This generalizes to 5 steps.

Ex: How many \( n \)-bit strings are there? \( 2^n \)

\[
\underbrace{2 \cdot 2 \cdot 2 \cdots 2}_n
\]

Ex: How many subsets does an \( n \)-element set have? \( \{a_1, a_2, \ldots, a_n\} \) \( 2^n \)

For each \( a_i \), choose to put it in the subset or leave it out.

Ex: How many 4-character passwords, where each is either a lowercase letter or a digit, and the first has to be a letter? \( 26 \cdot 36^3 \)

Ex: Same, but no restriction on first character, and no character can be repeated? \( 36 \cdot 35 \cdot 34 \cdot 33 \)

Permutations:

How many sequence sequences using \( \{1, 2, 3\} \) are there? In which each is used exactly once?

\[
3 \cdot 2 \cdot 1 = 6
\]

How many using \( \{1, 2, \ldots, n\} \)?

\[
n(n-1)(n-2) \cdots 3 \cdot 2 \cdot 1 = n! \quad 0! = 1.
\]

\text{k-permutations: How many sequences of length k use the characters \( \{1, 2, \ldots, n\} \) each either 0 or 1 time?}

\[
P(n, k) = n(n-1)(n-2)\cdots(n-k+1) = \frac{n!}{(n-k)!}
\]

Back to the password example:

\[
P(36, 4) = \frac{36!}{32!}
\]