

```

public class BinaryCounting {
    public static void main(String[] args) {
        String s = "The quick brown fox jumped over the lazy dogs.";
        String binaryRepresentation = totalBinaryString(s);

        boolean found = true;
        int i = 0;
        while (found) {
            String binaryString = toNormalBinary(i);
            //System.out.println("Searching for: " + binaryString);
            if (contains(binaryRepresentation, binaryString)) {
                binaryRepresentation =
totalSearchAndRemove(binaryRepresentation, binaryString);
                //System.out.println("New binary representation: " +
binaryRepresentation);
            }
            else {
                found = false;
                //System.out.println("The smallest binary string not
found is: " + binaryString);
            }
            i++;
        }

        String noLeadingZerosFinalBinary =
removeLeadignZeros(binaryRepresentation);
        //System.out.println("Final binary representation without
leading zeros: " + noLeadingZerosFinalBinary);

        String octalRepresentation =
binaryToOctal(noLeadingZerosFinalBinary);
        //System.out.println("Final octal representation: " +
octalRepresentation);

        found = true;
        i = 0;
        while (found) {
            String octalString = toNormalOctal(i);
            //System.out.println("Searching for: " + octalString);
            if (contains(octalRepresentation, octalString)) {
                octalRepresentation =
totalSearchAndRemove(octalRepresentation, octalString);
                //System.out.println("New octal representation: " +
octalRepresentation);
            }
            else {
                found = false;
                //System.out.println("The smallest octal string not
found is: " + octalString);
            }
        }
    }
}

```

```

        i++;
    }

    System.out.println(i-2);
}

public static int toAscii(char c) {
    return (int) c;
}

public static String toNormalBinary(int n) {
    return Integer.toBinaryString(n);
}

public static String toNormalOctal (int n) {
    return Integer.toOctalString(n);
}

public static String toBinary(int n) {
    return String.format("%8s",
Integer.toBinaryString(n)).replace(' ', '0');
}

public static String totalBinaryString(String s) {
    String binary = "";
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        int asciiValue = toAscii(c);
        String binaryString = toBinary(asciiValue);
        binary += binaryString;
    }
    return binary;
}

public static boolean contains(String s, String sub) {
    return s.contains(sub);
}

public static String searchAndRemoveStart (String s, String sub) {
    int index = s.indexOf(sub);
    String start = s.substring(0, index);
    String end = s.substring(index + sub.length());
    return start+end;
}

public static String searchAndRemoveEnd (String s, String sub) {
    int index = s.lastIndexOf(sub);
    String start = s.substring(0, index);
    String end = s.substring(index + sub.length());
    return start+end;
}

```

```

    }

    public static String totalSearchAndRemove (String s, String sub) {
        String result = s;
        //System.out.println("First index: " + firstIndex);
        result = searchAndRemoveStart(s, sub);
        if (result.lastIndexOf(sub) != -1) {
            result = searchAndRemoveEnd(result, sub);
        }
        return result;
    }

    public static String removeLeadignZeros (String s){
        int i = 0;
        while (i < s.length() && s.charAt(i) == '0') {
            i++;
        }
        return s.substring(i);
    }

    public static String binaryToOctal(String binary) {
        String octal = "";
        if (binary.length() % 3 != 0) {
            int leadingZeros = 3 - (binary.length() % 3);
            for (int i = 0; i < leadingZeros; i++) {
                binary = "0" + binary;
            }
        }
        for (int i = 0; i < binary.length(); i += 3) {
            String threeBits = binary.substring(i, i + 3);
            int octalDigit = Integer.parseInt(threeBits, 2);
            octal += Integer.toString(octalDigit);
        }
        return octal;
    }

    public static String octalToDecimal(String octal) {
        String decimal = "";
        decimal = Long.toString(Long.parseLong(octal, 8));
        return decimal;
    }
}

```