



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3.1

Student Name Vinay Mehra†

UID 21BCS8999

Branch: BE-CSE

Section/Group: 646 – A

Date of Performance 20 Mar 2024

Semester: 6

Subject Name: PBLJ lab

Subject Code: 21CSP-319

1. Aim: Write a Program to check the largest palindromic substring.

2. Objective: Develop a program to efficiently determine the largest palindromic substring within a given string using a single-pointer approach. The program should aim to balance simplicity and effectiveness in identifying and reporting the largest palindromic substring present in the input string.

3. Apparatus Used: IntelliJ Idea

4. Algorithm:

- **Initialize Variables:** Initialize maxLength to 0, start and end to 0.
- **Iterate Through the String:**
 - Loop through each character in the string.
 - For each character, expand around it to find palindromic substrings of odd and even lengths.
 - Update start, end, and maxLength if a longer palindrome is found.
- **Output the Result:** Output the substring from index start to end as the largest palindromic substring found.

5. Code:

```
import java.util.Scanner;

public class Main {

    public static String longestPalindrome(String s) {
        int n = s.length();
        if (n == 0) return "";

        boolean[][] dp = new boolean[n][n];
        int start = 0;
        int maxLength = 1;

        // All substrings of length 1 are palindromes
        for (int i = 0; i < n; i++) {
            dp[i][i] = true;
        }

        // Check for substrings of length 2
        for (int i = 0; i < n - 1; i++) {
            if (s.charAt(i) == s.charAt(i + 1)) {
                dp[i][i + 1] = true;
                start = i;
                maxLength = 2;
            }
        }

        // Check for substrings of length greater than 2
        for (int length = 3; length <= n; length++) {
            for (int i = 0; i < n - length + 1; i++) {
                int j = i + length - 1;
                if (s.charAt(i) == s.charAt(j) && dp[i + 1][j - 1]) {
                    dp[i][j] = true;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        if (length > maxLength) {
            start = i;
            maxLength = length;
        }
    }
}

return s.substring(start, start + maxLength);
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the string: ");
    String input = scanner.nextLine();

    String result = longestPalindrome(input);
    System.out.println("Longest palindromic substring: " + result);

    scanner.close();
}
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

6. Output:

```
D:\jdk\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Enter the string: doobeeboodadnsakdn
Longest palindromic substring: doobeebood

Process finished with exit code 0
```

7. Learning Outcome:

- **Problem-solving:** Practice breaking down complex problems into simpler steps to devise a solution.
- **Efficiency:** Learn to optimize algorithms for better performance by minimizing unnecessary steps.
- **String Handling:** Gain proficiency in working with strings by learning techniques like iterating through characters and extracting substrings.