**Tìm kiếm mẫu từ bên trái qua bên phải:** Harrison Algorithm, Karp-Rabin Algorithm, Morris-Pratt Algorithm, Knuth- Morris-Pratt Algorithm, Forward Dawg Matching algorithm , Apostolico-Crochemore algorithm, Naive algorithm.

**Tìm kiếm mẫu từ bên phải qua bên trái**: Boyer-Moore Algorithm , Turbo BM Algorithm, Colussi Algorithm, Sunday Algorithm, Reverse Factorand Algorithm, Turbo Reverse Factor, Zhu and Takaoka and Berry-Ravindran Algorithms.

**Tìm kiếm mẫu từ một vị trí cụ thể:** Two Way Algorithm, Colussi Algorithm , Galil-Giancarlo Algorithm, Sunday's Optimal Mismatch  Algorithm, Maximal Shift Algorithm, Skip Search, KMP Skip Search and Alpha Skip Search Algorithms.

**Tìm kiếm mẫu từ bất kỳ**: Horspool Algorithm, Boyer-Moore Algorithm, Smith Algorithm , Raita Algorithm.

# Tìm kiếm mẫu từ trái sang phải:

1. Knuth-Morris-Pratt (KMP):

**Trình bày thuật toán Knuth-Morris-Pratt (KMP):**

1. **Khởi tạo bảng lùi (failure function):**
   * Tạo một bảng lùi (còn được gọi là bảng pi) với độ dài bằng với độ dài của mẫu cần tìm kiếm.
   * Bắt đầu từ đầu của mẫu, di chuyển từng ký tự và tính toán giá trị của bảng lùi dựa trên ký tự trước đó.
2. **Tìm kiếm mẫu trong văn bản:**
   * Di chuyển từ đầu văn bản đến cuối văn bản, so sánh từng ký tự của văn bản với từng ký tự của mẫu.
   * Nếu có sự khác biệt, sử dụng bảng lùi để xác định vị trí tiếp theo trong mẫu cần so sánh.
   * Lặp lại quá trình này cho đến khi tìm thấy một bản sao của mẫu trong văn bản hoặc cho đến khi văn bản được duyệt qua hoàn toàn.

**Đánh giá độ phức tạp thuật toán:**

* Thời gian: O(n + m), trong đó n là độ dài của văn bản và m là độ dài của mẫu.
* Không gian: O(m), với m là độ dài của mẫu.

**Kiểm nghiệm thuật toán:**

* Kiểm tra kết quả của thuật toán với các bộ dữ liệu kiểm thử khác nhau, bao gồm các trường hợp tìm thấy mẫu trong văn bản và không tìm thấy.
* So sánh kết quả với kết quả mong đợi.

