

A

ALGORYTM

NAIWNY



ALGORYTM TEN JEST **NAJPROSTSZĄ METODĄ** NA ZNALEZIENIE WZORCA W TEKŚCIE. ZASADA DZIAŁANIA OPIERA SIĘ NA **PORÓWNYWANIU ODPOWIEDNICH LITER TEKSTU I WZORCA**, ZACZYNAJĄC OD PIERWSZEJ LITERY TEKSTU I WZORCA. **JEŚLI FRAGMENTY TEKSTU SĄ ZGODNE, PORÓWNYWANY JEST NASTĘPNY ZNAK TEKSTU Z NASTĘPNYM ZNAKIEM WZORCA** ITD. JEŚLI WYSTĄPIŁA **NIEZGODNOŚĆ** W DOWOLNYM MIEJSCU – ALGORYTM ROZPOCZYNA CAŁĄ PROCEDURĘ PRZESZUKIWANIA **OD DRUGIEGO ZNAKU TEKSTU I PIERWSZEGO ZNAKU WZORCA**. JEŚLI ZOSTANIE ZNALEZIONY CAŁY SZUKANY WZORZEC, ALGORYTM ROZPOCZYNA PRZESZUKIWANIE OD NASTĘPNEGO ZNAKU. INNYMI SŁOWY, **WZORZEC JEST ZAWSZE „PRZESUWANY” O JEDEN ZNAK**. JEST TO PODSTAWOWA WADA TEGO ALGORYTMU.

A A C B A A B T E K S T

A A B W Z O R Z E C

A A C B A A B

A A B

Przeszukiwanie rozpoczyna się od pierwszego znaku tekstu i pierwszego znaku wzorca.

Ponieważ $C \neq B$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 2giej litery tekstu:

A A C B A A B

A A B

Ponieważ $C \neq A$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 3ciej litery tekstu:

A A C B A A B

A A B

Ponieważ $C \neq A$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 4tej litery tekstu:

A A C B A A B

A A B

Ponieważ $B \neq A$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 5tej litery tekstu:

A A C B A A B

A A B

Ponieważ wszystkie 3 litery są zgodne, więc został odnaleziony poszukiwany ciąg.

A

ALGORYTM

SUNDAYA

ALGORYTM TEN JEST **SZYBSZĄ METODĄ** NA ZNALEZIENIE WZORCA W TEKŚCIE.

ALGORYTM TWORZY **TABLICĘ POMOCNICZĄ INDEKSOWANĄ LITERAMI**, KTÓRA MÓWI NAM O TYM, CZY LITERA ZNAJDUJE SIĘ W NASZYM WZORCU I NA KTÓREJ POZYCJI ZNAJDUJE SIĘ JEJ OSTATNIE WYSTĄPIENIE.

ZASADA DZIAŁANIA OPIERA SIĘ NA PORÓWNYWANIU ODPOWIEDNIICH LITER TEKSTU I WZORCA, ZACZYNAJĄC OD PIERWSZEJ LITERY TEKSTU I WZORCA.

JEŚLI FRAGMENTY TEKSTU SĄ ZGODNE, PORÓWNYWANY JEST NASTĘPNY ZNAK TEKSTU Z NASTĘPNYM ZNAKIEM WZORCA ITD. JEŚLI WYSTĄPIŁA NIEZGODNOŚĆ W DOWOLNYM MIEJSCU – ALGORYTM NAJPIERW SPRAWDZA NASTĘPNY ZNAK TEKSTU, KTÓRY ZNAJDUJE SIĘ TUŻ ZA OKNEM WYSZUKIWANIA.

JEŚLI ZNAK TEKSTU, KTÓRY ZNAJDUJE SIĘ TUŻ ZA OKNEM WYSZUKIWANIA NIE WYSTĘPUJE WE WZORCU, TO OKNO WZORCA MOŻEMY OD RAZU PRZESUNĄĆ O TYLE POZYCJI, ILE ZNAKÓW ZAWIERA WZORZEC +1.

JEŚLI NATOMIAST ZNAK TEKSTU, KTÓRY ZNAJDUJE SIĘ TUŻ ZA OKNEM WYSZUKIWANIA WYSTĘPUJE WE WZORCU, PROGRAM MAJĄCY INFORMACJĘ Z POMOCNICZEJ TABLICY, PRZESUWA WZORZEC(OSTATNIE WYSTĄPIENIE TEGO ZNAKU WE WZORCU) DO ODPOWIEDNIEGO ZNAKU W NASZYM TEKŚCIE

(TZN. WZORZEC POZYCJONUJEMY TAK, ABY ZGRAĆ POZYCJE ZNAKU WYSTĘPUJĄCEGO JEDNOCZEŚNIE W PRZESZUKIWANYM TEKŚCIE I WE WZORCU)

I PO KOLEI PORÓWNUJE ZNAKI.

Tab ['B'] = 1

NAZWA TABLICY
POMOCNICZEJ

ZNAK
ZNAJDUJĄCY SIĘ
W NASZYM
WZORCU

MIEJSCE,
NA KTÓRYM
ZNAJDUJE SIĘ
ZNAK
Z NASZEGO
WZORCA

PODOBIENSTWA

- WYSZUKIWANIE WZORCA W ŁAŃCUCHU TEKSTOWYM
- RZĄD CZASOWEJ ZŁOŻONOŚCI OBLICZENIOWEJ: PESYMISTYCZNA: $O(N \times M)$
- ALGORYTM NAIWNY JAK I ALGORYTM SUNDAYA **WYKONUJE WIELE POWTARZAJĄCYCH SIĘ PRZESZUKIWAŃ**, NIE WYKORZYSTUJĄC W PRZYPADKU NIEZGODNOŚCI KTÓREJŚ Z LITER WZORCA INFORMACJI WCZEŚNIEJ PRZEBADANEJ (NP. DLA TEKSTU 'ABAAAABC' I WZORCA 'ABAAAC' MARNUJE SIĘ PRZY PIERWSZEJ KOLIZJI 6 PORÓWNAŃ).

RÓŻNICE

- ALGORYTM SUNDAYA ZWYKLE **DUŻO SZYBCIEJ** DOCHODZI DO ROZWIĄZANIA
- W **ALGORYTMIE SUNDAYA** JEŚLI ZNAK Z PRZESZUKIWANEGO TEKSTU NIE WYSTĘPUJE DALEJ WE WZORCU, TO OKNO WZORCA MOŻEMY OD RAZU PRZESUNĄĆ O TYLE POZYCJI, ILE ZNAKÓW ZAWIERA WZORZEC +1. W PRZECIWNYM RAZIE WZORZEC POZYCJONUJEMY TAK, ABY ZGRAĆ POZYCJĘ ZNAKU WYSTĘPUJĄCEGO JEDNOCZEŚNIE W PRZESZUKIWANYM TEKŚCIE I WE WZORCU
ALGORYTM NAIWNY NIE POMIJA W TEN SPOSÓB ZNAKÓW W PRZESZUKIWANYM TEKŚCIE.
- ALGORYTM SUNDAYA TWORZY TABLICĘ POMOCNICZĄ DLA WZORCA, INDEKSOWANĄ LITERAMI I ZAPISUJĄCYMI W NIEJ POZYCJĘ, NA KTÓREJ ZNAJDUJĄ SIĘ NASZE ZNAKI.
- RZĄD CZASOWEJ ZŁOŻONOŚCI OBLICZENIOWEJ: **OPTYMISTYCZNA** ALGORYTM SUNDAYA $O(N / M + 1)$) ALGORYTM NAIWNY $O(N)$
- W **ALGORYTMIE NAIWNYM** WZORZEC JEST ZAWSZE „PRZESUWANY” O JEDEN ZNAK

N - LICZBA ZNAKÓW
TEKSTU

M - LICZBA ZNAKÓW
WZORCA

A B A B D A B B A A C B B B C D A

A B B

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,

A B A B D A B B A A C B B B C D A

A B B

Algorytm sprawdza czy pierwszy znak za oknem wyszukiwania znajduje się w naszym wzorcu,

Widzimy, że znak B występuje w naszym wzorcu,

A B A B D A B B A A C B B B C D A

A B B

S

U

N

D

A

Y

Program mający informację z pomocniczej tablicy, pozycjonuje wzorzec tak, aby zgrać pozycje znaku występującego jednocześnie w przeszukiwanym tekście i we wzorcu (w tym przypadku znak B).

Teraz widzimy, że znaki **A** oraz znak **B** się nie zgadzają, jedyna zgodność występuje na ostatniej pozycji,

A **B A B** D A B B A A C B B B C D A
A B B

Algorytm sprawdza czy kolejny znak za oknem wyszukiwania znajduje się w naszym wzorcu, widzimy, że znaku **D** nie ma w naszym wzorcu,

A B A B D **A B B** A A C B B B C D A
A B B

Algorytm przesuwa nasz wzorzec o **długość naszego wzorca + 1**

Algorytm sprawdza nasze znaki,

Widzimy, że występuje **zgodność** na wszystkich pozycjach,

Algorytm kończy pracę

A B A B D A B B A A C B B B C D A

A B B

Algorytm sprawdza znaki z naszego okna wyszukiwania ze znakami z naszego wzorca,

Ponieważ $A \neq B$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 2giej litery tekstu:

A B A B D A B B A A C B B B C D A

A B B

Ponieważ $B \neq A$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 3ciej litery tekstu:

A B A B D A B B A A C B B B C D A

A B B

Ponieważ $D \neq B$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 4tej litery tekstu:

N

A

I

W

N

Y

A B A B D A B B A A C B B B C D A

A B B

Ponieważ $B \neq A$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 5tej litery tekstu:

A B A B D A B B A A C B B B C D A

A B B

Ponieważ $D \neq A$, więc algorytm rozpoczyna się od początku, jednakże zaczynając od 6tej litery tekstu:

A B A B D A B B A A C B B B C D A

A B B

Ponieważ wszystkie 3 litery są zgodne, więc został odnaleziony poszukiwany ciąg.

LITERATURA

- mgr. Jerzy Wałaszek, Naiwne wyszukiwanie wzorca w tekście, http://eduinf.waw.pl/inf/alg/001_search/0046.php
- mgr. Jerzy Wałaszek, Złożoność obliczeniowa czasowa i pamięciowa algorytmów, http://eduinf.waw.pl/inf/utils/010_2010/0216.php
- prezentacja pt. "ALGORYTM SUNDAYA" opracowana przez Aleksandra Sadowskiego , z pomocą dr inż. Piotra Belinga
- dr inż. Błażej Cichy, Algorytmy przeszukiwania wzorca, https://www.issi.uz.zgora.pl/pl/didactic/cichy/aisd_05.pdf
- P. Wróblewski. Algorytmy, struktury danych i techniki programowania, Helion, 1996.
- notatki z wykładów prowadzonych przez dr Piotra Sielskiego z przedmiotu - Zaawansowane Algorytmy

Opracowano przez Martynę Gołacką, z pomocą dr inż. Piotra Belinga

