figure 5: The grid graph $G_{8\times 8}$

Exploration III

I asked ChatGPT the following question:

> Write the numbers $1, 2, \ldots, 64$ in the cells of the 8×8 board, different numbers in different cells, in such a way that adjacent numbers differ by as little as possible.

After thinking a little, ChatGPT said this:

> `To minimize the difference between adjacent numbers, we can arrange them in a serpentine pattern. Here's one possible arrangement.` (It gave the arrangement on the right.)

Then it said:

> `In this arrangement, adjacent numbers typically differ by 1, except for a few places where they differ by 8. This arrangement ensures that adjacent numbers have the least possible difference on average.`

01	02	03	04	05	06	07	08
16	15	14	13	12	11	10	09
17	18	19	20	21	22	23	24
32	31	30	29	28	27	26	25
33	34	35	36	37	38	39	40
48	47	46	45	44	43	42	41
49	50	51	52	53	54	55	56
64	63	62	61	60	59	58	57

Figure 6: ChatGPT's arrangement

Question: Is ChatGPT bluffing? What is the maximum difference between adjacent numbers in the above arrangement? Is there a better arrangement? How much better? Give it a try!

Definitions for our discussion 13 April: We may think of this problem in the language of graphs with vertices and edges. Suppose each vertex of a graph is labelled with a distinct integer. The *stretch* of this labelling is the *maximum* difference between the labels of adjacent vertices. Our goal is to find a labelling with *minimum* stretch. The stretch of the best labelling is the *stretch number* of the graph; we write $\mathsf{sn}(G)$ for the stretch number of the graph G. Both the words, *minimum* and *maximum*, are used above. The order in which they appear is important to the definition; make sure you understand it.

(a) Consider the graphs P_5 and C_5 described on the previous page. What is $\mathsf{sn}(P_5)$? What is $\mathsf{sn}(C_5)$? What is the stretch number of the path graph P_n with n vertices; what is the stretch number of the cycle graph C_n with n vertices?

(b) If the graph has N vertices, we may assume that in the optimal labelling the labels are $1, 2, \ldots, N$. (Why?)

(c) What is the stretch number of the $n \times n$ grid graph $G_{n \times n}$?

We will meet on 13, 27 April 2024, 10:00 am to 1:00 pm.

13 April: Summary of the discussion on Exploration I

After everybody constructed a sequence with 16 elements with no monotone subsequence with 5 elements, we briefly considered why sequences with 17 elements must have a a monotone subsequence of length 5. I did not wait for a solution. Someone suggested this approach.

> Every rearrangement of 16 elements that does not contain a monotone subsequence with 5 elements must be of one of two forms. It is then clear that such a sequence cannot be extended by inserting another element.

I stated that it was not clear why the first sentence of the above claim was true, but did not offer a concrete counterexample. (We will say a little more about this later.) I moved on to the proof using the pigeonhole principle.

Instructions and responses.

```
I: Write below each element of the sequence, the
length of the longest sequence starting from that
element.Suppose the sequence has no monotone
increasing subsequence of $5$ elements.
R: There are 17 numbers, each in the range {1,2,...,4};
so, ..., some number must appear at least 5 times.

I: Ignore this very popular number itself, ..., but
focus on the elements of the sequence under which this
popular number is written.
R: They must form a monotone decreasing subsequence.
```

Someone correctly pointed out that we seem to be using a pigeonhole principle where 17 balls are placed in four bins: $m = 17$, $n = 4$, so some bin must have at least five balls. I put down this theorem.

Theorem 1 (Erdös and Szekeres)**.** *In any sequence of* $k^2 + 1$ *distinct elements, there is a monotone increasing subsequence of _______ elements.*

Everybody soon realized how the original argument could be adapted to justify the theorem with $k + 1$ filled in the blank. Someone asked if "the converse was true," which is not quite what they meant: they wanted to know if something smaller than $k^2 + 1$ could be written in its place, but they immediately saw that they had already answered the question in the exploration with 16 replaced by k^2. The theorem called for an algorithm. Given such a sequence, how do we find a monotone subsequence of $k + 1$ elements. See the algorithm in Figure 7.

Solitaire sorting

Suppose the sequence is $a_1, a_2, \ldots, a_n$. We will move the numbers in the sequence into rows numbered, 0, 1, 2, ...

1. Initially, all the rows are empty; place $-\infty$ in row 0.
2. Consider elements $a_1, a_2, \ldots$ one by one.
   (a) With each element find the last row whose last element is smaller than the current element.
   (b) Insert the current element at the end of the next row.
   (c) Draw an arrow from the current element to the last element of the previous row.

Figure 7: An algorithm to find the Longest Increasing Subsequence (LIS)

Observations: We rushed through these observations. (If you did not catch everything, please try to justify them on your own.)

1. The elements in each row form a monotone decreasing subsequence of the original sequence;
2. If an element a_i is eventually placed in row ℓ, then there is a monotone increasing subsequence with ℓ elements ending with a_i;
3. If the last non-empty row is row L, then there is an increasing subsequence of L elements in the original sequence;
4. If the last non-empty row is row L, then there is no increasing subsequence of $L + 1$ elements in the original sequence.

14, 25, 3, 8, 30, 21, 11, 5, 17, 9, 28, 1, 19, 12, 24, 15, 20, 7, 2, 23, 29, 10, 18, 26, 16, 6, 13, 22, 4, 27.

Figure 8: A permutation σ of $1, 2, \ldots, 30$

How long does this algorithm take to find the *longest increasing subsequence (LIS)* on a sequence of n elements. Answer: To insert each element one needs to compare the new element to the last elements of all the rows; as the number of rows grows, this could even take $\approx n/2$ comparisons. In the worst case, we might need almost $\approx n^2$ steps in total. We made another observation.

> At every stage, the last element of the rows are sorted in increasing order.

We don't need to search through the rows sequentially to find the row to insert the new element in—we can use *binary search.* So if step 2 (a) is implemented using binary search, we can get everything done in $\approx n \log n$ steps.

Photo: Disha Kuzhively

Figure 9: We tried out the algorithm on the permutation σ with the numbers written on cards

Let us return to the theorem. Suppose the longest increasing sequence has length L. The elements of the sequence then reside in rows $1, 2, \ldots, L$. Say the longest of these rows has M elements. Then, $LM \geq n$. Do you see how the analysis of our algorithm actually yields a proof the theorem?

What about the approach we referred to above? How many sequences of 16 distinct elements are there where there is no monotone subsequence of 5 elements. Are there only a few that we most naturally constructed? Let us explore! Suppose we run the algorithm on such a sequence of length 16. What must we see in the end? Answer: four rows each with four elements in decreasing order from left to right; in the original sequence, each row as a decreasing subsequence. Now write these four rows one below another, packing the elements to the left. We get a 4×4 array of numbers. Since the original sequence had no monotone subsequence with five elements, this array has a very special structure. Clearly, the rows correspond to decreasing subsequences of the original sequence. The columns too are special. You should confirm (this will take some thought) that each column, when read from top to bottom, forms an increasing subsequence of the original sequence. Let us call an array where each row is decreasing from left to right and each column is increasing from top to bottom a *monotone array*.

7,5,9,6,15,2,4,13,11,16,1,14,3,12,8,10

7	5	2	1
9	6	4	3
15	13	11	8
16	14	12	10

Figure 10: A sequence obtained by rearranging $1, 2 \ldots, 10$, and the monotone 4x4 array obtained when we run the algorithm on this sequence

Explore, monotone arrays: Write numbers $1, 2, \ldots, 16$ in a 4×4 array. Sort each row in increasing order from left to right. Sort each column in increasing order from top to bottom. Do the rows continue to be sorted? Always? Why?

There is a marvellous formula to count the number of monotone 4×4 arrays (and actually much more, but we will discuss that a little later). Is it true that every such monotone array is the result of running the algorithm on a sequence with 16 elements with no monotone subsequence of length 5? Answer: yes, indeed given such an array, form a sequence by writing the rows one after another, top row first, then the second row, and so on. Why does this sequence of 16 elements have no monotone subsequence of 5 elements? Let us fix a 4×4 monotone array T. How many sequences of the numbers $1, 2, \ldots, 16$ when processed using our algorithm will give rise to T? Clearly the answer must be the same for every monotone array T. But what is this number?

Figure 11: What does the second 4×4 array look like?

The second array: Start with a sequence of numbers $1, 2, \ldots, 16$, (e.g., see fig. 10) where there is no monotone subsequence of length 5. Run the algorithm and construct the first tableau using the rows produced by the algorithm. Now, construct another 4×4 array R, whose cells are filled using the following rule (you may use the empty array in fig. 11). To determine the value in the cell ij, consider the number in this cell in the first tableau; say it is x. Suppose x appears in position ℓ of the original sequence, then we write ℓ in cell ij of the new tableau R. Note that each of the numbers $1, 2, \ldots, 16$, appears exactly once in the R. Stare at R carefully. What do you observe? If you are given T and R, can you reconstruct the original sequence from it?

The formula: The number of 4×4 monotone arrays is exactly

$$\frac{16!}{7 \cdot 6^2 \cdot 5^3 \cdot 4^4 \cdot 3^3 \cdot 2^2 \cdot 1} = 24,024\,.$$

In general, the number of $n \times n$ sorted tables is

$$\frac{n^2!}{\prod_{i=1}^{n} \prod_{j=1}^{n} (i + j - 1)}\,.$$

You are now ready to answer the following question!

> In how many ways can one arrange the numbers 1,2, ..., 16 so that the resulting subsequence has no monotone subsequence of 5 elements.

Where did the formula come from? Monotone arrays are closely related to *Young tableau*, which come in shapes that are not always square (e.g., see fig. 12; here the rows and columns are are both increasing). The expressions presented above are special cases of the wonderful *hook length formula* of Frame, Robinson and Thrall (1954) for counting the number of standard Young tableau of various shapes, even those that are not rectangular. At this website, you can enter the shape of the tableau and get the number of standard Young tableau of that shape:

https://www.integral-domain.org/lwilliams/Applets/Math/YoungDiagrams.php

1	3	4	7
2	8	5	
6	9	11	
10			

Figure 12: A standard Young tableau

27 April: Summary of the discussion on Explorations I, II and III

We reviewed the proof of the Erdős-Szekeres theorem and the solitaire sorting algorithm for finding the longest increasing subsequence. We observed that if a sequence of 16 elements has no monotone subsequence of 5 elements, then the resulting 4×4 array must be monotone. We did not carefully verify the claims about the structure of the array mentioned in the previous section. (Unfortunately, the example presented originally in fig. 10 had a error, so we could not use it; the error has been corrected now.) Everybody did notice that if we sort the rows of a 4×4 array and then sort its columns, then the rows remain sorted. We discussed why this happens. Starting from this, and following the discussion in the previous section, we hinted at how the formula for counting monotone arrays helps us count the sequences of 16 elements with no monotone subsequence of 5 elements.

We moved on to Exploration II. All groups after some time figured out that the robber can be caught if 8 cops are deployed, and somehow convinced themselves that 7 cops will not be enough. I waited for someone to formulate a crisp strategy for the robber to evade the cops perpetually. Eventually, to provoke an application of the pigeonhole principle, I asked them to complete the following sentence.

> If 7 cops are placed on an 8×8 grid graph, then
> some ... ...,
> and some ... ...

All groups realized that there always was a *safe* spot on the board for the robber, and the robber could quickly move to a new safe spot whenever a cop's move made her current spot unsafe. We did not bother to state the argument formally; everybody seemed comfortable with the notion of a *strategy*, both for the robber and for the cops.

Finally, we discussed Exploration III. An assignment with stretch 8 was found soon enough. Several attempts were made to obtain an assignment with stretch 7. Some of them looked promising to begin with, but none obtained a stretch better than 8. So we moved on to showing that 8 is the best possible. There are several arguments known that directly invoke the pigeonhole principle. I had hoped that the participants would spent some time at home with the problem in advance and perhaps develop some insights on their own. The problem is challenging and non-trivial, and it was not reasonable to develop a solution from scratch in the time we had left. Luckily, Exploration III has a connection to Exploration II. I asked the following.

> Show that if there is an assignment of numbers with stretch 7, then there is a strategy to chase down a robber on an 8×8 board with just seven cops.

I suggested how such a connection might be established by treating the numbers in the assignment as commands to the cops to occupy the various vertices of the 8×8 grid graph. Initially, the cops are placed in the vertices that corresponding to numbers $1, ..., 7$ in the assignment. At each step the cop occupying the vertex with the smallest number is marched to the a previously unvisited vertex with the smallest number; e.g., first, the vertex with number 1 will be vacated and the vertex numbered 8 will be occupied, then the vertex with number 2 will be vacated and the vertex with number 9 will be occupied. How can be sure that this sequence of moves by the cops will trap the robber eventually? That the assignment had a stretch of no more than 7 must have something to do with this. I only provided a sketch and left the participants to complete the argument themselves. Did anyone think this thorough completely later? I don't know. Some participants did appreciate the fact that a seemingly complicated question about 'static' assignments of numbers was settled by appealing to a question about the existence of a strategy in a game with 'dynamically' evolving positions and configurations. I hope everyone will feel encouraged to step back and look for unexpected connections between the various things that they discover in different contexts. I myself learnt many new things along the way; I am very grateful for having been asked to conduct these discussions on the pigeonhole principle.

(Last revised on 29 April 2024, by Jaikumar Radhakrishnan.)