Discrete mathematics is a vast topic, but here are some essential things that you should know to be able to solve programming contest problems:

* **Counting :** You are going to see a lot of problems like "Count the number of ways to do X". In most of the cases, enumerating all possible ways is simply not going to work, and you are expected to use some kind of [**dynamic programming**](https://en.wikipedia.org/wiki/Dynamic_programming). To be able to solve counting problems properly, you need to have a good understanding of certain topics such as:
  + [**Recurrence relations**](https://en.wikipedia.org/wiki/Recurrence_relation) : Ubiquitous in counting problems. In particular, learn to come up with recurrences for counting. Also, learn to convert recurrences of high complexity to lower complexity
  + **Matrix exponentiation** : In the case of recurrences that have constant coefficients, it is possible to express the final state vector as the product of the power of a matrix and the initial state vector. A good example is [calculating Fibonacci numbers using matrix multiplication](http://en.wikipedia.org/wiki/Fibonacci_number#Matrix_form). This trick helps to reduce the complexity of finding Nth term of some recurrences from O(N) to O(Log N)
  + [**Binomial coefficient**](http://en.wikipedia.org/wiki/Binomial_coefficient): Many counting problems with symmetries can be reduced to finding a few cases and multiplying them with some binomial coefficients. So learn to calculate these efficiently depending on the type of problem. There are different kinds of precomputations one does. Also, try to learn the properties of binomial coefficients modulo primes.
  + [**Pigeonhole principle**](http://en.wikipedia.org/wiki/Pigeonhole_principle): More of a proof-aid technique rather than something that you directly apply. But I have used this in cases where I was required to find two cases which share some property. [Birthday attack](http://en.wikipedia.org/wiki/Birthday_attack) can be thought of as the probabilistic variant of pigeonhole principle, but it is not that commonly used.
  + [**Inclusion–exclusion principle**](http://en.wikipedia.org/wiki/Inclusion%E2%80%93exclusion_principle) : How to formalise dealing with double counting. It is sometimes so much easier to double count in the beginning and use inclusion-exclusion later than to try some double counting avoiding recurrence
  + **Subsets and Permutations** : Learn to generate subsets of a set and permutations of a sequence. There are some counting problems where you are expected to sum up quantities over permutations or subsets. **DP over subsets** is something that you should learn too in this regard.
  + **Indistinguishable and distinguishable objects** : Learn the difference, and learn how counting changes between the two cases.
  + **Lattice problems** : Counting the number of ways to go between two points on a lattice with various kinds of restrictions is a very common problem. Learn the basic method, and how things like binomial coefficients and [Catalan numbers](http://en.wikipedia.org/wiki/Catalan_number)appear in the solutions.
  + **Counting on graphs** : The questions of this type that I have seen commonly include tree DPs. Counting problems on non-tree graphs are usually hard. You see problems coming up based on [Kirchhoff's theorem](http://en.wikipedia.org/wiki/Kirchhoff%27s_theorem) once in a while too.
  + [**Pólya enumeration theorem**](http://en.wikipedia.org/wiki/P%C3%B3lya_enumeration_theorem) : I have seen this coming up in contests only a couple of times. But learn this and if something comes up, it should be easy for you.
* **Learn to work with moduli :** Most of the times, you are asked to do the counting mod some big prime (most favourite is 10^9 +7). Learn to do the calculations without overflow, and learn some basic properties of primes from group theory which help solve some problems.
* **Probability and** [**Expectation value**](http://en.wikipedia.org/wiki/Expected_value) : Here are the basic things you should know:
  + **Calculating probabilities as ratios of counting problems** : Do this and you can calculate things to very good accuracy
  + **Recurrences for probabilities and expectation values :** In a lot of cases, even when you can count and take ratios to get probabilities, it is much easier to find a recurrence directly in terms of probabilities. Taking care of double counting can be a pain though
  + **Linearity of expectation value** : Learn to use it right (I can't) and many expectation value problems should be cakewalks for you
  + [**Random walk**](https://en.wikipedia.org/wiki/Random_walk): Modified random walk problems come up very often as probability/expectation value problems, so learn the basic results
  + **Coin toss problems** : Another set of practically overused problems

*PS: I have probably missed a couple of thing*