NAME: Semit

UID: 22BCS13024

Sec: 601/A

**1st: CLIMBING STAIRS**

class Solution {

public int climbStairs(int n) {

if (n == 1) return 1;

if (n == 2) return 2;

int first = 1, second = 2, result = 0;

for (int i = 3; i <= n; i++) {

result = first + second;

first = second;

second = result;

}

return result;

}

}

**2ND:UNIQUE PATHS**

class Solution {

    public int uniquePaths(int m, int n) {

        int[][] dp = new int[m][n];

        for (int i = 0; i < m; i++) {

            dp[i][0] = 1;

        }

        for (int j = 0; j < n; j++) {

            dp[0][j] = 1;

        }

        for (int i = 1; i < m; i++) {

            for (int j = 1; j < n; j++) {

                dp[i][j] = dp[i-1][j] + dp[i][j-1];

            }

        }

        return dp[m-1][n-1];

    }

    public static void main(String[] args) {

        Solution solution = new Solution();

        int result1 = solution.uniquePaths(3, 7);

        System.out.println("Unique paths for 3x7 grid: " + result1); // 28

        int result2 = solution.uniquePaths(3, 2);

        System.out.println("Unique paths for 3x2 grid: " + result2); //  3

    }

}

**3RD :COIN CHANGE**

class Solution {

    public int coinChange(int[] coins, int amount) {

        int n = coins.length;

        int[] dp = new int[amount+1];

        Arrays.fill(dp,amount+1);

        dp[0] = 0;

        for(int i=1;i<=amount;i++) {

            for(int c : coins) {

                if(i-c >= 0) {

                    dp[i] = Math.min(dp[i],dp[i-c] + 1);

                }

            }

        }

        if(dp[amount]>amount) {

            return -1;

        }

        return dp[amount];

    }

}

**4TH : JUMP GAME**

class Solution {

    public boolean canJump(int[] nums) {

       int reachable = 0;

       for(int i = 0; i < nums.length; i ++) {

           if(i > reachable) return false;

           reachable = Math.max(reachable, i + nums[i]);

       }

       return true;

    }

}