Expected Value

**Meaning:** Suppose we are trying some experiment (event) and we are focusing on some quantity **X** like speed, count, volume, count, etc., and want to find the average value of that quantity when that experiment is performed. For eg., Experiment is car is going from point A to B and it may face some speed breaker, traffic or internal problems like fuel etc, and we have the probability of occurrence those problems. Now we have to find the average time taken by the car to reach from A to B or expected time to reach the point B from A.

In mostly problems, even we are not given the probabilities, so we have to calculate them via trying all possibilities and possibilities of those events. **Ex**: We tossed two coins find the expected number of heads. So sample space: {(H, H), (H, T), (T, H), (T, T)} => converting it to the real number according the focus of the problem (number of heads) => {2, 1, 1, 0} => probabilities of each of those possibilities: {1/4, 1/2, 1/2, 1/4}. exp(number of heads) = 2\*1/4 + 1\*1/2 + 1\*1/2 + 0\*1/4 => 1/2 + 1 => 1.5

Expected values is also known as weighted average. Formula for expected value is:

**E(X) = Ʃ x.P(x).** X (random variable) => Some quantity for which the expected value has to be calculated like speed, count (occurrences), volume etc. P(x) = probability of that quantity x. x = element of the set X. E(X) => expected value of X when the experiment is performed.

**NOTE**: ƩP(x) = 1,must be true.

X is being said as **random variable**. It is necessary because in many cases like cards, coins we have to represent them in some numerical format and X helps to do so.

**Random Variable (X)**: It is a function which takes outcome from the sample space of experiment as input then it returns the real number. So, this is like a mapping the outcome to some real number. ‘X’ could have meaning or not. If some problem given you the values of ‘X’ then it also find, but if values are not given then you have find the possible values in experiment by giving some meaning to ‘X’. **Ex:** 2-coins are tossed and ‘X’ = # heads - # tails, find the expected value of X. Then, possible outcomes: S = {HH, HT, TH, TT}, X = {2, 0, 0, -2}. E(X) = 2\*1/4 + 0\*1/4 + 0\*1/4 + (-2)\*1/4 => 1/4(0) => 0.

**NOTE:** Another way to see this formula: **E(X) = Ʃ x.P(x)**

E(X) = (1/number of all possibilities) \* (sum of all possible value of X)

or

**E(X) = (1/|S|) \* (sum of values of X on each possible outcome of S)**

|S| => Size of sample space

It is like, suppose you have performed some experiment, the you wrote down each possible outcome which is represented by Xi , then:

**E(X) = 1/n (X1 + X2 … Xn) , where n = number of possible ways (size of sample space)**

**Xi = value of random variable of ith outcome.**

Like in the above problem (toss 2-coins), we have not given ‘n’ but by reading the statement we can see that there are ‘4’ possible ways, and we have calculated the value of ‘X’ for each outcome: E(X) = 1/4 \* (2 + 0 + 0 + (-2)) => 0.

There can be multiple X’s whose values are same like in above problem, but we have to consider it individually because according to sample space those outcomes are different.

**Ex**: Given an array of N integers, you picked a pair of items, find the expected sum of those items. Pair (x, y) ≠ Pair(y, x) and Pair(x, x) is also a valid pair.

**Soln**: So, as you can see this definition is quite helpful: E(X) = 1/n (X1 + X2 … Xn) .

S = {(1, 1), (1, 2)… (1, N), (2, 1), (2, 3)… (2, N)...(N, 1)...(N, N)}

The total number of pairs (n) = N2.

X = sum of the pair.

E(X) = 1/ N2 \* (Sum(1, 1), Sum(1, 2)…) => 1/N2 \* (2\*N\*(sum of all elements))

**Expected value is majorly of two types:**

1) When some finite operation is executed, then find the expected value of some item after those operations. Ex: Pick a card from the deck, what is expected number of the card. Ex: Toss 2 coins what is the expected number of the number of heads appear. In these types of problems, the operations are finite and easy to visualize.

To solve these types of problems, mostly we need to use the contribution sum or **Linearity of Expectation** or DP.

2) Here, it is just reversed of above part. Here, we have given some initial state, and we have to reach some final state, then find the expected number of operations to reach the final state. Here as you can see it is just opposite of the (1) case. Ex: You are at initially at position ‘0’ and you can make jump upto 1, 2 or 3, then find the expected number of jumps required to reach N. In these kind of problems, we generally need the **state based tree**, and then use recurrence or DP. Ex: [link](https://codeforces.com/contest/518/problem/D)

There is another type of (2) which is leads to **infinite calls**. Ex: find the expected number of times you have to toss the coin, so that you get heads first time. Here again to solve these problems you need recurrence (state tree) and then if the recurrence is depends on itself, so we solve the recurrence equation then we get the properly arranged equation where we can apply DP. Ex: in the given problem to toss the coin until 1st head, recurrence would be: **R = 1 \* 1/ 2 + 1 / 2 \* (1 + R)** (via state based tree), then solve the recurrence you get: R = 2. Initially, it will look like it goes to infinite but you have find rearrange the recurrence to get the correct result.

**Tip**: During calculating the Expected value, don’t try to bunch combine the terms to calculate probability, because this may become over-complicated and may lead to confusion. But if you leave them open, then you can see the patterns and if they want to combine then they would automatically combine and it makes formula really simple, otherwise you have to calculate the probability for each group and the groups and be exponential, but if you leave them, it becomes easy.

**Concept: Linearity of Expectation**

Suppose given, two random variables, **‘X’**, **‘Y’** whose expected values are known (or can be calculated). Now, consider **‘Z’** (random variable), which is defined as:

**Z = X + Y.** (it generally has some meaning)

We have to calculate, E(Z). It can be calculated in following way:

**E(X + Y) = E(X) + E(Y)**,

Actually here we are finding the expected value of the random variable **‘Z’** which depends on other random variables (X, Y). As you can see in **Z = X+Y**, there is **+** operation, so, you we use this concept in those problems when we have to deal like + situations like **Sum of values(area, lengths, etc..), subtraction, counting ways** etc, but I think it won’t work in those cases like minimums, maximums etc (**IT will work in all possible cases,** because minimums, maximums are mutually exclusive, that means in Z = X + Y, if X > 0 then Y = 0, if Y > 0 then X = 0). This concept is majorly used in Those events could be dependent or independent, but they feel like independent like below example. It is derived from the main expected value formula. It **doesn’t depends X, Y are independent or Not**, it always works if we are able to do so like in terms of area, counting etc.

**Tip:** First think in terms of random variable because it will be easy and intuitive. After that, you may need to construct some equation using random variables then you calculate the expected value using Linearity of Expectation.

**Ex:** Given two coins C1 and C2 are tossed, calculate the expected number of heads on C1 and tails on C2.

Soln: C1 => {H, T}, C2 => {H, T}, {C1, C2} => {HH, HT, TH, TT}

Let, X => number of heads on C1, Y => number of tails on C2.

Z => count of number of heads and number of tails (here we counting ways, adding makes sense)

Z = X + Y

E(Z) = E(X) + E(Y) => ½ + ½ => 1

**Indicator random variable (IRV)**: It is a random variable which could have only two values 1 or 0 with probability p and 1-p respectively. It indicates some item that it is picked or not.

IRV = {1, when some condition met

0, otherwise}

E(IRV) = 1\*p + 0 \* (1 -p) = p

E(IRV) = probability of being TRUE or 1.

**General way to use the Linearity of Expectation**:

1) Think in terms of **random variables** first. Define the goal random variable (Y) and then define the indicator random variables (Xi), and now define Y in terms of Xi. But make sure **this should make sense**. Y = X1 + X2 + X3  +…+ Xn

2) Now apply the linearity of expectation, E(Y) = E(X1)+ E(X2) + E(X3 )+…+ E(Xn).

3) Now calculate, E(Xi) individually and them add them by focusing on **Xi only**.

**Ex:** Choose 10 random cards find the expected value of the number of aces? X = number of aces.

**Soln**: Suppose Y = number of aces picked

X\_i => In ith turn, Is Ace picked or Not(1, if ace picked else 0)

Then by common sense we can do:

Y = X\_1 + X\_2 + ….+ X\_10

E(Y) = E(X\_1) + E(X\_2) + E(X\_3) +… + E(X\_10)

If we just **only focus on ‘ith’** turn

E(X\_i) = 1\*4/52 + 0 \* 48/52 => 1 / 13

Every operation is same, E(X\_1) = E(X\_2) =…. = E(X\_10)

E(Y) = 10\* 1 / 13

**NOTE**: Here in above example, it is about counting stuffs (Occurrence of aces specifically). What we did: we took contribution of each element (card), and in terms of counting the contribution would be either 0 or 1 (here, occurrences \* value = 1).

You can think in similar way that: you have given a grid whose expected area you have to calculate lets say E(Y), that grid is broken down in different parts (X\_1, X\_2..)

and you can calculate the expected area of those parts: E(X\_1), E(X\_2)… Then, by common sense you can say that: total area (Y) = sum of all broken parts:

Y = X\_1 + X\_2..

E(Y) = E(X\_1) + E(X\_2) …E(X\_n). Like here, Sum (adding) makes sense and it is correct that, why we have used the Linearity of expectation.

So, the main part to apply the linearity of Expectation is to break the down the problem into sub-problems and such that sub-problems depend on individual element/component then take their **contribution**. Do this first in terms of random variables.

**Contribution of any element ‘i’ => count(i) \* value\_of(i)**

Practice: <https://www.codechef.com/problems/THEGAME>

Soln: <https://www.codechef.com/viewsolution/42201336>

**NOTE**: E(X \* Y) ≠ E(X) \* E(Y) {Generally}

E(X \* Y) = E(X) \* E(Y) {When X, Y are independent}