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.

Code-:

```
package com.wipro.assignment;
class BoyerMoore
{
   static int NO_OF_CHARS = 256;
   static int max(int a, int b) {
return (a > b) ? a : b; }
   static void
badCharHeuristic(char[] str, int
size,
         int badchar[])
   {
      for (int i = 0; i <
NO_OF_CHARS; i++)
         badchar[i] = -1;
      for (int i = 0; i < size;
i++)
```

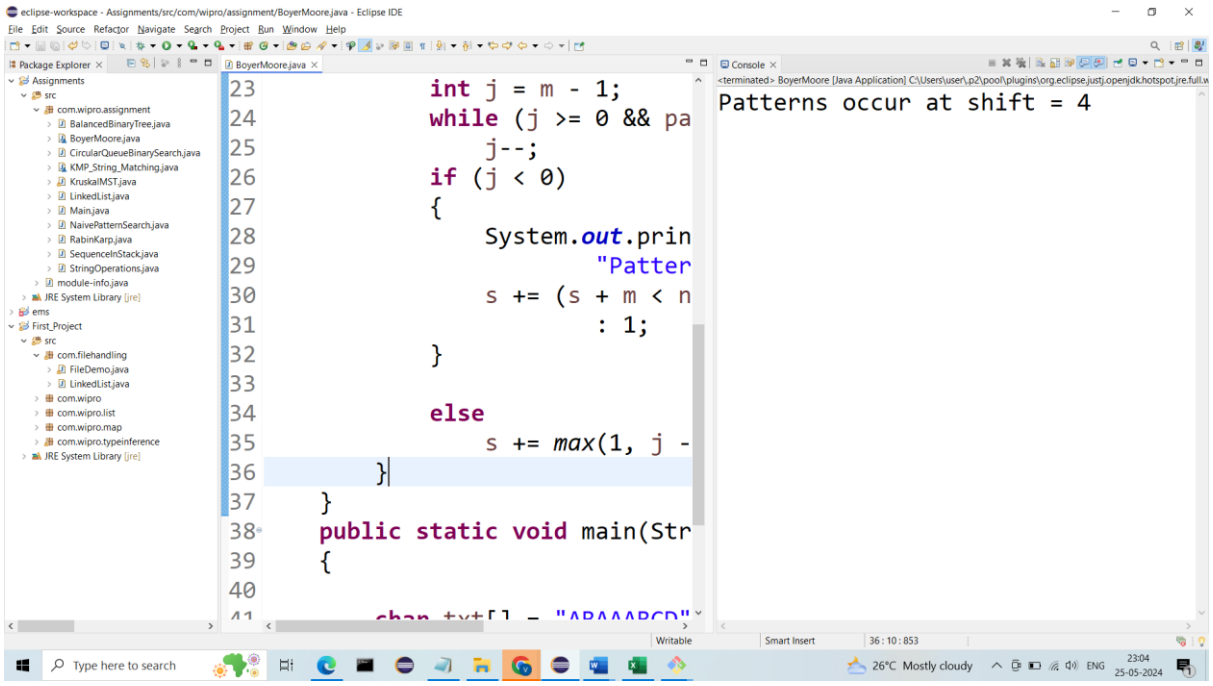
```
            badchar[(int)str[i]] = i;
    }
    static void search(char txt[],
char pat[])
    {
        int m = pat.length;
        int n = txt.length;
        int badchar[] = new
int[NO_OF_CHARS];
        badCharHeuristic(pat, m,
badchar);
        int s = 0;
        while (s <= (n - m))
        {
            int j = m - 1;
            while (j >= 0 && pat[j] ==
txt[s + j])
                j--;
            if (j < 0)
            {
                System.out.println(
                        "Patterns occur
at shift = " + s);
```

```
            s += (s + m < n) ? m -
badchar[txt[s + m]]
                    : 1;
        }

        else
            s += max(1, j -
badchar[txt[s + j]]);
      }
   }
   public static void
main(String[] args)
   {

      char txt[] =
"ABAAABCD".toCharArray();
      char pat[] =
"ABC".toCharArray();
      search(txt, pat);
   }
}
```

Output: -


eclipse-workspace - Assignments/src/com/wipro/assignment/BoyerMoore.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Assignments
src
com.wipro.assignment
BalancedBinaryTree.java
BoyerMoore.java
CircularQueueBinarySearch.java
KMP_String_Matching.java
KruskalMST.java
LinkedList.java
Main.java
NaivePatternSearch.java
RabinKarp.java
SequenceInStack.java
StringOperations.java
module-info.java
JRE System Library [jre]
ems
First_Project
src
com.filehandling
FileDemo.java
LinkedList.java
com.wipro
com.wipro.list
com.wipro.map
com.wipro.typeinference
JRE System Library [jre]
BoyerMoore.java
int j = m - 1;
while (j >= 0 && pa
j--;
if (j < 0)
{
System.out.prin
"Patter
s += (s + m < n
: 1;
}
else
s += max(1, j -
}
}
public static void main(Str
{
Console
<terminated> BoyerMoore [Java Application] C:\Users\user\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w
Patterns occur at shift = 4
Writable
Smart Insert
36 : 10 : 853
Type here to search
26°C Mostly cloudy
ENG
23:04
25-05-2024