# Riddler June 24, 2022:
# Goat parking functions

Mark Girard

June 27, 2022

## Riddler classic

From Quoc Tran comes a caprine conundrum:

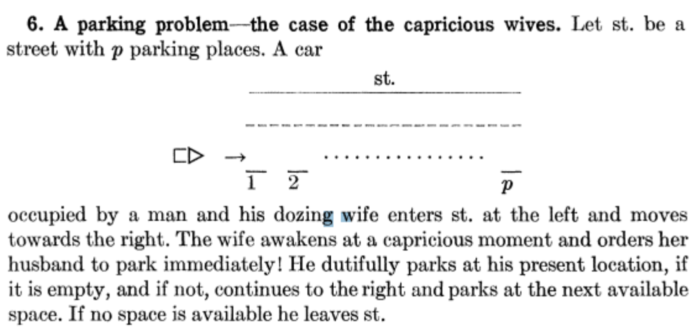
**Question 1.** *A goat tower has 10 floors, each of which can accommodate a single goat. Ten goats approach the tower, and each goat has its own (random) preference of floor. Multiple goats can prefer the same floor.*

*One by one, each goat walks up the tower to its preferred room. If the floor is empty, the goat will make itself at home. But if the floor is already occupied by another goat, then it will keep going up until it finds the next empty floor, which it will occupy. But if it does not find any empty floors, the goat will be stuck on the roof of the tower.*

*What is the probability that all 10 goats will have their own floor, meaning no goat is left stranded on the roof of the tower?*

## Solution

This Riddler is a take on the classic *parking function* problem—a well studied combinitorial problem for which there seems to be much literature. The problem seems to have first been studied by Konheim and Weiss [1]. It's worth taking a look at the original discussion of the problem (despite the outdated heteronormative overtones) to see why this is called the "parking function" problem. A snippet of the original paper is included here in Figure 1.


**6. A parking problem—the case of the capricious wives.** Let st. be a street with $p$ parking places. A car



occupied by a man and his dozing wife enters st. at the left and moves towards the right. The wife awakens at a capricious moment and orders her husband to park immediately! He dutifully parks at his present location, if it is empty, and if not, continues to the right and parks at the next available space. If no space is available he leaves st.

Figure 1: A snip of the original paper by Konheim and Weiss [1]

## Counting parking functions

We can more rigorously define the problem as follows. Essentially, we are tasked here with computing the number of *parking functions*, which we can define below.

**Definition 2.** A *parking function* on $n$ elements is a function $f : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ such that the values $f(1), f(2), \ldots, f(n)$ when sorted in non-decreasing order

$$f(k_1) \leq f(k_2) \leq \cdots \leq f(k_n)$$

satisfy $f(k_j) \leq j$ for each $j \in \{1, 2, \ldots, n\}$.

To see why the concept of a parking function captures the essence of the problem, consider an arbitrary function $f : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ such that the value of $f(k)$ indicates the preferred floor of the $k^{\text{th}}$ goat. Such a function is a valid preference function (yielding a parking arrangement of goats where no goat is stuck on the roof) if and only if, for each $j$ the $j^{\text{th}}$ highest preference $f(k_j)$ is not higher than the $j^{\text{th}}$ floor.

There are a total of $n^n$ functions on $\{1, 2, \ldots, n\}$, but how many of them are parking functions? If we let $P_n$ denote the number of parking functions, surprisingly it turns out that

$$P_n = (n+1)^{n-1}$$

for each $n$ (and we can define $P_0 = 1$). This was first proved by Konheim and Weiss in their original paper using recursion and generating functions—a rather complicated method that I won't reconstruct here. Instead I'll examine the problem in two ways. First by constructing a recursive sequence that yields the number of parking functions, and second by a combinatorial proof that directly shows that the number of parking functions is $(n+1)^{n-1}$.

## A recursive sequence for $P_n$

We can recursively generate a sequence that counts the number of valid parking functions for $n+1$ goats as follows. We first count the number of ways that the first $n$ goats can have their preferences in $n+1$ floors such that every goat gets a spot. After the first $n$ goats have settled, there will be one empty floor. For a given $k \in \{1, 2, \ldots, n+1\}$, how many ways can the first $n$ goats prefer floors such that the $(k+1)^{\text{th}}$ floor is empty after they all settle? The answer to this is

$$\binom{n}{k} P_k P_{n-k},$$

because we must first choose which of the $k$ goats go in the first $k$ spots, then the two groups of $k$ and $n-k$ goats must respectively settle in their partitions (i.e., in the bottom $k$ floors or the upper $n-k$ floors). Finally, if the $(k+1)^{\text{th}}$ floor is empty, there are only $k+1$ possible floors that the final $(n+1)^{\text{th}}$ goat can prefer so that it ends up settling in the $(k+1)^{\text{th}}$ floor. The number of parking functions for $n+1$ goats can therefore be computed recursively as

$$P_{n+1} = \sum_{k=0}^{n} \binom{n}{k} (k+1) P_k P_{n-k}$$

where we define $P_0 = 1$.

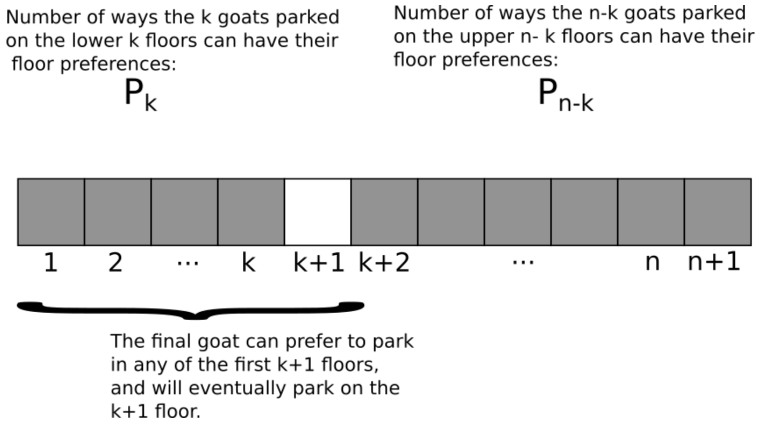



Figure 2: How many ways can $n+1$ goats prefer floors? After the first $n$ goats park in the tower, there must be exactly one floor empty. For each $k \in \{1, 2, \dots n+1\}$, there are $\binom{n}{k}$ ways to split the goats into groups that settle on the lower $k$ floors and upper $n-k$ floors respectively. There are then only $k+1$ floors that the last goat can prefer, otherwise it's condemned to the roof...

It's worth mentioning that this is the same recursive sequence considered by Konheim and Weiss in their generating function method. In particular, they considered the function

$$g(x) = \sum_{n=0}^{\infty} \frac{P_n}{n!} x^n$$

and showed that this function must satisfy $g(x) = xe^{g(x)}$, which in turn is satisfied uniquely by

$$g(x) = \sum_{n=0}^{\infty} \frac{(n+1)^{n-1}}{n!} x^n.$$

## A combinitorial proof

Here we show a more direct combinatorial proof that $P_n = (n+1)^{n-1}$. This proof is due to Pollak and was first published by Foata and Riordin [2].

To construct all parking functions $f : \{1, 2, \dots, n\} \to \{1, 2, \dots, n\}$, let's first consider a slight variant of the goats in the tower game.

Consider the same scenario as earlier with the goats (where there are $n$ goats who have preferences for one of the $n$ different floors in the tower), but now label the roof as the $(n+1)^{\text{th}}$ floor which is an allowable preference and an allowable place for up to one goat to stand. There are now $(n+1)^n$ ways in which the goats can have preferences in this modified tower. Now, however, if the roof is full and another goat finds itself on the roof, that goat instead goes back to the groud floor (floor 1) and tries again. After each goat has situated itself on a floor of the new tower, there is exactly one floor empty. By symmetry, there are equally many ways for the goat's preferences to leave any of the $n+1$ floors empty. The preference function on the modified tower is a parking function exactly when the $(n+1)^{\text{th}}$ is empty (and otherwise it would not be a valid parking function). Hence, of the $(n+1)^n$ functions from $\{1, 2, \dots, n\}$ to $\{1, 2, \dots, n, n+1\}$, only $\frac{1}{n+1}$ of them yield a valid parking function. Moreover, every possible parking function can be constructed this way. It follows that

$$P_n = \frac{(n+1)^n}{n+1} = (n+1)^{n-1}.$$

## A final note

The original problem statement asks us to find the probability of the goats all ending up on a floor by themselves, given that each goat chooses its preference randomly. This probability can therefore be expressed as

$$\frac{(n+1)^{n-1}}{n^n}.$$

For $n = 10$, this is equal to

$$\frac{11^9}{10^{10}} = \frac{2357947691}{10000000000} = 0.2357947691,$$

or roughly 23.6%.

## References


[1] Alan G. Konheim and Benjamin Weiss. “An occupancy discipline and applications.” *SIAM Journal on Applied Mathematics* 14.6 (1966): 1266-1274.

[2] Dominique Foata and John Riordan. “Mappings of acyclic and parking functions.” *Aequationes Mathematicae* 10.1 (1974): 10-22.