**Maher Salman**

**Naive Algorithm**

The naive algorithm is a straightforward approach to pattern matching. It involves comparing each possible substring of the text with the pattern to find all occurrences. This method is simple but can be inefficient for large texts due to its time complexity.

**MCS (Minimum Common Subsequence) Algorithm**

The MCS algorithm is designed to improve efficiency in pattern matching by leveraging a precomputed set of patterns (lossy MCSs). These patterns are generated to maximize overlap and minimize redundant comparisons, making the search process more efficient.

**Comparison between Naive and MCS Algorithms**

**Efficiency**

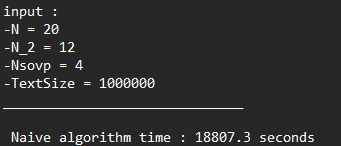
* **Naive Algorithm**: The time complexity of the naive algorithm is O(n⋅m), where n is the length of the text and m is the length of the pattern. This quadratic complexity can be slow for large texts.
* **MCS Algorithm**: The MCS algorithm improves efficiency by reducing the number of comparisons needed. The precomputation of lossy MCSs adds an initial overhead, but the search process itself is faster.

**Memory Usage**

* **Naive Algorithm**: Memory usage is minimal, as it primarily involves storing the text and pattern.
* **MCS Algorithm**: Requires additional memory for storing the lossy MCSs and their associated data structures, leading to higher memory usage compared to the naive approach.

**Output :(on alphabet size = 15 )**

**Naive algorithm :**

****

**A screenshot of a computer

Description automatically generatedSearch Based On lossy MCSs :**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Explanation :**

**Searching On text size 1Mb , naive take 18807 second = 5.2Hours**

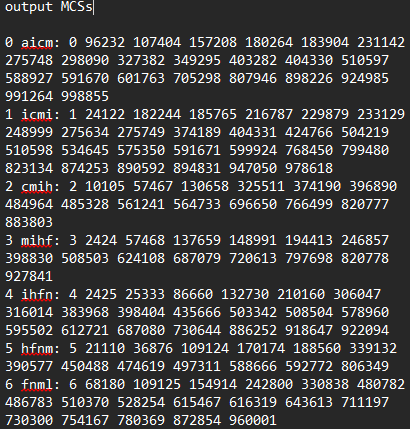
**On the other hand , using MCS take :**

* **Nsovp = 4 - > 7 second**
* **Nsovp = 5 - > 59 second = 1 minute**
* **Nsovp = 6 - > 257 second = 4 minute**

**the conclusion MCSs with Nsovp smaller we get it to be faster .**

**with Nsovp = 6 we get algorithm faster 73 times from naive algorithm.**

* **To check output we take first output using Nsovp = 4**

**A screen shot of a computer

Description automatically generated**

**Using**

**A screen shot of a computer

Description automatically generated**

**According to the output we can see that we find the same output , also using the text indexes we can check all MCSs output to see that the output is correct . and we found all possible strings according to the lossy MCSs.**

**Summary**

* The naive algorithm is simple and easy to implement but can be inefficient for large texts due to its quadratic time complexity.
* In contrast, the MCS algorithm, though more complex and requiring additional memory, is significantly faster due to its structured approach and efficient filtering. This makes the MCS algorithm more effective in handling larger datasets and more suitable for large-scale pattern matching tasks where efficiency is critical.