

Let  $n$  denote the number of steps and let  $m$  denote the array length.

1. DP.  $O(n \cdot \min(n, m))$ .

2. If we cannot stay in the same place: DP, the answer is

$$\binom{n}{n/2} - \sum_{i \geq 1} \left( \binom{n}{n/2 - (i-1)m - (2i-1)} - \binom{n}{n/2 + im + 2i} \right) - \sum_{i \geq 1} \left( \binom{n}{n/2 + im + (2i-1)} - \binom{n}{n/2 - im - 2i} \right).$$

$O(n)$ . <https://www.zhihu.com/question/346654767/answer/1110765408>

Now we can stay in the same place, and the answer is

$$\sum_{n' \leq n, n' \text{ even}} \binom{n}{n'} \left[ \binom{n'}{n'/2} - \sum_{i \geq 1} \left( \binom{n'}{n'/2 - (i-1)(m-1) - (2i-1)} - \binom{n'}{n'/2 + i(m-1) + 2i} \right) - \sum_{i \geq 1} \left( \binom{n'}{n'/2 + i(m-1) + (2i-1)} - \binom{n'}{n'/2 - i(m-1) - 2i} \right) \right].$$

We can compute the solution in  $O(n \cdot \frac{n}{m})$  time, by preprocessing factorial mod  $p$  and its inverse, then each binomial can be computed in  $O(1)$  time.

If  $m \leq \sqrt{n}$ , use the first algorithm, otherwise use the second algorithm.  $O(n\sqrt{n})$ .

3. Another way to derive: <https://hyper-meta.blogspot.com/2023/10/the-long-journey-from-on2-to-on15.html>.

4.  $\tilde{O}(\sqrt{n})$  using polynomials or linear recursions. see my article <https://zhuanlan.zhihu.com/p/344219746>.

#### Number of Ways to Stay in the Same Place After Some Steps

##### Submission Detail

31 / 31 test cases passed.

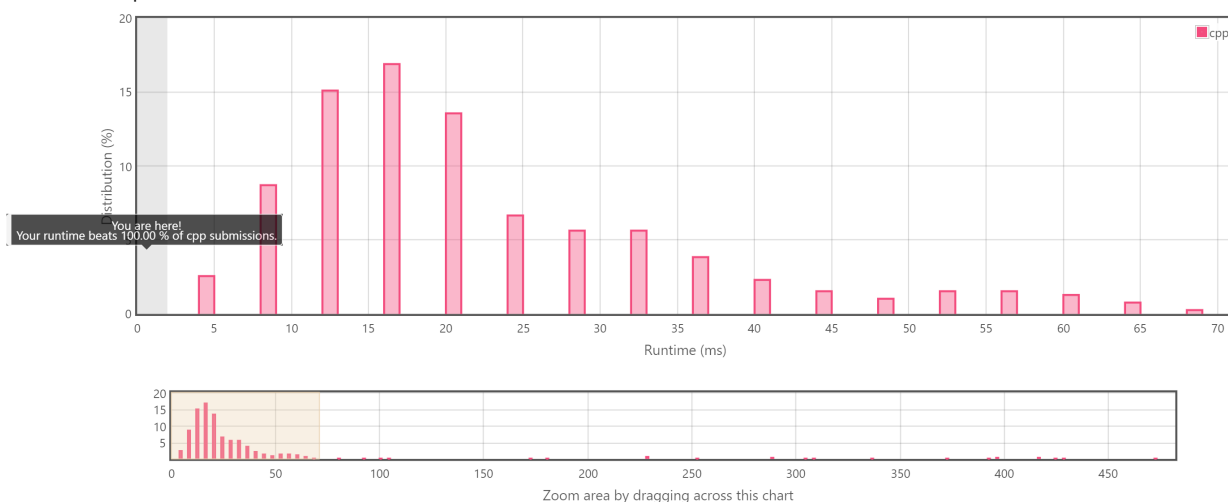
Runtime: 0 ms

Memory Usage: 6 MB

Status: Accepted

Submitted: 0 minutes ago

##### Accepted Solutions Runtime Distribution



## References