Signup and get free access to 100+ Tutorials and Practice Problems   Start Now

← Notes

▲ Number Theory - II

72   Number Theory   Probability   Probability-theory   Permutation   Combinatorics   Combinations
         Code Monk

Introduction

Problems in competitive programming which involve Mathematics are are usually about number theory, or geometry. If you know number theory, that increases your ammo heavily in solving a lot of tougher problems, and helps you in getting a strong hold on a lot of other problems, too.

Problems in competitive programming require insight, so just knowing some topics is not enough at all. All of the problems requires more or less math tough. For instance, solving large systems of equations and approximating solutions to differential equations.

Set Theory

Before we proceed, let us get through the some common set operations.
Q.) What is a Set?
-In mathematics, a set is a collection of distinct objects, considered as an object in its own right.
For example, the numbers 2, 4, and 6 are distinct objects when considered separately, but when they are considered collectively they form a single set of size three, written {2,4,6}.
Sets are one of the most fundamental concepts in mathematics.
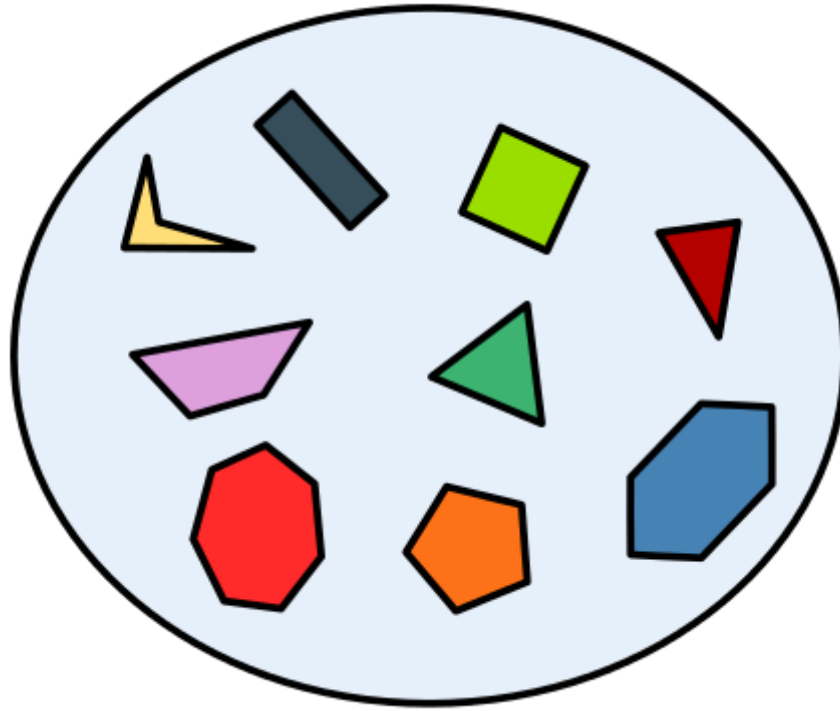
A set of polygons in a Venn diagram.

Subsets

If every member of set **A** is also a member of set **B**, then **A** is said to be a **subset** of **B**, written
A ⊆ B (also pronounced A is contained in B).
Equivalently, we can write **B ⊇ A**, read as B is a **superset** of A, B includes A, or B contains A.
The relationship between sets established by ⊆ is called **inclusion** or **containment**.
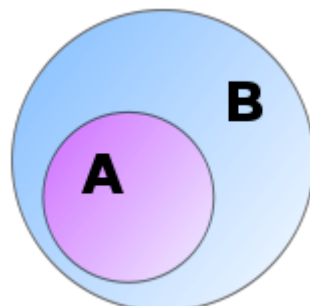If A is a subset of, but not equal to, B, then A is called a proper subset of B, written A ⊊ B (**A** is
a **proper subset** of **B**) or B ⊋ A (B is a **proper superset** of A).
Example:
- {1, 3} ⊆ {1, 2, 3, 4}.
- {1, 2, 3, 4} ⊆ {1, 2, 3, 4}.
The empty set (denoted by the symbol ∅) is a subset of every set and every set is a subset of
itself:
- ∅ ⊆ A.
- A ⊆ A.

A is subset of B.

Basic operations on Sets

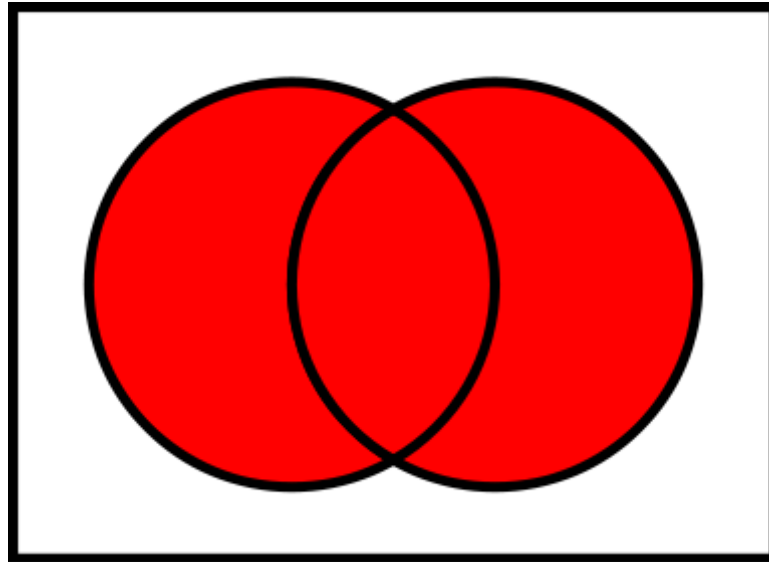There are several fundamental operations for constructing new sets from given sets.

- **Unions**:

  Two sets can be "added" together. The union of A and B, denoted by A ∪ B, is the set of all things that are members of either A or B.

  Examples:

  -{1, 2} ∪ {1, 2} = {1, 2}.

  -{1, 2} ∪ {2, 3} = {1, 2, 3}.

  -{1, 2, 3} ∪ {3, 4, 5} = {1, 2, 3, 4, 5}

The **union** of A and B, denoted A ∪ B.

  Some basic properties of unions:

  -A ∪ B = B ∪ A.

  -A ∪ (B ∪ C) = (A ∪ B) ∪ C.

  -A ⊆ (A ∪ B).

  -A ∪ A = A.
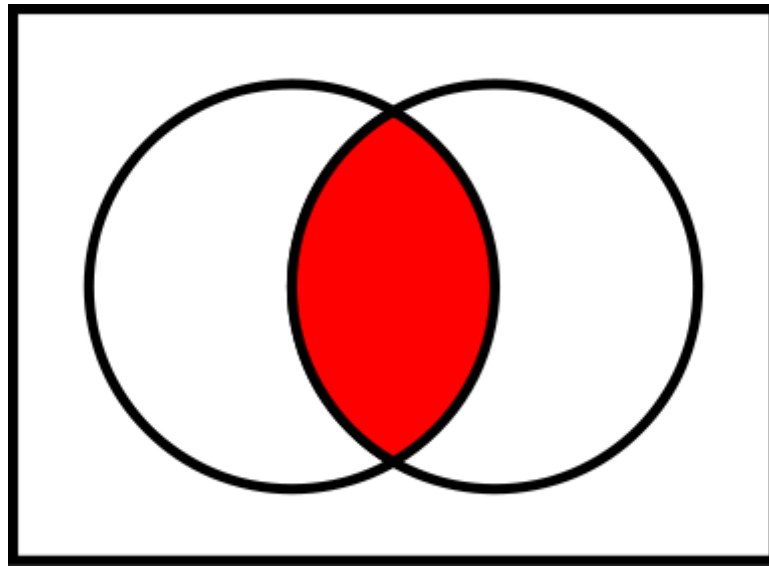
  -A ∪ ∅ = A.

  -A ⊆ B if and only if A ∪ B = B.

- **Intersections**:

  A new set can also be constructed by determining which members two sets have "in common". The intersection of A and B, denoted by A ∩ B, is the set of all things that are members of both A and B. If A ∩ B = ∅, then A and B are said to be disjoint.

  Examples:

  -{1, 2} ∩ {1, 2} = {1, 2}.

  -{1, 2} ∩ {2, 3} = {2}.

The **intersection** of A and B, denoted A ∩ B.

Some basic properties of intersections:

-A ∩ B = B ∩ A.

-A ∩ (B ∩ C) = (A ∩ B) ∩ C.

-A ∩ B ⊆ A.

-A ∩ A = A.

-A ∩ ∅ = ∅.

-A ⊆ B if and only if A ∩ B = A.

- **Complements**: Two sets can also be "subtracted". The relative complement of B in A (also called the set-theoretic difference of A and B), denoted by A B (or A − B), is the set of all elements that are members of A but not members of B. Note that it is valid to "subtract" members of a set that are not in the set, such as removing the element green from the set {1, 2, 3}; doing so has no effect.
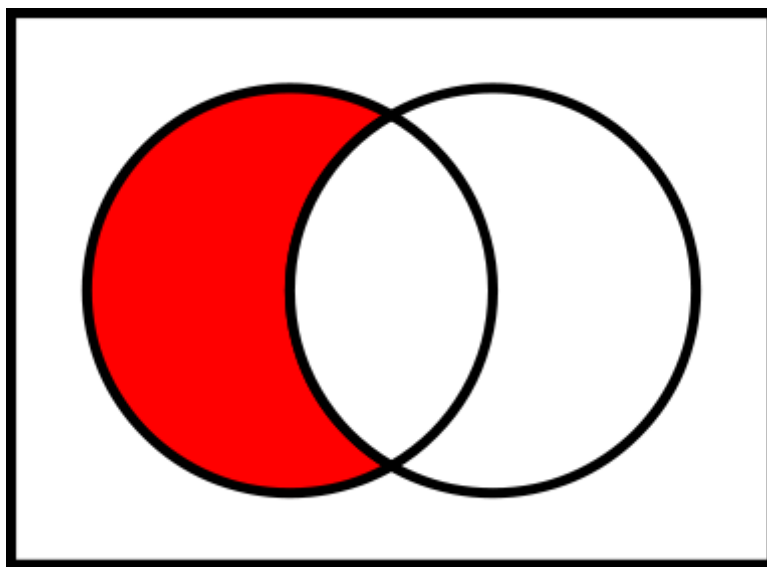 In certain settings all sets under discussion are considered to be subsets of a given universal set U. In such cases, U A is called the absolute complement or simply complement of A, and is denoted by A'.
  Examples:
  -{1, 2} {1, 2} = ∅.
  -{1, 2, 3, 4} {1, 3} = {2, 4}.
  -If U is the set of integers, E is the set of even integers, and O is the set of odd integers, then U E = E' = O.

The **relative complement** of B in A.
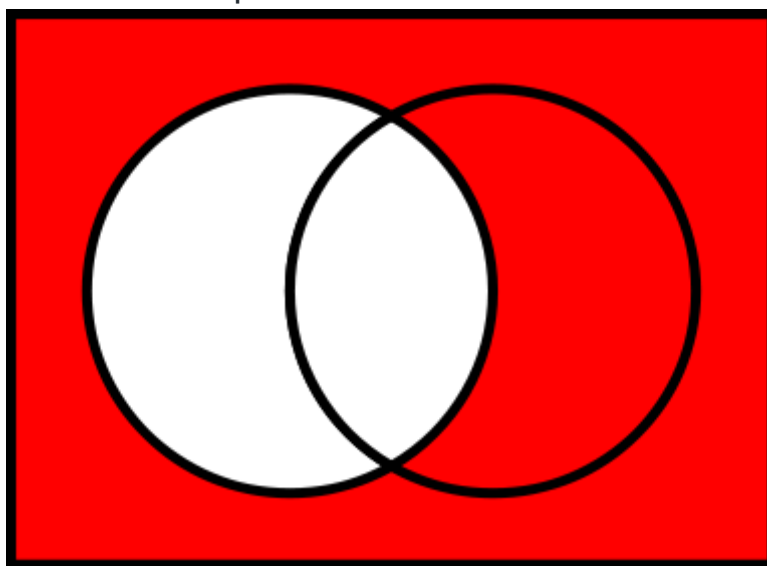
The **complement** of A in U.

Some basic properties of complements:

-A B ≠ B A for A ≠ B.

-A ∪ A′ = U.

-A ∩ A′ = ∅.

-(A′)′ = A.

-A A = ∅.

-A B = A ∩ B′.

-U′ = ∅ and ∅′ = U.

Basics of Combinatorics

NOTE: $|A|$ here denotes the number of elements in Set A.

The basic rules of combinatorics one must remember are:

1. **The Rule of Product**:

   The product rule states that if there are $|A|$ possibilities from set A and $|B|$, then there are $|A| \times |B|$ ways to combine one from A and one from B.

2. **The Rule of Sum**:

   The sum rule states that if there are $|A|$ possibilities from set A and $|B|$ possibilities from set B, then there are $|A| + |B|$ ways for either A or B occur assuming the elements of A and B are distinct.

3. **Inclusion-Exclusion Formula: (also know as the sieve principle)**:

   $$|A \cup B| = |A| + |B| - |A \cap B|$$

   Generalising the above formula, we get:

   $$\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{i=1}^{n} (-1)^{i-1} \sum_{\substack{I \subseteq \overline{1,n}, \\ |I|=i}} \left| \bigcap_{j \in I} A_j \right|$$

   The reason this works is that summing the sets double counts certain possibilities, namely, those occurring in both sets.
   Double counting is a slippery aspect of combinatorics, which can make it difficult to solve problems via inclusion-exclusion.

   These rules can be used for a finite collections of sets.

Basics of Permutation

- **Permutation of Distinct Objects**:
  Suppose we have to permute **k** objects out of **n** distinct objects, then the total number of permutations is given by

  $$P = n * (n – 1) * (n – 2) * ... * (n – k + 1)$$

  which when simplified gives us

  $$P = n! / (n – k)!$$

  where

  $$n! = n * (n – 1) * (n – 2) * ... * 2 * 1.$$

- **Permutation with Repetition**:
  If we have total n objects, where we have n1 objects of type 1, n2 objects of type 2 and so on upto k.

Thus **n1+n2+...+nk=n**. So the number of permutations of these objects is given by:

$$\binom{n}{n_1, n_2, \ldots, n_k} = C_n(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k!}$$

Combinatorial Objects

A bijection is a one-to-one mapping between the elements of one set and the elements of another. Counting the size of one of the sets automatically gives you the size of the other set. Exploiting bijections requires us to have a repertoire of sets which we know how to count, so we can map other objects to them.

- **Combinations without repetition**:

  In combinations we choose a set of elements (rather than an arrangement, as in permutations) so the order doesn't matter. The number of different k-element subsets (when each element can be chosen only once) of n-element set is:

  $$\binom{n}{k} = C_k^n = \frac{n!}{k! \cdot (n-k)!}$$

- **Combinations with repetition**:

  Let's say we choose k elements from an n-element set, the order doesn't matter and each element can be chosen more than once. In that case, the number of different combinations is:

  $$f_k^n = \binom{n+k-1}{k} = \frac{(n+k-1)!}{k! \cdot (n-1)!}$$

  It is useful to know that f_k^n is also the number of integer solutions to this equation:

  $$x_1 + x_2 + \ldots + x_n = k, x_i \geq 0 (i \in \overline{1, n})$$

Binary Vectors

Accordingly, some knowledge of the basic combinatorial properties of binary vectors is rather important. Let's have a look at some simple things associated with them:

1. Number of binary vectors of length **n**: 2^n.

2. Number of binary vectors of length **n** and with **k** '1' is $\binom{n}{k}$ since we just choose **k** positions for our '1's.

3. The number of ordered pairs (a, b) of binary vectors, such that the distance between them (k) can be calculated as follows: $\binom{n}{k} \cdot 2^n$.
   The distance between a and b is the number of components that differs in a and b.

Recurrence Relations

Recurrence relations make it easy to count a variety of recursively defined structures. Recursively defined structures include trees, lists, well-formed formulae, and divide-and-conquer algorithms - so they lurk wherever computer scientists do.

A recurrence relation is an equation which is defined in terms of itself. They are useful because many natural functions are easily expressed as recurrences:

- **Polynomials**: $a_n = a_{n-1} + 1, a_1 = 1 \longrightarrow a_n = n$

- **Exponentials**: $a_n = 2a_{n-1}, a_1 = 2 \longrightarrow a_n = 2^n$

  Weird but interesting functions otherwise hard to represent:

  $$a_n = na_{n-1}, a_1 = 1 \longrightarrow a_n = n!$$

  It is often easy to find a recurrence as the answer to a counting problem. Solving the recurrence to get a nice closed form can be somewhat of an art, but as we shall see, computer programs can easily evaluate the value of a given recurrence even without the existence of a nice closed form.

Binomial Coefficients

$$\binom{n}{k}$$

The most important class of counting numbers are the binomial coefficients, where counts the number of ways to choose k things out of n possibilities.

- **Paths Across a Grid**:

  $$n \times m$$

  How many ways are there to travel from the upper-left corner of an   grid to the lower-right corner by walking only down and to the right? Every path must consist of n m to the right. Every path with a different set of downward moves is distinct, so there are

  $$\binom{n + m}{n}$$

  such sets/paths.

  Computation of binomial coefficients can sometimes cause arithmetic overflow in intermediate steps. So a more stable way to perform calculations is using the following formula:

  $$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

Other Counting Sequences

?

1. **Catalan Numbers**: The recurrence and associated closed form

$$C_n = \sum_{k=0}^{n-1} C_k C_{n-1-k} = \frac{1}{n+1}\binom{2n}{n}$$

   [Catalan Numbers and it's uses](#).

2. **Eulerian Numbers**: The Eulerian numbers $\left\langle{n \atop k}\right\rangle$ count the number of permutations of length n with exactly k ascending sequences or runs. A recurrence can be formulated by considering each permutation p of 1,2,...,n-1. There are n places to insert element n, and each either splits an existing run of p or occurs immediately after the last element of an existing run, thus preserving the run count. Thus

$$\left\langle{n \atop k}\right\rangle = k\left\langle{n-1 \atop k}\right\rangle + (n-k+1)\left\langle{n-1 \atop k-1}\right\rangle.$$

3. **Integer Partitions**: An integer partition of n is an unordered set of positive integers which add up to n. For example, there are seven partitions of 5, namely, (5), (4, 1), (3, 2), (3, 1, 1), (2, 2, 1), (2, 1, 1, 1) and (1, 1, 1, 1, 1) .

   The easiest way to count them is to define a function **f(n,k)** giving the number of integer partitions of **n** with largest part at most **k**. In any acceptable partition the largest part either does or does not reach with limit, so **f(n,k)=f(n-k,k)+f(n,k-1)** . The basic cases are **f(1,1)=1** and **f(n,k)=0** whenever k>n.

Introduction to Probabilities

Basics

Working with probabilities is much like conducting an experiment. An **outcome** is the result of an experiment or other situation involving uncertainty. The set of all possible outcomes of a probability experiment is called a **sample space**. Each possible result of such a study is represented by one and only one point in the sample space, which is usually denoted by S. Let's consider the following experiments:

Rolling a die once: Sample space S = {1, 2, 3, 4, 5, 6}

Tossing two coins: Sample space S = {(Heads, Heads), (Heads, Tails), (Tails, Heads), (Tails, Tails)}

We define an **event** as any collection of outcomes of an experiment. Thus, an event is a subset of the sample space S. If we denote an event by **E**, we could say that **E⊆S**. If an event consists of a single outcome in the sample space, it is called a simple event. Events which consist of more than one outcome are called compound events.

?

What we are actually interested in is the probability of a certain event to occur, or **P(E)**. By definition, **P(E)** is a real number between 0 and 1, where 0 denotes the impossible event and 1 denotes the certain event (or the whole sample space).

| Event:	Impossible Event	Even chance	Certain
Probability:	0	1/2	1
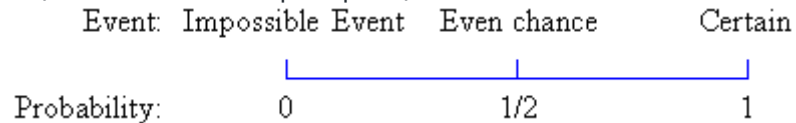
As stated earlier, each possible outcome is represented by exactly one point in the sample space. This leads us to the following formula:

$$P(E) = \frac{|E|}{|S|}$$

Independent Events

When two events are said to be independent of each other, what this means is that the probability that one event occurs in no way affects the probability of the other event occurring. An example of two independent events is as follows; say you rolled a die and flipped a coin. The probability of getting any number face on the die in no way influences the probability of getting a head or a tail on the coin.
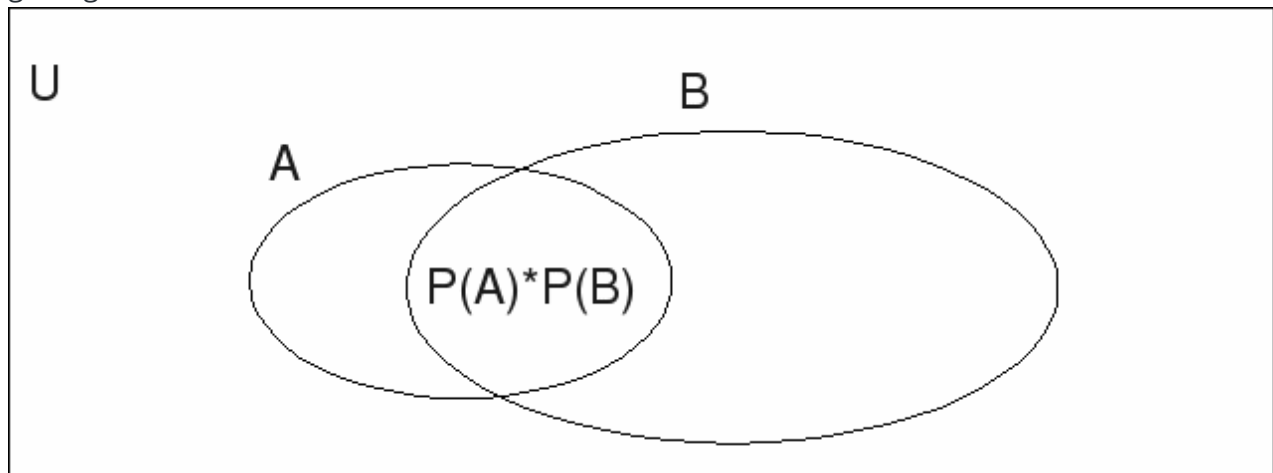
If two events A and B are independent, then neither can influence the other, and we can write

$$P(A \cap B) = P(A) \times P(B).$$

Conditional Probability

Conditional probability deals with further defining dependence of events by looking at probability of an event given that some other event first occurs.
Conditional probability is denoted by the following:

$$P(B|A).$$

The above is read as the **probability that B occurs given that A has already occurred.**

?

The above is mathematically defined as:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$
.

Rules of Probability

When dealing with more than one event, there are certain rules that we must follow when studying probability of these events. These rules depend greatly on whether the events we are looking at are Independent or dependent on each other.
First acknowledge that

$$P(S) = P(A \cup B \cup C)$$

1. **Multiplication Rule (A∩B):**
   This region is referred to as 'A intersection B' and in probability; this region refers to the event that both **A** and **B** happen. When we use the word **and** we are referring to multiplication, thus **A and B** can be thought of as **AxB** or (using dot notation which is more popular in probability) **A•B**
   If **A** and **B** are **dependent** events, the probability of this event happening can be calculated as shown below:

   $$P(A \cap B) = P(A \cup B) - (P(A \text{ only}) + P(B \text{ only}))$$.

   If **A** and **B** are **independent** events, the probability of this event happening can be calculated as shown below:

   $$P(A \cap B) = P(A) \times P(B)$$.

   Conditional probability for two independent events can be redefined using the relationship above to become:

   $$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

   $$P(B|A) = \frac{P(A) \times P(B)}{P(A)}$$

   $$P(B|A) = P(B)$$

   The above is consistent with the definition of independent events, the occurrence of event **A** in no way influences the occurrence of event **B**, and so the probability that event **B** occurs given that event **B** has occurred is the same as the probability of event **B**.

2. **Additive Rule (A∪B):**
   In probability we refer to the addition operator (**+**) as **or**. Thus when we want to we want to define some event such that the event can be A or B, to find the probability of that event:

   $$P(A + B) = P(A \cup B)$$
   $$P(A \cup B) = A + B - P(A \cap B)$$

   Thus it follows that:

   $$P(A + B) = P(A) + P(B) - P(A \cap B)$$

?

But remember from set theory that and from the way we defined our sample space above:

$$(A \cup B) = 1 - (A \cup B)'$$

and that:

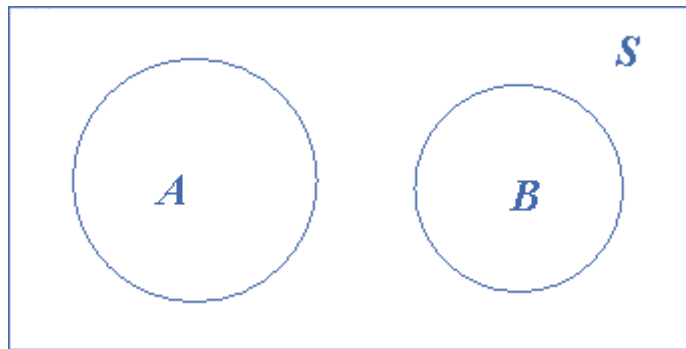$$(A \cup B)' = C$$

So we can now redefine out event as

$$P(A + B) = 1 - P(A \cup B)'$$
$$P(A + B) = 1 - P(C)$$

Mutually Exclusive Events

Two events are called mutually exclusive or disjoint if they do not have any outcome common between them. If the two events **A** and **B** are mutually exclusive, then A∩B=ɸ (**null set**). For three mutually exclusive events **A**, **B** and **C**, we have A∩B∩C=ɸ.

Rules of Probability for Mutually Exclusive Events

- **Multiplication Rule**

    From the definition of mutually exclusive events, we should quickly conclude the following:
    $$P(A \cap B) = 0$$

- **Addition Rule**

    As we defined above, the addition rule applies to mutually exclusive events as follows:
    $$P(A + B) = 1$$

- **Subtraction Rule**

    From the addition rule above, we can conclude that the subtraction rule for mutually exclusive events takes the form:
    $$P(A \cup B)' = 0$$

Conditional Probability for Mutually Exclusive Events

We have defined conditional probability with the following equation:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

We can redefine the above using the multiplication rule:

$$P(A \cap B) = 0$$

hence

$$P(B|A) = \frac{0}{P(A)} = 0$$.

Bayes' Theorem

In probability theory and statistics, Bayes' theorem describes the probability of an event, based on conditions that might be related to the event.

Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)},$$

where A and B are events - P(A) and P(B) are the probabilities of A and B without regard to each other. - P(A | B), a conditional probability, is the probability of A given that B is true. - P(B | A), is the probability of B given that A is true.

Extended Form

Often, for some partition{Aj} of the event space, the event space is given or conceptualized in terms of P(Aj) and P(B | Aj). It is then useful to compute P(B) using the law of total probability:

$$P(B) = \sum_j P(B \mid A_j)P(A_j),$$

$$\Rightarrow P(A_i \mid B) = \frac{P(B \mid A_i)\,P(A_i)}{\sum_j P(B \mid A_j)\,P(A_j)}.$$

Randomized Algorithms

We call randomized algorithms those algorithms that use random numbers to make decisions during their execution. Unlike deterministic algorithms that for a fixed input always give the same output and the same running-time, a randomized algorithm behaves differently from execution to execution. Basically, we distinguish two kind of randomized algorithms:

1. Monte Carlo algorithms: may sometimes produce an incorrect solution - we bound the probability of failure.
2. Las Vegas algorithms: always give the correct solution, the only variation is the running time - we study the distribution of the running time.
   Read these lecture notes from the College of Engineering at UIUC for an example of how these algorithms work.

   The main goal of randomized algorithms is to build faster, and perhaps simpler solutions. Being able to tackle "harder" problems is also a benefit of randomized algorithms. As a result, these algorithms have become a research topic of major interest and have already been utilized to more easily solve many different problems.

Some problems have many possible solutions, where a number of which are also optimal. The classical approach is to check them one by one, in an established order. But it cannot be guaranteed that the optima are uniformly distributed in the solution domain. Thus, a deterministic algorithm may not find you an optimum quickly enough. The advantage of a randomized algorithm is that there are actually no rules to set about the order in which the solutions are checked and for the cases when the optima are clustered together, it usually performs much better.

Delving deeper into the Principle of Inclusion-Exclusion

The verbal formula

The principle of inclusions-exclusions is as follows:

To calculate the size of combining several sets, it is necessary to sum the sizes of these sets **separately**, then subtract the size of all **pairwise** intersections of these sets, add back the sizes of intersections of all possible **triples** of sets, subtract the sizes of the intersections **of fours**, and so on, up to the intersection **of all** sets.

The formulation in terms of sets

In mathematical form the above verbal formulation is as follows:
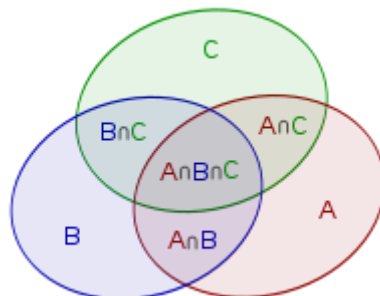
$$\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{i=1}^{n} |A_i| - \sum_{\substack{i,j: \\ 1 \le i < j \le n}} |A_i \cap A_j| + \sum_{\substack{i,j,k: \\ 1 \le i < j < k \le n}} |A_i \cap A_j \cap A_k| - \ldots + (-1)^{n-1} |A_1 \cap \ldots \cap A_n|.$$

It can be written more compactly, using the sum of the subsets. We denote B the set whose elements are A_i. Then the principle of inclusions-exclusions takes the form:

$$\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{C \subseteq B} (-1)^{size(C)-1} \left| \bigcap_{e \in C} e \right|.$$

Formulation using Venn diagrams

Let the chart shows three figures A, B and C.

Then the area of their Union $A \cup B \cup C$ is equal to the sum of the areas A, B and C less double-covered areas $A \cap B$, $A \cap C$ and $B \cap C$ but with the addition of three covered area

$A \cap B \cap C$:
$$S(A \cup B \cup C) = S(A) + S(B) + S(C) - S(A \cap B) - S(A \cap C) - S(B \cap C) + S(A \cap B \cap C).$$

Similarly, it can be summarized to Association of **n** figures.

The formulation in terms of probability theory

If there are A_i (i=1,...,n) events, $P(A_i)$ their probabilities, the probability of their Association (i.e., what will happen to at least one of these events) is:

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i) - \sum_{\substack{i,j: \\ 1 \le i < j \le n}} P(A_i \cap A_j) +$$
$$+ \sum_{\substack{i,j,k: \\ 1 \le i < j < k \le n}} P(A_i \cap A_j \cap A_k) - \dots + (-1)^{n-1} P(A_1 \cap \dots \cap A_n).$$

This sum can also be written as the sum of the subsets of a set B whose elements are the events A_i.

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{C \subseteq B} (-1)^{size(C)-1} \cdot P\left(\bigcap_{e \in C} e\right).$$

Proof of principle of inclusions-exclusions

For the proof it is convenient to use a mathematical formulation in terms of set theory:

$$\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{C \subseteq B} (-1)^{size(C)-1} \left| \bigcap_{e \in C} e \right|,$$

where B, we recall, is the set consisting of A_is.

We need to prove that any element contained in at least one of the sets A_i, will allow for a formula exactly once. (Note that other elements not contained in any of A_i, can not be considered as absent on the right side of the formula).

Consider an arbitrary element **x** contained in exactly **k>=1** one of the sets of A_i. We will show that it will be calculated by the formula exactly once.

Note that:

- in those terms, in which $size(C) = 1$, the item **x** will be counted exactly **k** once, with a plus sign;
- in those terms, in which $size(C) = 2$, the item **x** will be taken into account (with a minus sign) exactly C_k^2 once because **x** it is counted only in those terms that correspond to two sets of **k** sets containing **x**;
- in those terms, in which $size(C) = 3$, the item **x** will be counted exactly C_k^3 once, with a plus sign;
- ...

?

- in those terms, in which $size(C) = k$, the item **x** will be counted exactly $C_k'^k$ once, with the sign $(-1)^{k-1}$;

- in those terms, in which $size(C) > k$, the item **x** will be counted zero times. Thus, we need to calculate a sum of [binomial coefficients](#):

$$T = C_k^1 - C_k^2 + C_k^3 - \ldots + (-1)^{i-1} \cdot C_k^i + \ldots + (-1)^{k-1} \cdot C_k^k.$$

Easiest way to calculate this amount is by comparing it to the decomposition in the Newton binomial expression $(1 - x)^k$.

$$(1 - x)^k = C_k^0 - C_k^1 \cdot x + C_k^2 \cdot x^2 - C_k^3 \cdot x^3 + \ldots + (-1)^k \cdot C_k^k \cdot x^k.$$

It can be seen that **x=1** the expression $(1 - x)^k$ represents not that other, as $1 - T$. Therefore $T = 1 - (1 - 1)^k = 1$, what we wanted to prove.

Implementations and Examples:

1. Using the recurrent formula **nCr=(n-1)C(r)+(n-1)C(r-1)** we can calculate the nCr%non-prime or nCr%prime or simply nCr (for small n only):
   [Check the implementation here](#).

2. Computing binomial coefficients i.e. N choose R using O(n) precomputation. Use this for large value of N and when (NchooseR)%prime is needed.
   [Check the implementation here](#).

3. **Example**: The number of relatively Prime numbers in the given range
   This is an easy question based on the principle of Inclusion-Exlcusion.
   Given n and r. You want to count the number of integers in the interval $[1; r]$, coprime with n.
   Algorithm: Go straight to the inverse — calculate the number of not mutually Prime numbers.
   Consider all the Prime factors of a number n; we denote them using $i = 1 \ldots k$.
   How many numbers are in the interval r divisible by p_i ? They are:

   $$\left\lfloor \frac{r}{p_i} \right\rfloor$$

   However, if we simply sum these numbers, we get the wrong answer — some numbers will be totaled several times (those that are divided to several p_i). Therefore it is necessary to use the formula of inclusions-exclusions.
   For example, 2^k to iterate over a subset of the set of all p_i's, calculate their product, and to add or subtract in the formula of inclusions-exclusions next term.
   The final **implementation** is to count the number of relatively Prime numbers:
   [You can check the main implementation part here](#).

Problems for Practice:

- **Utkarsh and Jumps**:

  This is fairly easy problem based on probability and dynamic-programming.

  We know that the probability for Utkarsh to be at X=0 is 1.0. Using this we compute the probability of Utkarsh being at that position at some point of time.

  We use the fact the to reach ith position Utkarsh has 2 choices: either he could reach the ith position from (i-2)th positon or he could reach the ith position from (i-3)th position.

  So the probability to reach the ith position would be P(i-2) * (p/100.0) + P(i-3) * (1-p/100.0) where P(i) is the probability of Utkarsh being at the ith position.

  You can check my accepted solution for reference.

  My Code.

- **Ankit and Race Team**:

  This problem is based upon Combinatorics.

  We have to find the summation of minimum strengths of student in each group of size k. Naive approach of forming all groups of size k and then computing might give TLE. But we can use the fact that each after sorting all the students in increasing order of their strengths, we would know the exact number of times ith students would be minimum in all the groups in which he/she can be present.

  So after sorting we find the possible number of selections for ith student such that strength[i] would be minimum which comes out to be **(n-i-1)C(k-1)**. Summing over all i's we get the answer.

  You can check my accepted solution for reference.

  My Code.

- **Tic Tac Toe**:

  Easy level problem on Combinatorics.

- **Roy and Little Mario**:

  Problem can be reduced to implementation of Tribonacci Series.

- **So Random**:

  Given the probability of raining P, we know the probability of no rain will be (1-P) and since Raj will be out for time interval t, so the probability of not raining in this interval would be (1-P)^t.

  Since, we need the probability of rain in this interval, we subtract probability of no rain in the interval which is (1-P)^t from 1, which is our required answer.

Author: Tanmay Chaudhari

| Like 0 | Tweet | G+ |

COMMENTS (16) ⟳   SORT BY: Relevance ▾

Login/Signup to Comment

Manish Kushwaha 2 years ago

all problems
https://www.hackerearth.com/code-monk-number-theory-ii/problems/
▲ 1 vote ● Reply ● Message ● Permalink

Chandra Shekhar 3 years ago
Can author provide sample program to implement above mentioned concepts and links to problems based on these concepts on hacekrearth.
▲ 0 votes ● Reply ● Message ● Permalink

Tanmay Chaudhari 3 years ago
Yeah! sure..give me some time please :)
▲ 0 votes ● Reply ● Message ● Permalink

Chandra Shekhar 3 years ago
OK.. :)
▲ 0 votes ● Reply ● Message ● Permalink

Chandra Shekhar 3 years ago
theory is well written :)
▲ 0 votes ● Reply ● Message ● Permalink

codetopass 3 years ago
Nice article. Can you explain your code for Ankit & race problem? Thanks
▲ 0 votes ● Reply ● Message ● Permalink

Mohit Singh 3 years ago
Yr @author bhiya ager hme formule hi padhne hote to hm coding ni karte.....
y to aap ne wikipidia se utha kar yaha post kar diya ..... Na koi example or na hi koi code ....
hme exam m copy ni bharni h bhiya....
Where is [developer:ptk23] ..... plzz explane again that tutotial ..
▲ 0 votes ● Reply ● Message ● Permalink

Ahmed Mohsen 3 years ago
too many formulas and no codes at all !!
we need codes also .. it's called "code"monk man :(!
▲ 0 votes ● Reply ● Message ● Permalink

Manish Singla 3 years ago
It will be great if some illustrative examples are there
▲ 0 votes ● Reply ● Message ● Permalink

Ankur Khandelwal 3 years ago
Hey.. Very good written.. upload some example that uses above written theory...
▲ 0 votes ● Reply ● Message ● Permalink

Mayank Pratap Singh 3 years ago
Really enjoyed this article.. :)
▲ 0 votes ● Reply ● Message ● Permalink

Rahul Raja 3 years ago
https://www.hackerearth.com/users/tanmayc25/" target="_blank">@Tanmay Chaudhari, can you please explain why for the SO RANDOM problem, the time T is T/1440 and not only T? Why has T been divided by total minutes in a day?
▲ 0 votes ● Reply ● Message ● Permalink

Mayank Pratap Singh 2 years ago

?

Most of the things taken from e-maxx.ru :P

▲ 0 votes ● Reply ● Message ● Permalink

Mayank Pratap Singh 2 years ago

Most of the things taken from e-maxx.ru :P

▲ 0 votes ● Reply ● Message ● Permalink

sShuvo Ehsan ✎ Edited 2 years ago

can anyone please explain me the above using formula of "Combinations with repetition"?

▲ 0 votes ● Reply ● Message ● Permalink

Abhishek Kumar ✎ Edited 2 years ago

It is equivalent to the number of ways of distributing k coins among n beggars which is C(k+n-1,n-1) = C(k+n-1,k)

▲ 0 votes ● Reply ● Message ● Permalink

🪄 AUTHOR

📄 1 note

TRENDING NOTES

Python Diaries Chapter 3 Map | Filter | For-else | List Comprehension
written by Divyanshu Bansal

Bokeh | Interactive Visualization Library | Use Graph with Django Template
written by Prateek Kumar

Bokeh | Interactive Visualization Library | Graph Plotting
written by Prateek Kumar

Python Diaries chapter 2
written by Divyanshu Bansal

Python Diaries chapter 1
written by Divyanshu Bansal

more ...

About Us	Innovation Management	Technical Recruitment
University Program	Developers Wiki	Blog

?

Site Language: English   ▼ | Terms and Conditions | Privacy |© 2018 HackerEarth

?