Name: Tejaswini Anil Kamble

Email: teju000kamble@gmail.com

Assignments: Day 11

Task 1: String Operations

Write a method that takes two strings, concatenates them, reverses the result, and then extracts the middle substring of the given length. Ensure your method handles edge cases, such as an empty string or a substring length larger than the concatenated string.

Ans: Source Code

```
🏽 FS_JavaProgramming - DSA_JavaAssignments/src/com/wipro/day11/StringOperations.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
 🗂 + 🔡 😭 i 💝 '♡ i 🗎 i 'v i 😭 '૭' + i 'փ + 💿 + 💁 + 'ஹ + 'ஹ 'ଡ 'ଡ 'ଡ 'ଡ 'ଡ 'ଡ 'ଡ 'ଡ ' 😭 ' 🗁 ' 😭 ' 🗷 ' 😭 ' 💮 + '집 + '♡ + ' 😁 '
🔓 Project Explorer 🗶 📙 🛘 🖟 BFSGraph,java 🚺 TowerofHanoi... 🚺 Dijkstra.java 🚺 StringOpera... 🗶 🏚 NaivePattern... 🚺 KMPAlgorithm... 🚺 RabinKarp,java 🚺 BoyerMoore.java
          🗏 🕏 🎖 🍃 🖇 🛘 1 package com.wipro.day11;

✓ I

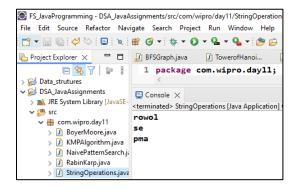
✓ DSA JavaAssignments

                              3 public class StringOperations {
   > A JRE System Library [JavaSE- 4
       public static String concatenateReverseAndExtract(String str1, String str2, int length) {

✓ 

 com.wipro.day11

                                         if (str1 == null) str1 = "";
if (str2 == null) str2 = "";
                                           String concatenated = str1 + str2;
       > ① RabinKarp.java 10
> ① StringOperations.java 11
                                            String reversed = new StringBuilder(concatenated).reverse().toString();
                             10
     > # com.wipro.day20
                                         if (length > reversed.length()) {
     > 🔠 com.wipro.day7and8
                             13
                                                 return reversed;
     > 🌐 com.wipro.generic
                             14
     > # com.wipro.linkedlist
                             15
     > # com.wipro.nonlinear
                                            int start = (reversed.length() - length) / 2;
     > # com.wipro.threads
                              17
                                            return reversed.substring(start, start + length);
     > II module-info.java
                              18
     primes.txt
                              19
 ems
 irstjava 👺
                             20⊝
                                       public static void main(String[] args) {
                                            System.out.println(concatenateReverseAndExtract("hello", "world", 5));
System.out.println(concatenateReverseAndExtract("", "test", 2));
System.out.println(concatenateReverseAndExtract("example", "", 3));
                              21
22
 mvcstruture 🎏
                              24
                              25 }
```



Task 2: Naive Pattern Search

Implement the naive pattern searching algorithm to find all occurrences of a pattern within a given text string. Count the number of comparisons made during the search to evaluate the efficiency of the algorithm.

Ans: Source Code

```
🌉 FS_JavaProgramming - DSA_JavaAssignments/src/com/wipro/day11/NaivePatternSearch.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
🎦 Project Explorer 🗶 📅 🗖 💹 BFSGraph.java 🔃 TowerofHanoi... 📝 Dijkstra.java

☑ StringOpera... ☑ NaivePattern... X ☑ KMPAlge
                        2
        3 public class NaivePatternSearch {
> 📂 Data_strutures
                         4
public static void searchPattern(String text, String pattern) {
  > 🚵 JRE System Library [JavaSE- 📗 5⊖

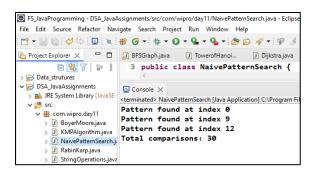
✓ 

Æ src

                                   int n = text.length();

com.wipro.day11

                         7
                                   int m = pattern.length();
     > 🚺 BoyerMoore.java
                        8
                                   int comparisons = 0;
     > 🚺 KMPAlgorithm.java
     > NaivePatternSearch.ji 10
                                  for (int i = 0; i <= n - m; i++) {
     > 🚺 RabinKarp.java
                        11
                                        int j;
     > 🚺 StringOperations.java 12
                                        for (j = 0; j < m; j++) {
   > 🌐 com.wipro.day20
                        13
                                            comparisons++:
   > 🔠 com.wipro.day7and8
                                            if (text.charAt(i + j) != pattern.charAt(j)) {
   > 🌐 com.wipro.generic
                        15
   > 🌐 com.wipro.linkedlist
                        16
   > H com.wipro.nonlinear
                        17
   > 🖶 com.wipro.threads
                                        if (j == m) {
                        18
   > 🚺 module-info.java
                       19
                                            System.out.println("Pattern found at index " + i);
    primes.txt
                        20
> 📂 ems
                        21
                                    }
> 📂 firstjava
                        22
> 📂 mvcstruture
                        23
                                    System.out.println("Total comparisons: " + comparisons);
                        24
                        25
                        26⊖
                                public static void main(String[] args) {
                        27
                                    String text = "AABAACAADAABAABA";
                        28
                                    String pattern = "AABA";
                                    searchPattern(text, pattern);
                        29
                        30
                                }
                        31 }
```



Task 3: Implementing the KMP Algorithm

Code the Knuth-Morris-Pratt (KMP) algorithm in C# for pattern searching which pre-processes the pattern to reduce the number of comparisons. Explain how this pre-processing improves the search time compared to the naive approach.

Ans: Source Code

```
🌉 FS_JavaProgramming - DSA_JavaAssignments/src/com/wipro/day11/KMPAlgorithm.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
 🔓 Project Explorer 🗶 😑 🕝 🔝 BFSGraph.java 🔃 TowerofHanoi... 🗓 Dijkstrajava 🔃 StringOpera... 🖟 NaivePattern... 🖟 KMPAlgorithm... 🗴 🗓 RabinKarp.java 🕡 BoyerMoore.jav
                            | 1 package com.wipro.day11;
strutures | 2 public class KMPAlgorithm | 3 {
  > 🔛 Data_strutures
   DSA JavaAssignments
     > Busing Marketing Straing ( ) See The Straing

    D BoyerMoorejava
    D KMPAlgorithm.java
    NaivePatternSearch.ji
    RabinKarp.java

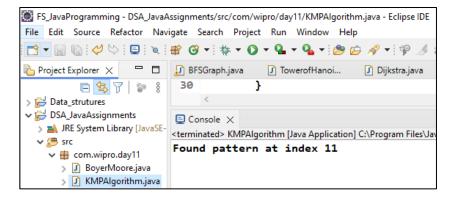
                                                                                                                                 int[] lps = new int[m];
             > ① StringOperations.java
> 冊 com.wipro.day20
> ♣ com.wipro.day7and8
                                                                                                                          computeLPSArray(pattern, m, lps);
                                                                                                                            int i = 0;
while (i < n) {
    if (pattern.charAt(j) == text.charAt(i)) {</pre>
               > # com.wipro.generic
             corn.wipro.generic
com.wipro.linkedlist
com.wipro.nonlinear
com.wipro.nonlinear
module-info.java
     primes.txt

ems

firstjava
                                                                                       19
                                                                                                                                                                System.out.println("Found pattern at index " + (i - j));
     system.out.printin("Found pattern at index " + (1 - j),
    j = lps[j - 1];
} else if (i < n && pattern.charAt(j) != text.charAt(i)) {
    if (j != 0) {
        j = lps[j - 1];
    } else {</pre>
                                                                                        24
                                                                                        27
28
                                                                                        29
                                                                                                              }
                                                                                                                                             }
```

```
FS JavaProgramming - DSA JavaAssignments/src/com/wipro/day11/KMPAlgorithm.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X TowerofHanoi... P Dijkstra.java P StringOpera... NaivePattern... KMPAlgorithm... X P RabinKarp.java P BoyerMoore.java
        □ 🔄 🎖 👺 🖇 30
}
                                   private static void computeLPSArray(String pattern, int m, int[] lps) {
   > N JRE System Library [JavaSE- 33®
 lps[0] = 0;
                                              if (pattern.charAt(i) == pattern.charAt(len)) {
                                                   len++;
lps[i] = len;
    > # com.wipro.day20
> # com.wipro.day7and8
> # com.wipro.generic
> # com.wipro.linkedlist
                                                 if (len != 0) {
    len = lps[len - 1];
} else {
                            44
    > # com.wipro.nonlinear
> # com.wipro.threads
> I module-info.java
                                                      lps[i] = len;
i++;
                            47
    primes.txt
  firstjava
mvcstruture
                            49
                            50
51
52
                                             }
                                       }
                                    }
                           53
54⊝
                                    public static void main(String[] args) {
   String text = "WELCOMEINTOWIPRRO";
   String pattern = "WIPRRO";
                                         KMPSearch(pattern, text);
                                    }
```

Output:

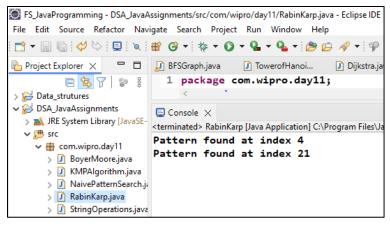


Task 4: Rabin-Karp Substring Search

Implement the Rabin-Karp algorithm for substring search using a rolling hash. Discuss the impact of hash collisions on the algorithm's performance and how to handle them.

Ans: Source Code

```
🌉 FS_JavaProgramming - DSA_JavaAssignments/src/com/wipro/day11/RabinKarp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
 🗂 ㅜ 🖫 🕼 | 🗳 ♡ | 🖳 | 🐿 🖒 | 🖶 🎯 ㅜ | 🐎 ㅜ 🕥 ㅜ 🤽 ㅜ 😘 ㅜ 😭 ㅜ 😭 🌽 🍪 😥 🍿 🖠 🕬 📵 🔳 👚 : [위 ㅜ 🙌 ㅜ 🌣 ㅜ 🗘 ㅜ 🖒 ㅜ 🗘
🏠 Project Explorer 🗶 📅 📗 BFSGraph.java 📝 TowerofHanoi... 📝 Dijkstra.java 📝 StringOpera... 📝 NaivePattern... 📝 KMPAlgorithm...
        3 public class RabinKarp {
> ■ JRE System Library [JavaSE-
                               public static void search(String pattern, String text, int prime) {
                                   int m = pattern.length();
    com.wipro.day11
                                    int n = text.length();int p = 0; int t = 0;int h = 1;
      > D BoyerMoore.java
                                   int d = 256;
      > MPAlgorithm.java
                                   for (int i = 0; i < m - 1; i++) {
      > 🚺 NaivePatternSearch.ja 🥊 9
                                         h = (h * d) % prime;
     > 🚺 RabinKarp.java
                                    for (int i = 0; i < m; i++) {
      > I StringOperations.java
                                        p = (d * p + pattern.charAt(i)) % prime;
t = (d * t + text.charAt(i)) % prime;}
    > # com.wipro.day20
    > 🔠 com.wipro.day7and8
                                     for (int i = 0; i <= n - m; i++) {
    > 🌐 com.wipro.generic
                         14
                                         if (p == t) {
    > 🌐 com.wipro.linkedlist
                         15
                                              boolean match = true;
    > H com.wipro.nonlinear
                         16
                                              for (int j = 0; j < m; j++) {
    > # com.wipro.threads
                         17
                                                  if (text.charAt(i + j) != pattern.charAt(j)) {
    > 🚺 module-info.java
                         18
                                                      match = false;
    primes.txt
                         19
                                                       break;} }if (match) {
 > 📂 ems
                                                  System.out.println("Pattern found at index " + i); } }
 > 📂 firstjava
                                          if (i < n - m) {
 > 📂 mvcstruture
                                              t = (d * (t - text.charAt(i) * h) + text.charAt(i + m)) % prime;
                         22
                         23
                         24
                                              if (t < 0) {
                         25
                                                  t = (t + prime);}}}}
                         26<sup>©</sup> public static void main(String[] args) {
                                String text = "TEJASWINI KAMBLE TEJASWINI";
                         27
                         28
                                     String pattern = "SWINI";
                         29
                                    int prime = 101;
                         30
                                     search(pattern, text, prime); }}
```



Task 5: Boyer-Moore Algorithm Application

Use the Boyer-Moore algorithm to write a function that finds the last occurrence of a substring in a given string and returns its index. Explain why this algorithm can outperform others in certain scenarios.

Ans: Source Code

```
FS JavaProgramming - DSA JavaAssignments/src/com/wipro/day11/BoyerMoore.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
 🔓 Project Explorer 🗴 🖳 🗖 📗 BFSGraph, java 🗓 Towerof Hanoi... 📗 Dijkstra java 📗 String Opera... 📗 Naive Pattern... 📗 KMPAlgorithm... 📗 Rabin Karp, java 🗓 Boyer Moore java 🗴
        > 📂 Data_strutures
3 public class BoyerMoore {
  > 

■ JRE System Library [JavaSE- 4
                                 private static int[] buildBadCharTable(String pattern) {

✓ 

 com.wipro.day11

                                     int[] badChar = new int[256];
for (int i = 0; i < 256; i++) {</pre>
                          6
      > D BoyerMoore.java
      badChar[i] = -1;
      > RabinKarp.java
                                     for (int i = 0; i < pattern.length(); i++) {
   badChar[(int) pattern.charAt(i)] = i;</pre>
      > I RabinKarp java
> I StringOperations java
    > 🌐 com.wipro.day20
    > 🔠 com.wipro.day7and8
                         13
                                      return badChar:
                         14
                                 }
    > # com.wipro.linkedlist
    > 🌐 com.wipro.nonlinear
                         16⊖
                                 public static int findLastOccurrence(String text, String pattern) {
    > # com.wipro.threads
     module-info.java
                         17
                                     int m = pattern.length();
                                      int n = text.length();
    primes.txt
 > 📂 ems
                          19
                                      int[] badChar = buildBadCharTable(pattern);
 irstjava 🔐
                         20
                                      int shift = 0;
                                     int lastOccurrence = -1;
 > 📂 mvcstruture
                         21
                          23
                                     while (shift \leftarrow (n - m)) {
                          24
                                          int j = m - 1;
                          26
                                          while (j >= 0 && pattern.charAt(j) == text.charAt(shift + j)) {
                          27
                          29
                          30
                                          if (j < 0) {
                                               lastOccurrence = shift;
```

```
✓ 

    com.wipro.day11

     > D BoyerMoore.java
                          29
     >  KMPAlgorithm.java 30
>  NaivePatternSearch.ji 31
                                           if (j < 0) {
                                                lastOccurrence = shift;
     > 🚺 RabinKarp.java
                          32
                                                shift += (shift + m < n) ? m - badChar[text.charAt(shift + m)] : 1;</pre>
     > 🚺 StringOperations.java
                          33
                                           } else {
   > # com.wipro.day20
                                               shift += Math.max(1, j - badChar[text.charAt(shift + j)]);
  > 🔠 com.wipro.day7and8
  > # com.wipro.generic
> # com.wipro.linkedlist
                          36
                                      }
                          37
   > # com.wipro.nonlinear
                                       return lastOccurrence;
                         38
   > # com.wipro.threads
                         39
                                 }
   > 1 module-info.java
                          40
  primes.txt
                          419
                                  public static void main(String[] args) {
ems
irstjava 😽
                          42
                                       String text = "ABAAABCD";
                                       String pattern = "ABC";
                          43
mvcstruture
                          45
                                       int index = findLastOccurrence(text, pattern);
                          46
                                       if (index != -1) {
                                           System.out.println("Last occurrence of pattern found at index " + index);
                          47
                          48
                                       } else {
                          49
                                           System.out.println("Pattern not found");
                          50
                          51
                                  }
                          52 }
```

