Clash Royale Heroes Analysis

Stephen Xu

December 18, 2025

1 Introduction

Clash Royale is a mobile strategy game enjoyed by millions of players around the world, including myself. In December 2025, Clash Royale released an update featuring a brand-new gameplay feature: heroic cards. While heroes have introduced exciting and unique mechanics into the game, unlocking them has proven to be a laborious task.

Players unlock heroes by collecting 200 hero fragments, which automatically summons a heroic card. The trouble is, summoning a hero does not necessarily guarantee a *new* card. Rather, players run the risk of summoning a duplicate hero, which they are meagerly compensated for. Heroes are quite strong in Clash Royale, so players want to obtain as many unique heroes as possible. Therefore, many Clash Royale players are extremely frustrated by this process, calling it yet another cash grab by Supercell. Although I personally resonate with these sentiments, this at least opens the door to some *very* interesting mathematics.

In this document, we will answer the following questions:

- How likely is it to obtain k unique heroes after m hero summons, given there are n heroes currently in the game?
- How are these probabilities distributed as k goes from 1 to $\min(m, n)$?
- Is the hero unlock system really that bad?

I assume:

- Hero summons are independent of all previous summons and give equal probability of obtaining any hero in the game, regardless of which ones are currently unlocked.
- The reader has a basic understanding of discrete math and counting concepts.

Skip to section 5 (conclusions) if you are not interested in the math. Enjoy!

2 Surjections $[m] \to [n]$

Define $[m] = \{1, 2, \cdots, m\}$ for any positive integer m. Before we proceed with tackling the main problem, we first discuss an important background problem. Given positive integers m, n with $m \geq n$, we aim to find the number of surjective mappings from $[m]$ to $[n]$, which we denote as $U(m, n)$. Recall that a surjection $f : A \to B$ is a function where for all $b \in B$, there exists some $a \in A$ such that $f(a) = b$.

It is easier to find the number of mappings that are *not* surjections. Without any restrictions, there are n^m total mappings from $[m]$ to $[n]$. Next, define $A_1, A_2, \cdots, A_n$ where each A_k is the set of all mappings that do not include element k. All non-surjective mappings must therefore belong to at least one such A_k, so the number of surjective mappings equals to

$$n^m - |A_1 \cup A_2 \cup \cdots \cup A_n|. \tag{1}$$

By the Principle of Inclusion-Exclusion (PIE) and symmetry, we have

$$\begin{aligned} |A_1 \cup A_2 \cdots \cup A_n| &= (|A_1| + \cdots + |A_n|) - (|A_1 \cap A_2| + \cdots |A_{n-1} \cap A_n|) + \cdots \\ &= \binom{n}{1}|A_1| - \binom{n}{2}|A_1 \cap A_2| + \cdots + (-1)^n \binom{n}{n-1}|A_1 \cap A_2 \cap \cdots A_{n-1}|. \end{aligned}$$

Here, I did not include the last term $|A_1 \cap \cdots \cap A_n|$, as every mapping to $[n]$ must hit at least one element. For any $1 \leq k \leq n-1$, observe

$$|A_1 \cap \cdots \cap A_k| = (n-k)^m,$$

since each of the m elements in $[m]$ have $n - k$ choices to map to in $[n]$. Substituting this result into our PIE expression and back into (1) gives us the following expression for the total surjections:

$$U(m, n) = n^m - \binom{n}{1}(n-1)^m + \binom{n}{2}(n-2)^m + \cdots + (-1)^{n-1}\binom{n}{n-1}1^m.$$

This nicely collapses into the summation below:

$$\boxed{U(m, n) = \sum_{j=0}^{n}(-1)^j (n-j)^m \binom{n}{j}}$$

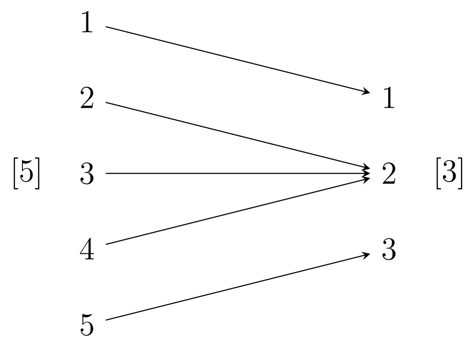
3 Stirling Numbers of the Second Kind

The problem above allows us to transition nicely to discussing Stirling numbers of the second kind. I never heard of these numbers prior to this hero analysis, so I hope you'll learn a little from this too. Formally, define $S(n, k)$ as the number of ways to partition $[n]$ into k non-empty subsets/groups, where n, k are non-negative integers. For example, if $n = 5$ and $k = 3$, one possibility is:

$$\{1, 2, 3, 4, 5\} \to \{\{1\}, \{2, 3, 4\}, \{5\}\}.$$

Clearly, if $n < k$, $S(n, k) = 0$. We also take $S(0, 0) = 1$, as there is 1 way to split the empty set into zero groups.

But this problem is nearly identical to the $[n] \to [k]$ surjection problem discussed earlier! *One* equivalent representation of the example above would be:



However, the main difference in the Stirling number problem is that the groups are unordered - we do not care about how they are labeled. So, if we assign a label to the k groups in the Stirling number problem, we precisely get the surjection problem. Therefore,

$$k! \cdot S(n,k) = U(n,k) \implies \boxed{S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^j (k-j)^n \binom{k}{j}} \tag{2}$$

Stirling numbers can also be defined recursively. Consider where element n is grouped into. There are two cases:

- If n is assigned to a singleton set/group, there are $S(n-1,k-1)$ ways to group the remaining $n-1$ elements into $k-1$ groups.
- Otherwise, n joins an existing group. In this case, we first partition $\{1, 2, \cdots, n-1\}$ into k groups in $S(n-1,k)$ ways. Then, n can join any of the k existing groups.

All-together, the recursive definition for $S(n,k)$ is:

$$\boxed{S(n,k) = S(n-1,k-1) + k \cdot S(n-1,k)}$$

with base cases

$$S(0,0) = 1, \quad S(n,0) = 0 \text{ for } n \geq 1, \quad S(0,k) = 0 \text{ for } k \geq 1.$$

As a side note, recall binomial coefficients abide by

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

with $\binom{0}{0} = 1$. These definitions are so close, despite the underlying problems being different!

In my opinion, the recursive definition is a smidge easier to work with than equation (2). In particular, it allows us to write a bottom-up dynamic programming algorithm to compute an entire batch at a time:

Algorithm 1: Compute Stirling Numbers of the Second Kind

```
Input: Integer x ≥ 1
Output: List L where L[y] = S(x, y) for 0 ≤ y ≤ x
Initialize a (x + 1) × (x + 1) table dp with all entries 0;
Set dp[0][0] ← 1;
for i ← 1 to x do
    for j ← 1 to i do
        dp[i][j] ← dp[i − 1][j − 1] + j · dp[i − 1][j];
return dp[x];
```

This algorithm appears as the *stirling* function in *main.py*.

4 Unique Hero Distribution Analysis

We are now (finally) ready to analyze the unique hero problem.

If there are n heroes in the game and we perform m hero summons, what is the probability distribution of the number of unique heroes obtained? Equivalently, for any $1 \leq k \leq \min(m, n)$, what is $P(\text{exactly } k \text{ unique heroes})$? (Observe that we use $\min(m, n)$ as the upper bound for unique heroes - if $m < n$, we obtain up to m unique and if $m > n$, we get up to n unique)

First of all, performing m hero summons with n heroes in the game results in n^m possible hero sequences (Giant, Mini P, Giant, Musk, Knight for example). If k unique heroes are obtained in this process, then we are effectively doing a surjection from $[m]$ to the set of k heroes. There are $\binom{n}{k}$ ways to choose the k unique heroes and $U(m, k)$ ways to map the surjection. From (2), this means

$$P(k \text{ unique heroes}) = \frac{\binom{n}{k} U(m, k)}{n^m} = \frac{\binom{n}{k} k! \cdot S(m, k)}{n^m}.$$

Using the identity $\binom{n}{k} k! = (n)_k$, where $(n)_k$ is the falling product $n(n-1)\cdots(n-k+1)$, the probability is:

$$\boxed{P(k \text{ unique heroes}) = \frac{(n)_k \cdot S(m, k)}{n^m}}$$

Summing over all possible k gives us another neat identity:

$$\sum_{k=1}^{\min(m,n)} (n)_k \cdot S(m, k) = n^m$$

which is a confirmation that the probabilities should sum to 1. We can compute the $S(m, k)$'s all at once via Algorithm 1.

Let's also discuss the **expected** number of unique heroes one would obtain by summoning heroes m times with n heroes currently in the game. Refer to this as $\mathbb{E}(m, n)$. A straightforward but tedious way to compute this expectation is via the probability distribution:

$$\mathbb{E}(m, n) = \sum_{k=1}^{\min(m,n)} \frac{(n)_k \cdot S(m, k)}{n^m} \cdot k.$$

But there is a much cleaner closed-form expression for $\mathbb{E}(m, n)$. Let I_i represent the indicator variable (zero or one) that hero i appears at some point in the summoning. Then, if X is the random variable used to the represent the number of unique heroes obtained, we have

$$X = \sum_{i=1}^{n} I_i.$$

By linearity of expectation,

$$\mathbb{E}[X] = \mathbb{E}(m, n) = \sum_{i=1}^{n} \mathbb{E}[I_i]. \tag{3}$$

The expected value of each I_i is simply the probability hero i is unlocked at some point. This is the complement of the probability hero i is never unlocked. Each of the m summons have a $1 - 1/n$ probability to miss hero i, so

$$\mathbb{E}[I_i] = 1 - \left(1 - \frac{1}{n}\right)^m.$$

Substituting into (3) gives

$$\boxed{\mathbb{E}(m, n) = n\left(1 - \left(1 - \frac{1}{n}\right)^m\right)}$$

I think it's quite interesting that the nasty summation we had for $\mathbb{E}(m, n)$ on the previous page simplifies into this. This concludes the math! See the next page for some plots and general conclusions.

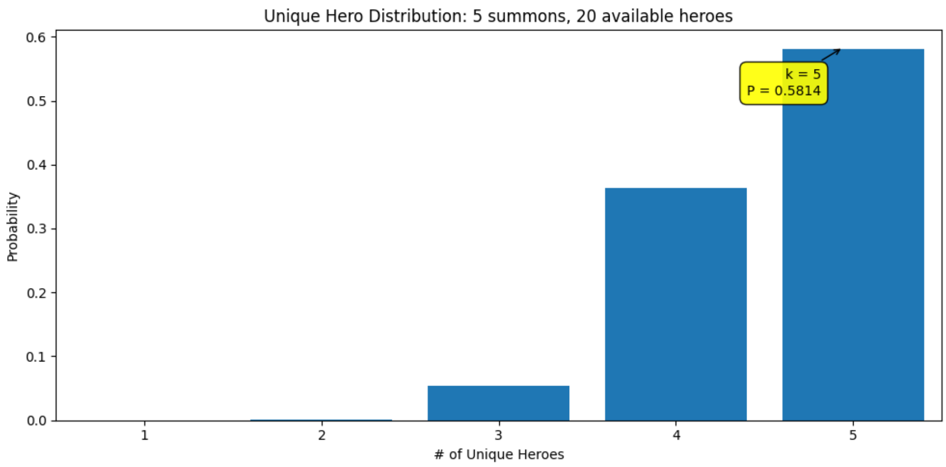
5 Conclusions

Through the GitHub repository, you are more than welcome to tweak the the number of heroes, n, and the number of summons, s to see how the probability distribution changes. You may also wish to use the logarithmic scale to see everything clearly. Some general observations:

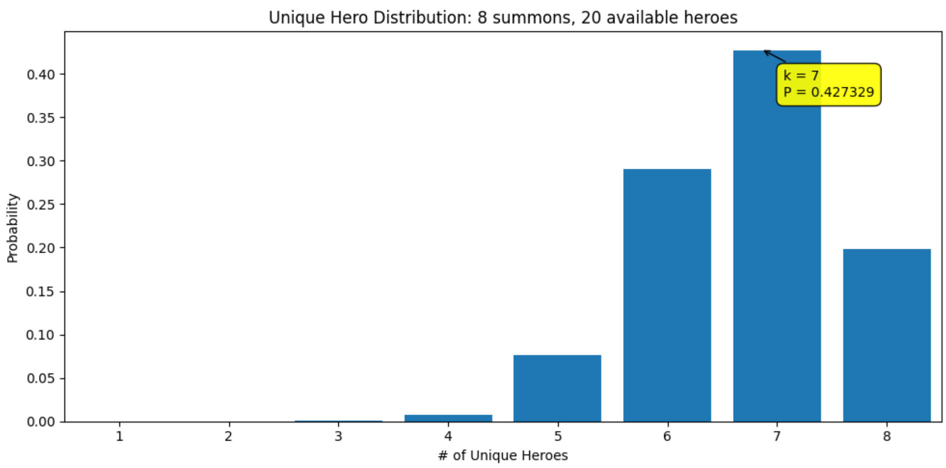
- If $s \gg n$, the distribution becomes more and more left-skewed. The more times you summon heroes, the more likely it is to obtain most or all of them.
- If $s \approx n$, the distribution is roughly normal with peak around $k = \frac{2}{3}s$. For example, with 30 summons and 30 available heroes, you can expect to obtain 19 or 20 unique heroes.
- If $s \ll n$, you will expect to get close to s unique heroes. This makes sense, as there isn't much chance for overlap.

Let's dive into a concrete example. As many of us are aware, the level 16 cap will remain in Ranked for about 6 more months. Once this cap is removed, Ranked will undoubtedly be crawling with maxed players running 2 heroes. If you wish to stay competitive in ranked, it is paramount that you unlock as many of the (likely broken) heroes as possible. I estimate there will be around $n = 20$ heroes in the game by then. Varying s, we get:

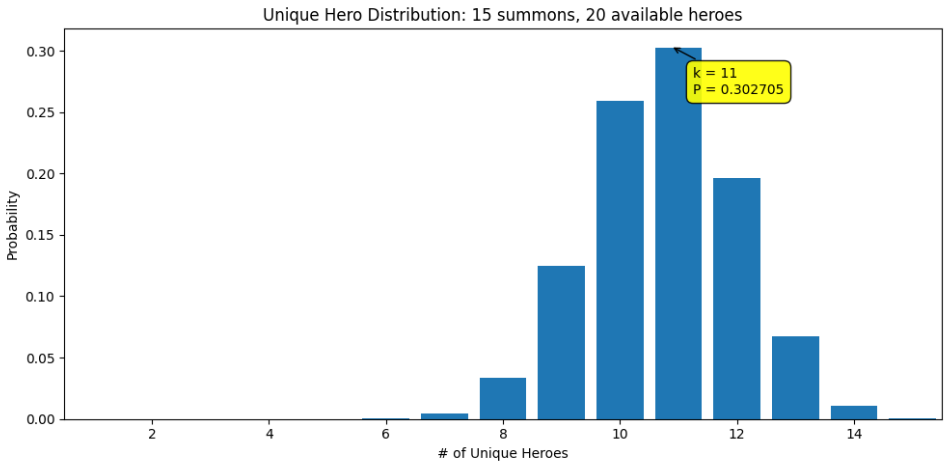
- $s = 5$:



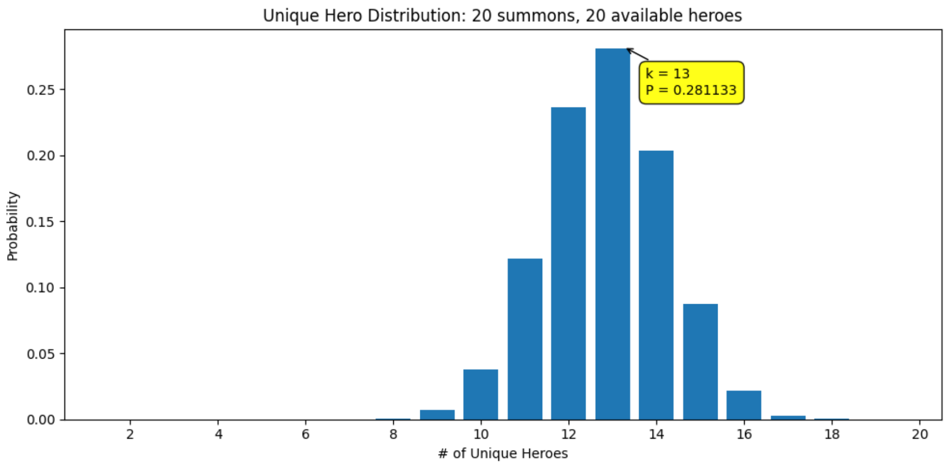
- $s = 8$:



- $s = 15$:



- $s = 20$:



It makes sense that when s is relatively small, you will not encounter too many duplicate heroes. Although when $s = 8$, there is only a 20% chance to unlock 8 out of 8 unique heroes. However, the plots for $s = 15$ and $s = 20$ are rather discouraging. As you obtain more and more heroes, it gets progressively more difficult to unlock new ones. If you manage to summon heroes twenty times, you can only expect to obtain about 13 heroes. In fact, you would need to summon heroes **37** times to expect to get 17 out of 20.

Using the expectation formula in the previous section, you need to summon n times to unlock 63% of the heroes, $1.6n$ to unlock 80%, and close to $3n$ to expect 95%. These are ridiculously high numbers! Of course, there is a caveat that you summon heroes as they are being added. The plots above assume you summon s heroes while there are 20 in the game, rather than summoning one at a time over the course of many months. It's tricky to account for the gradual addition of heroes into the game, so perhaps this is worth a follow-up. But regardless, the current system makes it very difficult to unlock all heroes without spending money.

Free-to-play players realistically only expect to summon a couple of heroes per month *at best*. I conclude it will be **near impossible** for them to unlock all heroes without a ridiculous amount of luck. Thank you for reading.