**#537 Editorial**

[**https://codeforces.com/blog/entry/64989**](https://codeforces.com/blog/entry/64989)

**Learnt:**

* **Binary Indexed Tree / Fenwick Tree** - O(logn) operation to get sum in range of array and to update the ith element. (for reconstructing the tree on an update).

For constructing the tree it takes – O(nlogn ) time complexity.

Space required is O(n) for storing the \*BIT int array.

* Calculating Binary Coefficient **C(n, r) mod p** – using **Lucas’s Theorem** and finding Multiplicative modular Inverse using **Fermat’s Little theorem** and **repeated squaring method**. Refer <https://discuss.codechef.com/questions/3869/best-known-algos-for-calculating-ncr-m> .

Idea is to pre-calculate both factorials modulo p, and inverse factorials modulo p.

* Modular Exponentiation – Repeated squaring method.
* Application of Lucas’s Theorem – find if **C(n, r) mod p** is even or odd?
* Euler’s Totient function.