SOLUTION

Problem of the Month (5/25)

May 2025 Problem of the Month. Submitted by Aria Dougherty, Tarleton State University, Stephenville, TX

Theorem. *Suppose that there are 100 passengers, numbered 1 through 100, boarding a plane that has exactly 100 seats. Passenger number 1 is the first to board the plane and randomly takes a seat. Passenger number two boards next and will sit in seat number 2 unless it is already occupied. In that case, passenger number 2 will take a random seat. This continues for all passengers through passenger number 99.*

Passenger 100 (the last to board) will take the only available seat left. They will be in either seat 1 or seat 100, with even probability.

Corollary. *The result above is the same if the 100 is replaced with any arbitrary $n \in \mathbb{N}$ and the final passenger to board is considered. This result extends to a result on bijective functions $f : \{1, \ldots, n\} \to \{1, \ldots, n\}$ which are mostly increasing (i.e. $x \leq f(x)$ for all $x \in \{1, \ldots, n\}$ except for one value, $f^{-1}(1)$); all such functions are such that $f(n) \in \{1, n\}$, and half of this finite number of functions (there are 2^{n-1} such functions) go to each possible outcome.*

Proof. We adopt the notation throughout this proof that P_n represents passenger n and S_n represents seat n (so P_3 is the 3^{rd} passenger and S_3 is the 3^{rd} seat). We will also adopt the terminology that S_n is P_n's "assigned" seat.

We first note that the only possible seats for P_{100} will be S_1 and S_{100} (i.e. $P(P_{100} \to S_n) = 0$ for any $n \in \mathbb{N}$ such that $1 < n < 100$). This is because seat n (for all such n) *cannot* be empty after P_n boards the plane, as they will always sit in their assigned seat if they are able to.

This means P_{100} only has 2 possible seats to end up in, the question remains as to how likely they are to end up in either seat. We will now show that P_{100} sits in S_1 or S_{100} (the only seats left) with even probability (in other words, $P(P_{100} \to S_1) = P(P_{100} \to S_{100}) = 0.5$). We begin with an observation about seating patterns.

Observation. *If P_1 sits in their assigned seat, then P_2 through P_{100} will also sit in their assigned seats.*

If P_1 sits in S_n for some $n > 1$, then P_2 through P_{n-1} will all sit in their assigned seats; in other words, the next passenger who will receive a choice of where to sit (i.e. have the ability to sit somewhere randomly) is P_n. When P_n sits, they will either sit in S_1 or P_k for some $k > n$. If they sit in S_1, then P_{n+1} through P_{100} must all sit in their assigned seats. If P_n sits in some S_k, then P_{n+1} through P_{k-1} will all sit in their assigned seats, and P_k gets the next choice.

This pattern continues until somebody sits in S_1 (potentially P_{100} if a prior passenger chooses to sit in S_{100}).

Generally speaking, the seating occurs in blocks which are determined entirely by choice of random seat for anyone whose seat was already chosen (starting with P_1 who gets the first choice) - this will produce a strictly increasing chain of natural numbers (called the "choice-chain") until someone (potentially P_{100}) sits in seat 1 (thus terminating the chain; we will denote this chain by $[n_1, n_2, \ldots, n_k, 1]$, representing $P_1 \to S_{n_1}$, $P_{n_1} \to S_{n_2}$, $\ldots$, $P_{n_k} \to S_1$). If the choice-chain terminates before 100 can be an element of it (necessarily the second-to-last number), then P_{100} sits where assigned; otherwise P_{100} sits in S_1 to terminate the chain. At this point, through this comparison one can conclude that the probabilities are 0.5 since the proposed problem is answered by the following question.

Related Question and Answer. *Consider $\mathscr{S}_n$ the set of all sequences of natural numbers no greater than n such that:*

1. *the sequence terminates with a 1, and*
2. *the (potentially empty) subsequence prior to 1 (i.e. everything except the last element) is a strictly increasing subsequence.*

For any $1 < k \le n$, what is the probability that an element $s \in \mathscr{S}_n$ chosen at random has k as an element of the sequence.

ANSWER: *The probability is 0.5. These sequences can be constructed through a binary tree starting with each natural number i from 2 to n and making the yes/no choice to put i in the sequence; the number of sequences constructible in this manner is equivalent to the size of the power set on $\{2, \ldots, n\}$, and it is well-known that the probability that a subset of $\{2, \ldots, n\}$ chosen at random will contain a specific element i is 0.5.*

By considering the answer above specifically for the probability that 100 is an element of a randomly chosen sequence in $\mathscr{S}_{100}$ (and recalling that there are only 2 possible seats for P_{100} to begin with), we see the claimed result: $P(P_{100} \to S_1) = P(P_{100} \to S_{100}) = 0.5$. □

We note that this solution can also be explored through expanding decision trees and making an argument inductively on P_1's initial choice (excepting that the set of cases where $P_1 \to S_1$ and $P_1 \to S_{100}$ are already established as being a result for $P_{100} \to S_{100}$ and $P_{100} \to S_1$, respectively). To make this argument, use the following inductive outline:

- **Base Case:** If $P_1 \to S_{99}$, then $P(P_{100} \to S_1) = P(P_{100} \to S_{100}) = 0.5$.
- **Inductive Hypothesis:** Fix $n < 99$. If $P_1 \to S_i$ for any $i > n$, then $P(P_{100} \to S_1) = P(P_{100} \to S_{100}) = 0.5$.
- **Inductive Step:** If $P_1 \to S_n$ (same n as in the inductive hypothesis), then $P(P_{100} \to S_1) = P(P_{100} \to S_{100}) = 0.5$.

Figure 1 demonstrates what these decision trees look like (terminating at S_1 or S_{100} since both qualify as an end to the chain - if $P_i \to S_1$ then $P_{100} \to S_{100}$; if $P_i \to S_{100}$ then $P_{100} \to S_1$), starting with P_1's choice of seat (in this instance S_{97}) and then proceeding by considering where the next passenger to sit randomly might choose to sit (in this instance P_{97}). It is easily seen (by exhaustion) that if $P_1 \to S_{97}$, then $P(P_{100} \to S_1) = P(P_{100} \to S_{100}) = 0.5$. However, the decision trees for the inductive hypothesis (note that the inductive hypothesis covers more than one choice) are visible in the middle of the the tree.

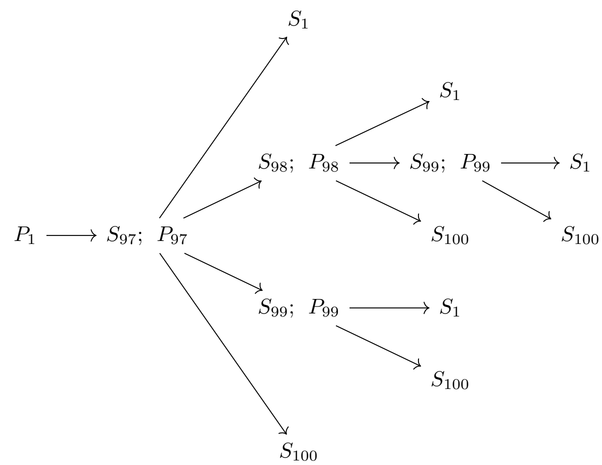

Figure 1: Choice tree for the initial choice $P_1 \to S_{97}$.

The nature of the argument is that if $P_1 \to S_n$, then there are $101-n$ choices of where P_n might sit. Consider calculating the probability that $P_{100} \to S_1$. Note that there is an even $\frac{1}{101-n}$ probability that P_n sits in any of these seats; one results in the unfavorable outcome that $P_{100} \to S_{100}$ (if $P_n \to S_1$), but the rest could still be favorable. If $P_n \to S_{100}$ then we're done. If $P_n \to S_i$ for some i between n and 100 (exclusively) then (by the inductive hypothesis) we know there is a probability of $\frac{1}{2}$ that $P_{100} \to S_1$. Hence, the net favorable probability that $P_{100} \to S_1$ can be calculated as follows:

$$\begin{aligned} P(P_{100} \to S_1) &= \frac{1}{101-n} \cdot 0 + \frac{(101-n)-2}{101-n} \cdot \frac{1}{2} + \frac{1}{101-n} \cdot 1 \\ &= 0 + \frac{99-n}{2(101-n)} + \frac{2}{2(101-n)} \\ &= \frac{1}{2}. \end{aligned}$$

This induction completes the proof in a more direct manner, omitting discussion of the related results.

Commentary on the May 2025 Problem of the Month by Dr. Ezra Halleck

The Lost Boarding Pass problem

The May 2025 Problem of the Month is often stated in a more general form. For instance, Peter Winkler on 2/7/22 had as his "mindbenders for the quarantined" problem (presented weekly via momath and email during the pandemic):

> "One hundred people line up to board a full jetliner, but the first has lost his boarding pass and takes a random seat instead. Each subsequent passenger takes his or her assigned seat if available, otherwise a random unoccupied seat. What is the probability that the last passenger to board finds his seat occupied?"

I will discuss how a solution to our version can be used to solve this more general one. I will also provide some references to solutions and a related problem. However, I begin by presenting some code that simulates the problem, first in R and then in python.

```
mixup <-function(n){
  empty=1:n
  selected=sample(n,1)
  empty=setdiff(empty, selected)
  for (i in 2:(n-1)){
    if (i %in% empty){
        empty=setdiff(empty, i)
    }
    else{
      randNum=sample(n-i+1,1)
      selected=empty[randNum]
      empty=setdiff(empty, selected)
      }
    }
  return(empty)#assigned seat for nth passenger
  }
```

```
n=100
N=10000
seats=replicate(N,mixup(n))
table(seats)
```

```
seats
   1  100
5030   4970
```

Note how no libraries have been used, meaning that all the functions are coming from base R. Using Rstudio on a 2015 MacBook pro, this example takes about 30 seconds to execute. In contrast, the following Python code executed on the same 2015 MacBook using Anaconda takes a second or 2.

```
import numpy as np
def mixup(n):
    empty = list(range(n)) #integers between 0 and n-1 inclusive
    selected = np.random.randint(n)# random integer between 0 and n-1 inclusive
    empty.remove(selected)
    for i in range(1, n-1):
        if i in empty: #passenger i sits in assigned seat
            empty.remove(i)
        else: #since passenger i's assigned seat is not available,
             #i randomly selects a seat among those still empty
            rand_int = np.random.randint(n-i)
            selected = empty[rand_int]
            empty.remove(selected)
    return empty[0] # assigned seat for nth passenger
                #(actually (n-1)st since python indexes from 0)
```

```
n = 100
N = 10000
```

```
seats = np.array([mixup(n) for _ in range(N)])+1
#adding 1 switches from 0 to (n-1) indexing to the standard 1 to n indexing

def make_freq_table(a_list):
   outcomes=list(set(a_list))
   freqs=[]
   for i in outcomes:
      tally=0
      for j in a_list:
         if j==i:
            tally+=1
      freqs.append(tally)
   return [outcomes,freqs]

my_freq_table=make_freq_table(seats)
for sublist in my_freq_table:
   print(*sublist)
```

```
1 100
5032 4968
```

To make use of one of our solutions to solve the more general version of the problem, relabel the seat assigned to the i^{th} person who enters as seat “i”. Once the problem is solved, then reverse the relabeling to answer the question in terms of the original seat assignments. So, we conclude that the last person who enters has a 50% chance of sitting in the first passenger’s assigned seat and a 50% chance of sitting in their own assigned seat.

According to https://www.cut-the-knot.org/Probability/LostPass.shtml#, the problem and the first solution they present appears in P. Winkler's *Mathematical Puzzles: A Connoisseur's Collection* (A K Peters, 2004, pp. 35-37).

The site provides 3 additional solutions by Bollobas, Saletic and Papadopulos, respectively.

Our version was looked at by Jim Dotten in 2013 and the site suggests a proof by induction.

Cut the knot also looks at the related problem of determining the average number of happy (non-displaced passengers): https://www.cut-the-knot.org/Probability/NumberOfHappyPassengers.shtml

The problem was asked by Alexander Pipersky. Two solutions are provided, the first by Konstantin Knop.