# *The Ball Problem - Editorial*

## Difficulty:

Medium - Hard

## Prerequisites:

Combinatorics
Divide and Conquer - (Tutorial)
Fast Fourier Transform - (Tutorial)
Generating Function - (Tutorial)

## Problem in Brief:

Given an Array A, and a number K, count the different possible ways to choose K elements from A. Two ways are considered different if count of elements is different in them for any element.

## Editorial:

We can convert this problem to a combinatorial one. Notice that the equation $X_1 + X_2 + \ldots.. X_{200000} = N$ where $X_i$ is the number of balls with color i that we choose is the solution to our problem.

If we could choose as many balls of any color that we wanted then the solution would be C(N + K - 1, K) where C(i, j) is the binomial coefficient which is the number of ways to choose j objects from the given i objects.
But we have a restriction:
Let $L_i$ be the count of ball colored i in the given N balls.
The restriction $X_i <= L_i$ should be followed for all i colors.

To overcome this restriction we will use Generating Functions.

# *The Ball Problem - Editorial*

The solution to our problem is the coefficient of $X^K$ in the following polynomial:
$(X^0 + X^1 + X^2 + X^3 + .... X^{L-1}) * (X^0 + X^1 + X^2 + X^3 + .... X^{L-2}) * (X^0 + X^1 + X^2 + X^3 + .... X^{L-3}) * ... (X^0 + X^1 + X^2 + X^3 + .... X^{L-200000})$

Now to actually find this polynomial we will use Fast Fourier Transform. Notice that if we multiply the polynomials 1 by 1, The complexity will become $O(N^2 * \log(N))$ which is not useful for us.

To find this polynomial efficiently we will use Divide And conquer:
For a given (L, R)
We will find polynomials (L, mid) and (mid + 1, R)
And multiply them using Fast Fourier Transform to obtain our answer.
In the base case we will return the polynomial with all coefficients = 1 till freq[elem] where elem is the value of the element being considered in the base case and freq[x] is the number of times x appears in the N balls.

We finally return the Kth coefficient in polynomial of (1, 200000) as our answer.

## Time Complexity:

**$O(N * \log^2(N))$**

## Similar Problems:

[First]()
[Second]()
[Third]()