ALGORYTM SUNDAYA

OPIS I ZAŁOŻENIA

Algorytm służy do znajdowania wzorca w tekście.

Algorytm tworzy tabelę pomocniczą indeksowaną literami, dzięki czemu w czasie stałym możemy stwierdzić czy litera znajduje się w naszym wzorcu i na której pozycji znajduje się jej ostatnie wystąpienie. Następnie algorytm porównuje znaki we wzorcu ze znakami w tekście. Jeśli napotka niezgodność, nie sprawdza kolejnych znaków. Funkcja sprawdza następny znak z naszego tekstu, który znajduje się tuż za oknem wyszukiwania. Jeśli ten znak się nie zgadza ze znakiem w naszym wzorcu w to okno wyszukiwania przeskakuje o długość wzorca +1. Jeśli natomiast się zgadza, program mający informację z pomocniczej tablicy, przesuwa wzorzec (ostatnie wystąpienie tego znaku we wzorcu) do odpowiedniego znaku w naszym tekście i porównuje po kolei znaki.

Rząd czasowej złożoności obliczeniowej:

- Pesymistyczna: długość łańcucha * długość wzorca
- Optymistyczna: długość łańcucha/(długość wzorca +1)

PRZYKŁAD

Weźmy za przykład:

Łańcuch znaków: abcabbdcabcababcdaca

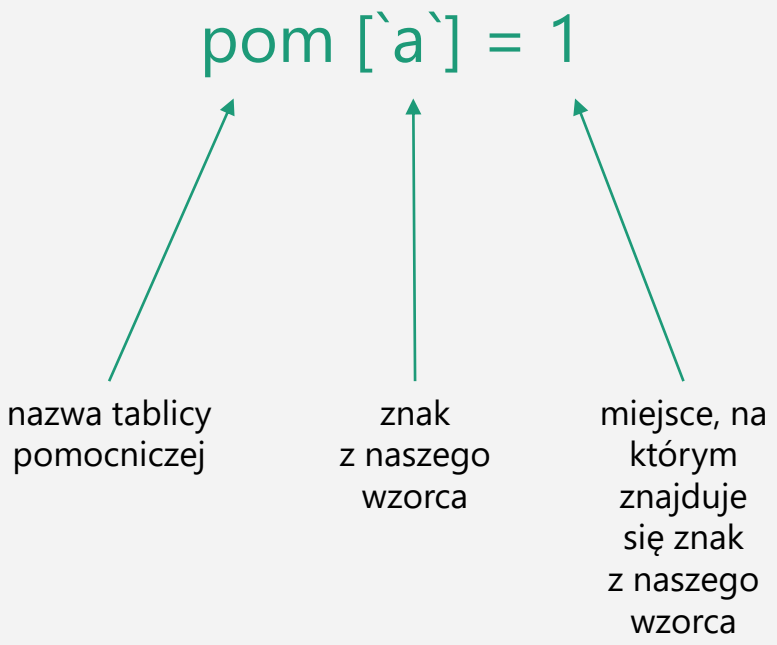
Wzorzec: aba

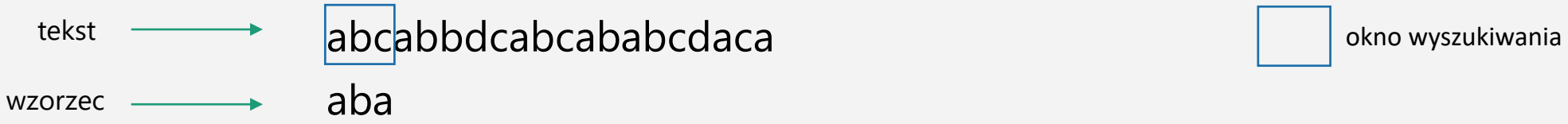
Nasz algorytm tworzy tablicę pomocniczą dla wzorca, indeksowaną literami i zapisującymi w niej pozycję, na której znajdują się, nasze znaki.

Dla naszego przykładu:

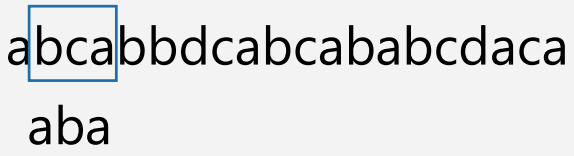
pom [`a`] = 2

pom [`b`] = 1



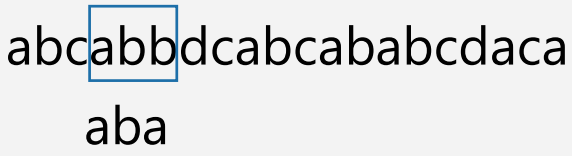


- Algorytm sprawdza znaki z naszego okna wyszukiwania ze znakami z naszego wzorca,
- Widzimy, że znaki na pozycji 0 się zgadzają, przechodzimy do kolejnego znaku,
- Widzimy, że znaki na pozycji 1 się zgadzają, przechodzimy do kolejnego znaku,
- Widzimy, że znaki na pozycji 2 nie są zgodne,
- Algorytm sprawdza czy pierwszy znak za oknem wyszukiwania znajduje się w naszym wzorcu,
- Widzimy, że znak a występuje w naszym wzorcu,



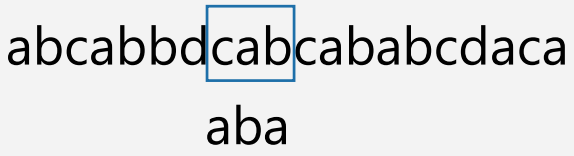
okno wyszukiwania

- Następuje takie przesunięcie, by znak a w tekście pokrył się z ostatnim jego wystąpieniem we wzorcu,
- Widzimy, że znaki a oraz znak b się nie zgadza, jedyna zgodność występuje na ostatniej pozycji,
- Algorytm sprawdza czy kolejny znak za oknem wyszukiwania znajduje się w naszym wzorcu, widzimy, że się znajduje, więc następuje takie przesunięcie, by znak b w tekście pokrył się z ostatnim jego wystąpieniem we wzorcu,



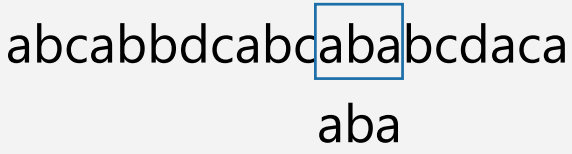
okno wyszukiwania

- Po przesunięciu algorytm sprawdza znaki, widzimy, że znaki a i znaki b się zgadzają, jedyna niezgodność występuje na ostatniej pozycji,
- Algorytm sprawdza wystąpienie we wzorcu kolejnego znaku za oknem wyszukiwania,
- Widzimy, że znaku d nie ma w naszym wzorcu



okno wyszukiwania

- Algorytm przesuwa nasz wzorzec o długość naszego wzorca + 1,
- Widzimy, że nie ma zgodności znaków na żadnej pozycji,
- Algorytm sprawdza kolejny znak za oknem wyszukiwania,
- Widzimy, że znak za oknem wyszukiwania nie znajduje się w naszym wzorcu



okno wyszukiwania

- Algorytm przesuwa nasz wzorzec o długość naszego wzorca + 1,
- Algorytm sprawdza nasze znaki,
- Widzimy, że występuje zgodność na wszystkich pozycjach,
- Algorytm kończy pracę

Opracowano przez Aleksandra Sadowskiego, z pomocą dr inż. Piotra Belinga