DESIGN ANALYSIS AND ALGORITHMS

LAB ASSIGNMENT-9

NAME: PRATHAPANI SATWIKA

REG.NO.: 20BCD7160

Write a program that uses Branch and bound algorithm to solve the String Matching problem

CODE :

```
package Lab9;
import java.util.*;
public class StringMatchingBranchandBound{
	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();
```

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

```java
package Lab9;
import java.util.*;
public class StringMatchingBranchandBound{
    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 :

```
<terminated> StringMatchingBranchandBound [Java Application] C:\Program Files\J
Patterns occur at shift = 4
```