



## **Design & Analysis of Algorithm (Lab)**

**Name: Ananya**

**SAPID: 590013832**

**B-33**

**Submitted to: Mr.Aryan Gupta**

**[https://github.com/ananya438/DAALAB ANANYA-590013832](https://github.com/ananya438/DAALAB_ANANYA-590013832)**

## Implement Rabin Karp Algorithm and analyze its time

```
public class RabinKarp {

    static final int d = 256; // number of characters

    static void search(String pat, String txt, int q) {

        int m = pat.length(), n = txt.length();

        int i, j, p = 0, t = 0, h = 1;

        for (i = 0; i < m - 1; i++)

            h = (h * d) % q;

        for (i = 0; i < m; i++) {

            p = (d * p + pat.charAt(i)) % q;

            t = (d * t + txt.charAt(i)) % q;

        }

        for (i = 0; i <= n - m; i++) {

            if (p == t) {

                for (j = 0; j < m; j++)

                    if (txt.charAt(i + j) != pat.charAt(j)) break;

                if (j == m)

                    System.out.println("Pattern found at index " + i);

            }

            if (i < n - m) {

                t = (d * (t - txt.charAt(i) * h) + txt.charAt(i + m)) % q;

                if (t < 0) t += q;

            }

        }

    }

}

public static void main(String[] args) {

    String txt = "ABCCDDAEFG";

    String pat = "CDD";

    int q = 101; // prime number

    search(pat, txt, q);

}
```

## O/P:

```
or a JavaFX application class must extend javafx.application.Application
● PS C:\Users\nannu\Desktop\JAVA DSA\DAA> cd "c:\Users\nannu\Desktop\JAVA DSA\DAA\" ; if ($?) { javac kabin.
Pattern found at index 3
○ PS C:\Users\nannu\Desktop\JAVA DSA\DAA>
```

## Complexity Analysis (Dynamic Programming)

### Time Complexity

Case	Complexity	Reason
Best / Average Case	$O(n + m)$	Hash comparison only
Worst Case	$O(n \times m)$	Many hash collisions

### Space Complexity

$O(1)$  — only constant extra space used.