Code

```python
mod = 1_000_000_007

@cache
def dp(n: int, k:int):
    if n == k:
        return 1
    if n <= 0 or k <= 0:
        return 0
    return (dp(n - 1, k - 1) + (n - 1) * dp(n - 1, k)) % mod

class Solution:
    def rearrangeSticks(self, n: int, k: int) -> int:
        return dp(n, k)
```

What It Does

For a set $A = \{1, \cdots, n-1\}$, if new element n comes in, $V_k(A^{'} := A \cup \{n\}) = V_{k-1}(A) + (n-1)V_k(A)$

1. Place n at rear

If we do so, since n is larger than any element in A, we would get $V_{k-1}(A)$ arrangements with k visible sitcks.

2. Place n where it doesn't affect the number of visible elements

We already have k visibles in $V_k(A)$ arrangements so we shouldn't affect them. To do so, we have no choice but to place n just in front of the last visible element, $n-1$.

Or we could simply think of a set $B = \{1, \cdots, n-2, n\}$ instead of A since $V_k(B) = V_k(A)$. Also, for any element $m < n$ and $X = \{1, 2, \cdots, m-1, m+1, \cdots, n-1, n\}$, $V_k(X) = V_k(A)$. So we have $n-1$ cases in total.