2928. Distribute Candies Among Children I

You are given two positive integers \( n \) and \( \text{limit} \).

Return the total number of ways to distribute \( n \) candies among \( 3 \) children such that no child gets more than \( \text{limit} \) candies.

Example 1:

Input: \( n = 5 \), \( \text{limit} = 2 \)
Output: \( 3 \)
Explanation: There are 3 ways to distribute 5 candies such that no child gets more than 2 candies: \((1, 2, 2)\), \((2, 1, 2)\) and \((2, 2, 1)\).

Example 2:

Input: \( n = 3 \), \( \text{limit} = 3 \)
Output: \( 10 \)
Explanation: There are 10 ways to distribute 3 candies such that no child gets more than 3 candies: \((0, 0, 3)\), \((0, 1, 2)\), \((0, 2, 1)\), \((0, 3, 0)\), \((1, 0, 2)\), \((1, 1, 1)\), \((1, 2, 0)\), \((2, 0, 1)\), \((2, 1, 0)\) and \((3, 0, 0)\).

Constraints:
- \( 1 \leq n \leq 50 \)
- \( 1 \leq \text{limit} \leq 50 \)

hint: use triple-nested for-loop

for i in range(limit+1):
    ... check i+j+k = n

6 \leq i \leq \text{limit}
Possible solution to the candy distribution problem.

```python
1 def distribute_candies(n: int, limit: int) -> int:
2     """Counts the number of ways to distribute 'n' candies
3     to 3 people so that each person has a maximum
4     of 'limit' candies."""
5     n_ways = 0
6     for i in range(limit + 1):
7         for j in range(limit + 1):
8             for k in range(limit + 1):
9                 if i + j + k == n:
10                     n_ways += 1
11                     # print(i, j, k)
12     return n_ways
```

Try $n = \ell = 50$: how much time does this take?
What about $n = \ell = 100$? $n = \ell = 250$? $n = \ell = 500$?
Our problem for today: in how many ways can you distribute $n$ candies to 3 people if every person is limited to $\ell$ candies.

Breaking the problem down:
1. In how many ways can you distribute $n$ candies to $k = 3$ people?
2. Then subtract ways that go over the limit $\ell$.

\[ i + j + k = n \]

How many $+$'s for $k$ people? $k-1$

\begin{align*}
\text{stars} & \quad p \\
\text{dividers (bars)} & \quad \ell-1
\end{align*}

in how many ways can we pick location of $(k-1)$ $+$'s (bars) from a total $(n+k-1)$ slots \( \binom{n+k-1}{k-1} \) "stars and bars"
Our problem for today: in how many ways can you distribute \( n \) candies to 3 people if every person is limited to \( l \) candies.

**Breaking the problem down:**

1. In how many ways can you distribute \( n \) candies to \( k = 3 \) people?
2. Then subtract ways that go over the limit \( l \).

we have total ways = \( \text{sb}(n, k) \)

\[ 1) \text{ suppose we give } (l+1) \text{ candies to 1 person; } n - (l+1) \text{ left} \]

\[ \text{how many ways to pick 1 person? } \binom{3}{1} = 3 \]

\[ -3 \times \text{sb}(n-(l+1), 3) \]

\[ 2) \text{ add back in } \# \text{ ways in which 2 people over limit; } n - 2(l+1) \text{ left} \]

\[ \binom{3}{2} = 3 \]

\[ +3 \times \text{sb}(n-2(l+1), 3) \]

\[ 3) \text{ subtract } \# \text{ ways in which 3 people over limit; } n - 3(l+1) \text{ left} \]

\[ \binom{3}{3} = 1 \]

\[ -\text{sb}(n-3(l+1), 3) \]

\[ \text{sb}(n, k): \]

check if \( n < 0 \) \[ \text{return } 0 \]

\[ \text{math.comb}(n+k-1, k-1) \]
Possible solution to the candy distribution problem.

```python
1 def stars_and_bars(n: int, k: int) -> int:
2     """Returns the number of ways 'n' objects can
3     be distributed to 'k' slots."""
4     if n < 0:
5         return 0
6     return math.comb(n + k - 1, k - 1)
7
8 def distribute_candies(n: int, limit: int) -> int:
9     """Counts the number of ways to distribute 'n' candies
10     to 3 people so that each person has a max of limit"""
11     n_ways = stars_and_bars(n, 3)
12     n_ways -= 3 * stars_and_bars(n - (limit + 1), 3)
13     n_ways += 3 * stars_and_bars(n - 2 * (limit + 1), 3)
14     n_ways -= stars_and_bars(n - 3 * (limit + 1), 3)
15     return n_ways
```

Try $n = l = 50$: how much time does this take?
What about $n = l = 100$? $n = l = 250$? $n = l = 500$?