**Problem Statement 1:-**

**Flipping a Coin**

**Puzzle based on the Flipping coin:**

You are in a room blind folded. There are 10 coins kept in front of you where 5 are kept heads up and 5 are kept heads down.

It is not possible to determine which side is up by touching them.

**Task:**

The task is to separate these coins into two piles of 5 such that both the piles have an equal number of heads up.

You are allowed to flip the coins any number of times.

**Explanation:-**

**Step 1:** Take the coins and arrange them into two piles with 5 coins each**.**

**Step 2:** Fixing one pile, flip all the coins in the other pile**.**

**Conclusion:** The number of heads in both the piles will become equal. It happens because the coins have only two probabilities, they can either have heads or a tail.

**To further simply this:-**

We know that are the 10 coins: **H H H H H T T T T T**

Now, let us consider the following cases:-

**Case 1:**

Let us consider the coins are divided in two piles in the following order

**P1:- H T T T T and P2:- T H H H H**

Now if we flip P1, then

**P1:- T H H H H**

Therefore, both the piles will have equal no. of heads.

**Case 2:**

Let us consider the coins are divided in the following order

**P1:- T H T H H and P2:- H T H T T**

Flip P1, it becomes

**P1:- H T H T T**

Therefore both the piles will have an equal no. of heads.

**Case 3:**

Let us consider the coins are divided in the following order

**P1:- H T H H H and P2:- T T T T H**

Flip P2 it becomes

**P2:- H H H H T**

Therefore both the piles will have equal no. of heads.

What is happening here is that we are fixing the number of heads in one pile whereas in the other pile we are flipping them.

The logic here is that the number of heads up and heads down coins is fixed in the beginning.

When we are dividing them into two piles, if one pile gets “x” heads up then the other pile will have “x” tails up and then we flip the other pile. It becomes “x” tails and the heads become tails.

**Problem Statement 2:-**

Count Binary Strings Without Consecutive Ones

In Count Binary Strings Without Consecutive Ones problem we have to count strings of length n such that each character of the string can only be 1 or 0. And we should not count strings with ’11’ as a sub string.

Given an integer n return the number of binary strings of length n such that they do not contain ’11’ as a sub string. The answer can be large so return the result modulo 1000000007.

**Example:**

Given n = 3

**Output:**

5

**Explanation:**

The length 3 possible binary strings are – 000, 001, 010, 011, 100, 101 , 110, 111

Out of these 011, 110 and 111 contain 11 as a sub string. Therefore the result is 5.

Count Binary Strings Without Consecutive Ones Solution

Solution 1 (Top Down dynamic programming)

For writing the top down solution we will write a function F(id , prev).

F(id , prev) means number of binary strings of length n with prev filled at (n+1)th place. Now we have to decide our transitions.

The transitions will be :-

f(n,prev) = f(n-1,0) + f(n-1,1) if(prev ==0)

or

f(n,prev) = f(n-1,0) if(prev==1)

We will be using a memo table to speed up the solution and avoid recalculation of values.

**Problem Statement 3:-**

**Coin Change Problem**

In the coin change problem, we have to count the number of ways of making change given an infinite supply of some coins.

We are given a value N ( rupees ). And we are given infinite value of M coins. What are the total number of ways we can make change using m coins. The order of coins doesn’t matter.

**Input:**

First line containing integer denoting total rupees for which we have to make change.

Second line contains integer m denoting number of coins.

Third line contains m integers denoting the denomination of each coin.

**Output:**

Number of ways to make change.

**Explanation:**

To build a top down solution we must follow the following steps –

Break Down the Problem –Since the order of a coin does not matter, the idea is to pick a coin and keep using it till you can. If at any point we decided to not pick the coin then we never pick this coin again. The recurrence relation is –

F(i,N) = Number of ways to make change of N using first i coins.

F(i,N) = F(i,N-coin[i]) + F(i-1,N) { ie using ith coin or not using ith coin}

Find the base case – We know if N is zero then we have 1 way which is 0 coins . If N becomes negative then there are zero ways. And the third base case is when we are left with zero coins but a positive value of N.

Check for optimal substructure and overlapping sub problems – It is easy to see that the problem can be broken down into smaller problems. The optimal substructure and overlapping sub problems properties can be visualized below.

