Algorytmy i struktury danych

Sprawozdanie z Zadania nr: 5

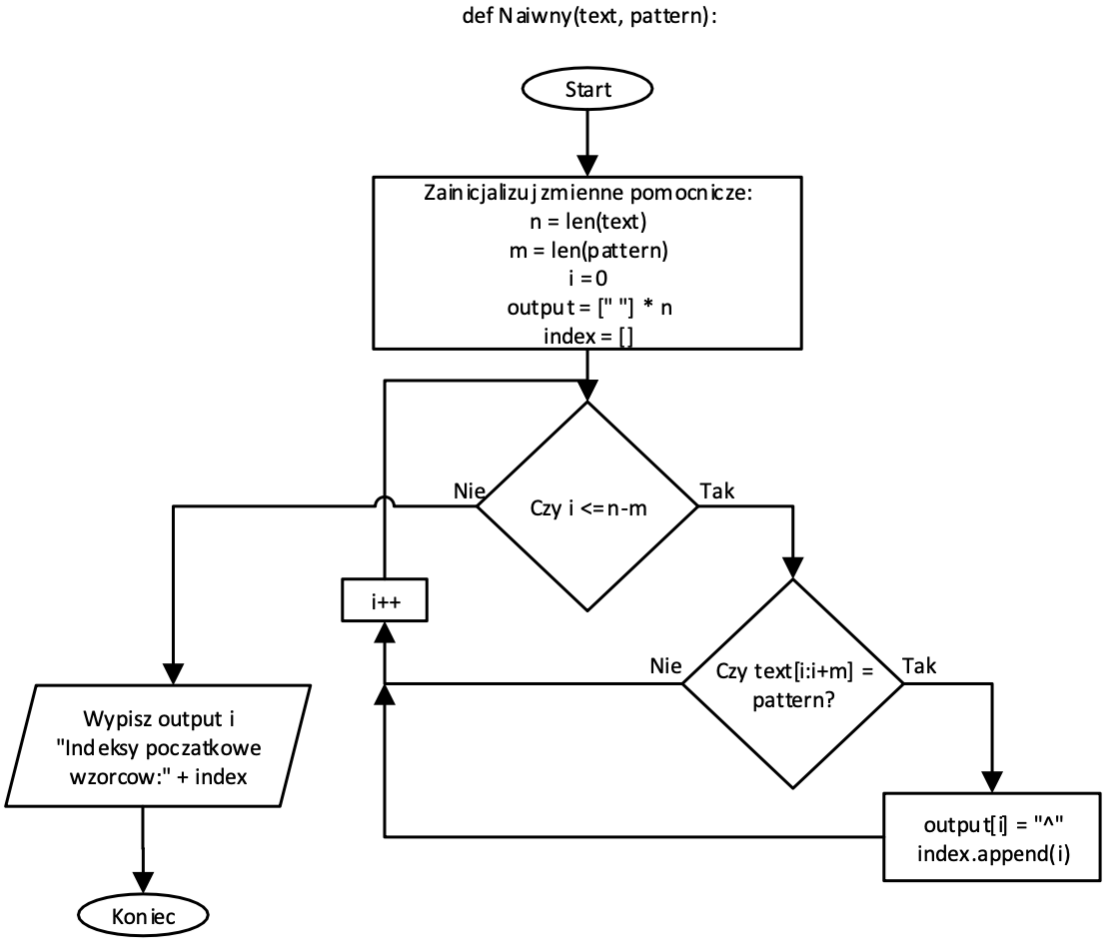
Imię i Nazwisko: Bartosz Ochnik

Data: 27.11.2024r

Naiwny

Opis teoretyczny: Działanie algorytmu naiwnego polega na przesuwaniu wzorca po tekście znak po znaku i porównywaniu go z każdym możliwym fragmentem tekstu o takiej samej długości. Algorytm nie wykorzystuje żadnych informacji o strukturze tekstu ani wzorca, przez co jego złożoność czasowa wynosi O(m*n), gdzie mm to długość wzorca, a n długość tekstu. Jest prosty do zaimplementowania, ale nieefektywny przy dużych danych.

Opis schematem blokowym:



Przykłady wykorzystania:

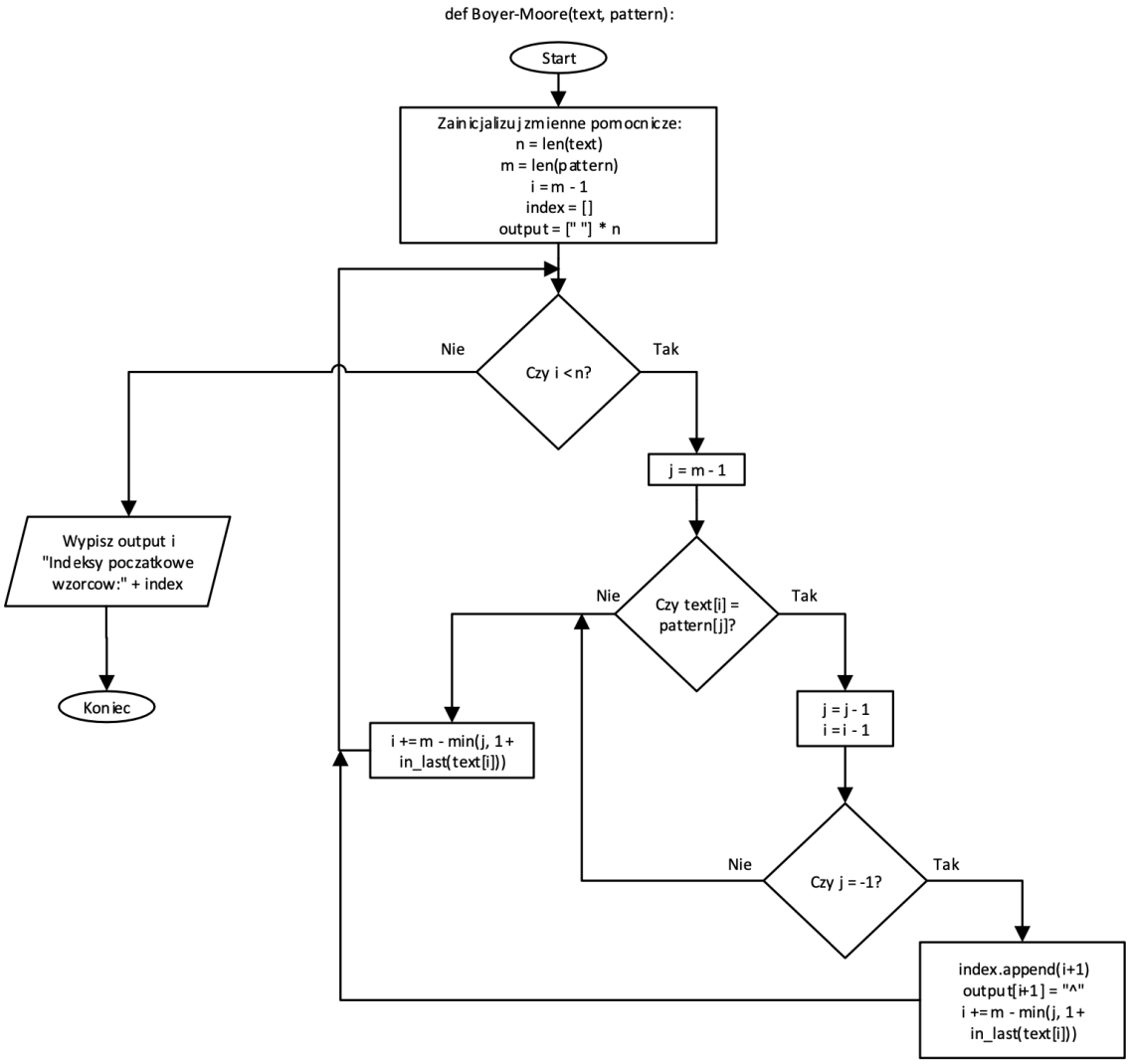
- Wyszukiwanie prostych fraz w plikach tekstowych.
- Podstawowy algorytm edukacyjny do nauki przetwarzania tekstu.
- Analiza krótkich logów w systemach monitorowania aplikacji.

Algorytmy i struktury danych

Boyer-Moore

Opis teoretyczny: Algorytm Boyera-Moore'a jest bardzo wydajny w praktyce dzięki wykorzystaniu informacji o wzorcu do przyspieszenia wyszukiwania. Porównuje wzorzec z tekstem od prawej strony i używa dwóch heurystyk (złego znaku oraz dobrego sufiksu), aby przeskakiwać po tekście w przypadku niedopasowań. Jego złożoność wynosi O(n/m) w najlepszym przypadku, a w najgorszym O(n*m).

Opis schematem blokowym:



Przykłady wykorzystania:

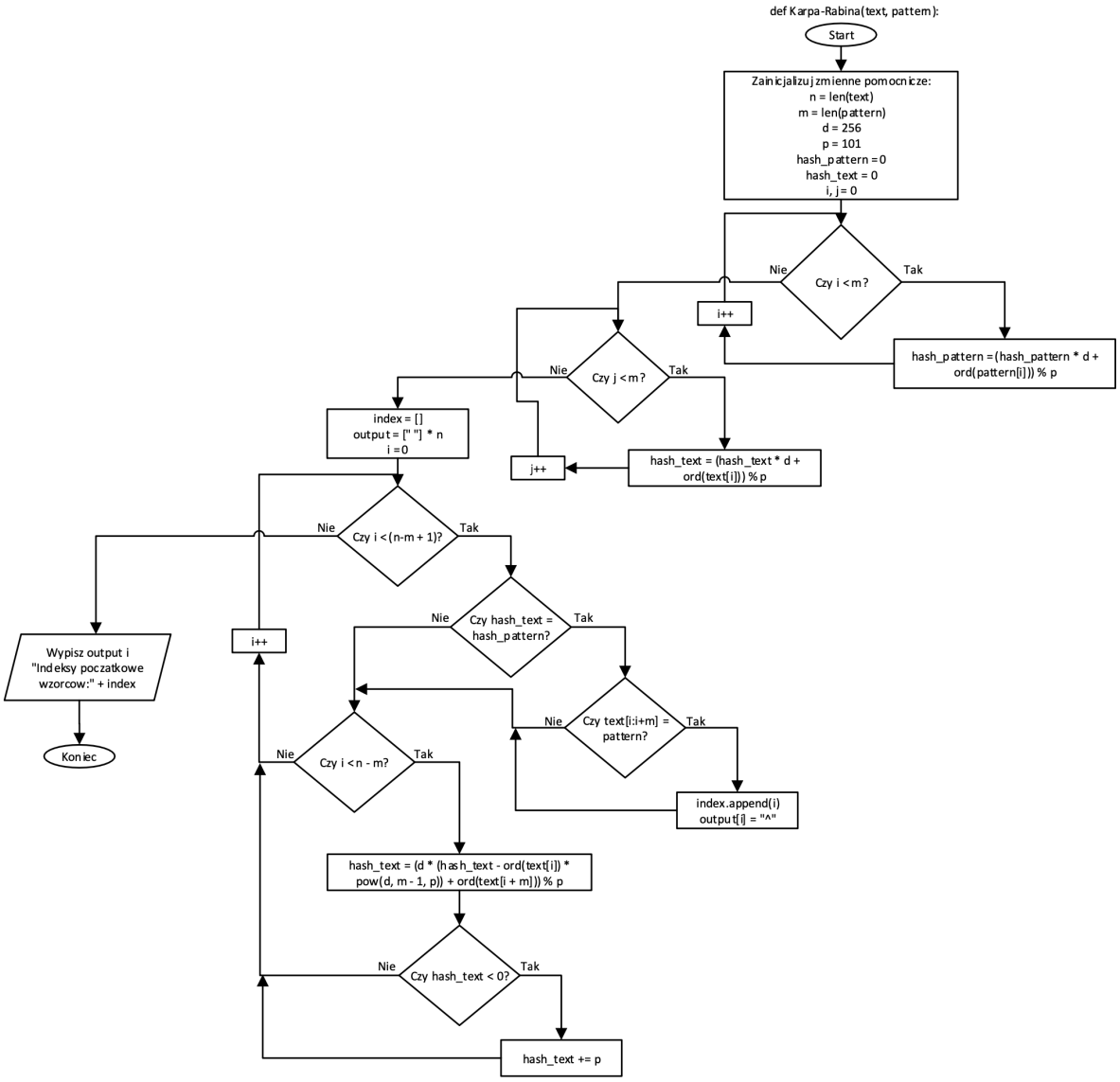
- Wyszukiwanie tekstów w dużych bazach danych (np. SQL).
- Analiza i kompresja danych, np. w algorytmach typu gzip.
- Wykrywanie sekwencji w systemach monitorowania bezpieczeństwa, np. w IDS/IPS.

Algorytmy i struktury danych

Karpa - Rabina

Opis teoretyczny: Algorytm Karpa-Rabina wykorzystuje funkcję haszującą, aby szybciej identyfikować potencjalne dopasowania wzorca w tekście. Najpierw oblicza hash wzorca, a następnie porównuje go z hashami kolejnych fragmentów tekstu o tej samej długości. Jeśli hashe są identyczne, sprawdza zgodność znaków. Dzięki tej strategii złożoność czasowa wynosi średnio O(n+m), ale w najgorszym przypadku O(n*m), zależnie od funkcji haszującej.

Opis schematem blokowym:



Przykłady wykorzystania:

- Weryfikacja integralności danych w systemach przechowywania plików.
- Wykrywanie plagiatów w dużych zbiorach tekstów.
- Wyszukiwanie wzorców w genomice, np. sekwencji DNA.