Android App Development Submission

Seasons of Code 2025

Manikarnika Sharma

Indian Institute of Technology, Bombay

May 2025 - July 2025

Contents

1 Android Studio Installation

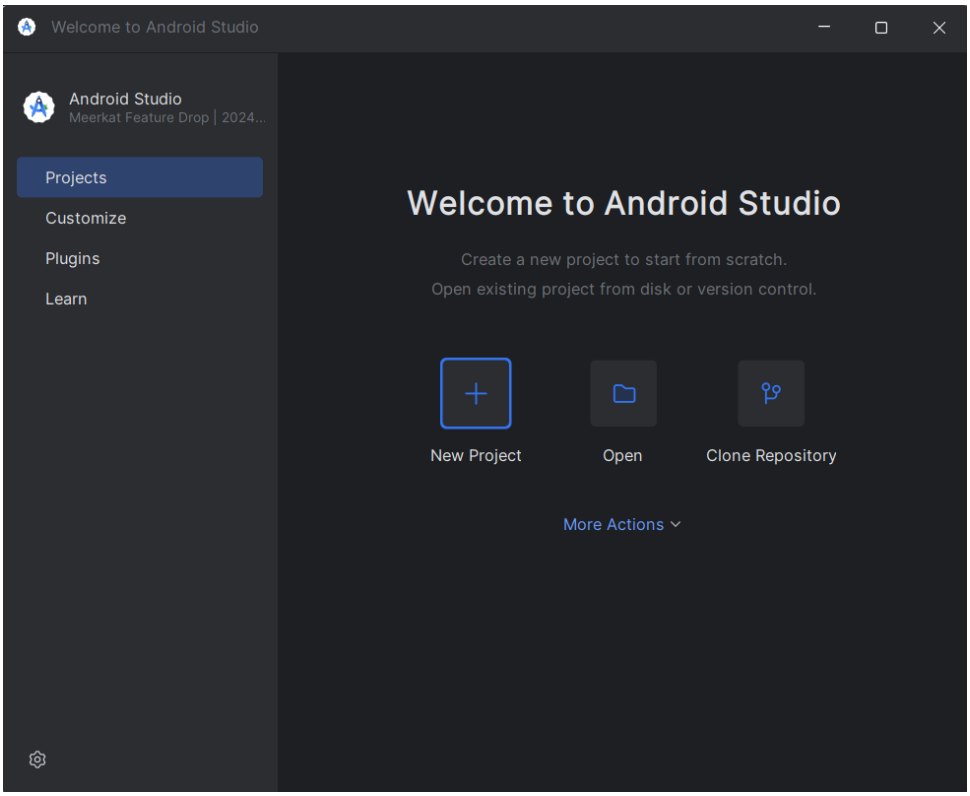

Figure 1: Android Studio

2 LeetCode Problems

2.1 2114. Maximum Number of Words Found in Sentences

Code

Java Auto

```java
class Solution {
    public int mostWordsFound(String[] sentences) {
        int countmax = 0;
        for(int i=0; i<sentences.length; i++){
            int count=1;
            for(int j=0; j<sentences[i].length(); j++){
                if(sentences[i].charAt(j)==' ') count++;
            }
            if (count>countmax) countmax=count;
        }
        return countmax;
    }
}
```

Saved Ln 24, Col 1

2.2 2235. Add Two Integers

Code

Java Auto

```java
class Solution {
    public int sum(int num1, int num2) {
    return num1+num2;
    }
}
```

Saved Ln 7, Col 1

2.3 1281. Subtract the Product and Sum of Digits of an Integer

Code

Java Auto

```java
class Solution {
    public int subtractProductAndSum(int n) {
        int product=1;
        int sum=0;
        int digit;
        int numofdigits = String.valueOf(n).length();
        for(int i = numofdigits-1; i>=0 ; i--){
            digit= n/((int)Math.pow(10, i));
            product *= digit;
            sum+= digit;
            n-= digit*Math.pow(10, i);
        }
        return product-sum;
    }
}
```

Saved Ln 16, Col 1

2.4 2413. Smallest Even Multiple

Code

Java Auto

```java
class Solution {
    public int smallestEvenMultiple(int n) {
        int worstcase = 2*n;
        for (int i=2; i<= worstcase; i++){
            if(i%2==0 && i%n==0){
                return i;
            }
            i++;
        }
        return worstcase;
    }
}
```

Saved Ln 14, Col 1

2.5 1108. Defanging an IP Address

Code

Java Auto

```java
class Solution {
    public String defangIPaddr(String address) {
        return address.replace(".", "[.]");
    }
}
```

Saved Ln 7, Col 1

2.6 1678. Goal Parser Interpretation

Code

Java Auto

```java
class Solution {
    public String interpret(String command) {
        command = command.replace("()","o").replace("(al)","al");
        return command;
    }
}
```

Saved Ln 8, Col 1

2.7 2469. Convert the Temperature

Code

Java Auto

```java
class Solution {
    public double[] convertTemperature(double celsius) {
        double ans[]= {celsius +273.15 , celsius * 1.80 +32.00};
        return ans;
    }
}
```

Saved Ln 9, Col 1

2.8 2427. Number of Common Factors

Code

Java Auto

```java
class Solution {
    public int commonFactors(int a, int b) {
        int min = Math.min(a, b);
        int count=0;
        for(int i=1; i<=a; i++){
            if(a%i==0 && b%i==0) count++;
        }
        return count;
    }
}
```

Saved Ln 11, Col 1

2.9 1342. Number of Steps to Reduce a Number to Zero

Code

Java Auto

```java
class Solution {
    public int numberOfSteps(int num) {
        int steps = 0;
        while(num>0){
        if(num%2 ==0) {
            num/=2;
            steps++;
        }
        else{
            num--;
            steps++;
        }
        }
        return steps;
    }
}
```

Saving... Ln 17, Col 1

2.10 292. Nim Game

Code

Java Auto

```java
class Solution {
    public boolean canWinNim(int n) {
        if (n%4==0) return false;
        return true;
    }
}
```

Saved Ln 7, Col 1

2.11 1920. Build Array from Permutation

Code

Java Auto

```java
class Solution {
    public int[] buildArray(int[] nums) {
        int ans[]= Arrays.copyOf(nums,nums.length);
        for (int i=0; i<nums.length; i++){
            ans[i] = nums[nums[i]];
        }
        return ans;
    }
}
```

Saved Ln 10, Col 1

2.12 1470. Shuffle the Array

Code

Java Auto

```java
class Solution {
    public int[] shuffle(int[] nums, int n) {
        int[] ans = new int[2 * n];
        for(int i=0; i<n; i++){
            ans[2*i] = nums[i];
            ans[2 * i + 1] = nums[i + n];
        }
        return ans;
    }
}
```

Saved Ln 11, Col 1

2.13 1365. How Many Numbers Are Smaller Than the Current Number

Code

Java Auto

```java
class Solution {
    public int[] smallerNumbersThanCurrent(int[] nums) {
        int ans[] =new int[nums.length];
        for(int i=0; i<nums.length; i++){
            int count=0;
            for(int j=0; j<nums.length; j++){
                if(nums[i]>nums[j]) count++;
            }
            ans[i]=count;
        }
        return ans;
    }
}
```

Saved Ln 14, Col 1

2.14 1688. Count of Matches in Tournament

Code

Java Auto

```java
class Solution {
    public int numberOfMatches(int n) {
         int count = 0;
        while (n>1) {
            int past = n / 2;
            count += past;
            n = past+ (n%2);
        }
        return count;
    }
}
```

Saved Ln 12, Col 1

2.15 509. Fibonacci Number

Code

Java Auto

```java
class Solution {
    public int fib(int n) {
        int f0=0; int f1=1; int fn=0;
        int i=0;
        if (n==1) return f1;
        while(i<n-1){
            fn=f1+f0;
            f0=f1;
            f1=fn;
            i++;
        }
        return fn;
    }
}
```

Saved Ln 15, Col 1

2.16 1431. Kids With the Greatest Number of Candies

Code

Java Auto

```java
class Solution {
    public List<Boolean> kidsWithCandies(int[] candies, int extraCandies) {
        List<Boolean> result = new ArrayList<>(Collections.nCopies(candies.length, true));
        int max=0;
        for(int i=0; i<candies.length; i++){
            if (max<candies[i]) max=candies[i];
        }
        for(int i=0; i<candies.length; i++){
            if (candies[i]+extraCandies<max){
                result.set(i,false);
            }
        }
        return result;
    }
}
```

Saving... Ln 16, Col 1

2.17 1748. Sum of Unique Elements

Code

Java Auto

```java
class Solution {
    public int sumOfUnique(int[] nums) {
        HashMap<Integer,Integer> freq = new HashMap<>();
        for(int n:nums){
            freq.put(n,freq.getOrDefault(n,0)+1);
        }
        int sum=0;
        for(int i: freq.keySet()){
            if (freq.get(i) == 1) {
        sum += i;}
        }
        return sum;
    }
}
```

Saved Ln 15, Col 1

2.18 232. Implement Queue using Stacks

Code

Java Auto

```java
class MyQueue {
    Stack<Integer> stackIn;
    Stack<Integer> stackOut;

    public MyQueue() {
        stackIn = new Stack<>();
        stackOut = new Stack<>();
    }

    public void push(int x) {
        stackIn.push(x);
    }

    public int pop() {
        if (stackOut.isEmpty()) {
            while (!stackIn.isEmpty()) {
                stackOut.push(stackIn.pop());
            }
        }
        return stackOut.pop();
    }

```

```
23      public int peek() {
24          if (stackOut.isEmpty()) {
25              while (!stackIn.isEmpty()) {
26                  stackOut.push(stackIn.pop());
27              }
28          }
29          return stackOut.peek();
30      }
31
32      public boolean empty() {
33          return stackIn.isEmpty() && stackOut.isEmpty();
34      }
35  }
36
```