Analysis of Algorithm

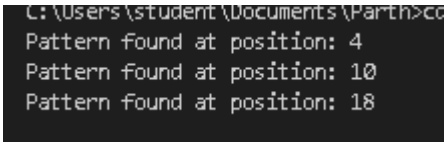
Practical no 11 : String Matching

Code :

```
public class NaiveStringMatching {
  public static void naiveFindPatrn(String mainString, String pattern, int[] array) {
    int patLen = pattern.length();
    int strLen = mainString.length();
    int index = 0;
    // outer for loop
    for(int i = 0; i <= (strLen - patLen); i++) {
      int j;
      // to check for each character of pattern
      for(j = 0; j < patLen; j++) {
        if(mainString.charAt(i+j) != pattern.charAt(j))
          break;
      }
      // to print the index of the pattern is found
      if(j == patLen) {
        array[index] = i;
        index++;
      }
    }
  }
  // main method starts
  public static void main(String[] args) {
    // main string
    String mainString = "ABAAABCDBBABCDDEBCABC";
    // pattern to be found
    String pattern = "ABC";
    int[] locArray = new int[mainString.length()];
    naiveFindPatrn(mainString, pattern, locArray);
```

```
    // to print the indices
    for(int i = 0; i < locArray.length && locArray[i] != 0; i++) {
      System.out.println("Pattern found at position: " + locArray[i]);
    }
  }
}
```

Output :



Analysis :

naiveFindPatrn Method:

- Parameters:
  - mainString: The string in which the pattern is being searched.
  - pattern: The pattern that is being searched for in the mainString.
  - array: An array that stores the indices where the pattern is found.
- Outer Loop (Line 10):
  - This loop iterates over the mainString from i = 0 to i = strLen - patLen to check for a potential match for the pattern starting at each position in the main string.
- Inner Loop (Line 14):
  - This loop compares each character of the mainString starting from index i to each character of the pattern.
  - If a mismatch occurs, the inner loop breaks, and the outer loop moves to the next position in the main string.
- Pattern Matching Check (Line 20):
  - If the inner loop successfully compares all characters of the pattern with the substring of mainString, the pattern is found, and its starting index i is added to the array.

` main Method (Line 23):

- This method initializes the main string and the pattern.
- It calls naiveFindPatrn to search for the pattern in the mainString.
- It prints the indices where the pattern is found.

Summary of Complexities:

- Time Complexity:

    - Worst case: O(n×m)O(n \times m)O(n×m)
    - Best case: O(n×m)O(n \times m)O(n×m)
    - Average case: O(n×m)O(n \times m)O(n×m)
- Space Complexity:
    - O(n)O(n)O(n)