

Implementacja

W ramach laboratorium zaimplementowałem wymienione algorytmy w języku python i C++. Implementacje w pythonie uruchamiałem zarówno za pomocą zwykłego interpretera, jak i środowiska pypy.

Wzorzec dla którego implementacja naiwna jest zbyt wolna

Przy pomocy skryptu pythona wygenerowałem tekst składający się z tysięcy razy powtórnego słowa, mającego postać tysięcy znaków A i jednego B. Następnie wygenerowałem wzorzec będący pojedynczym takim słowem. Implementacja naiwna dla tak przygotowanego testu (nazwanego AAB) okazała się wielokrotnie wolniejsza, co widać w sekcji z wynikami testów wydajnościowych.

Wzorzec, dla którego przeliczenie tablicy przejścia automatu jest długie

Z implementacji algorytmu KMP i automatu skończonego wydzieliłem funkcje odpowiedzialne za przeliczenie tablicy przejścia. Przygotowałem też tekst, składający się z powtórnego 10 000 razy zbioru liter z alfabetu łacińskiego. Zmierzyłem czasy konstrukcji tablic, otrzymując następujące wyniki:

Sposób implementacji Algorytm Zmierzony czas konstrukcji

C++	Automat	7 131ms
C++	KMP	118ms
Python	Automat	6 934ms
Python	KMP	671ms

Wyniki testów wydajnościowych

Metodologia

Pierwszy pomiar czasu wykonałem z użyciem wykorzystywanego przez Olimpiadę Informatyczną pakietu [sio2jail](#), którego specyfika została opisana szerzej w pracy [praca](#), za pośrednictwem wrappera [oiejg](#). Pomiar oparty jest o liczniki sprzętowe.

Kolejnego pomiaru dokonałem z wykorzystaniem programu time z systemu linux, zapisując zmierzony czas "real".

Algorytm naiwny

Czas zmierzony przy pomocy sio2jail:

Sposób implementacji Kod Test "art" Test "kruszwil" Test "AAB"

C++	3ms	3 658ms	3 511ms
Python	238ms	269 830ms	1 388ms
PyPy	45ms	8 716ms	1 566ms

Czas zmierzony przy pomocy narzędzia time:

Sposób implementacji Kod Test "art" Test "kruszwil" Test "AAB"

C++	10ms	763ms	436ms
Python	73ms	51 122ms	305ms
PyPy	58ms	1 840ms	290ms

Algorytm kmp

Czas zmierzony przy pomocy sio2jail:

Sposób implementacji Kod Test "art" Test "kruszwil" Test "AAB"

C++	2ms	3 044ms	9ms
Python	250ms	276 915ms	1 428ms
PyPy	54ms	5 347ms	66ms

Czas zmierzony przy pomocy narzędzia time:

Sposób implementacji Kod Test "art" Test "kruszwil" Test "AAB"

C++	5ms	900ms	11ms
Python	87ms	52 761ms	302ms
PyPy	64ms	1 532ms	257ms

Automat skończony

Czas zmierzony przy pomocy sio2jail:

Sposób implementacji Kod Test "art" Test "kruszwil" Test "AAB"

C++	4ms	5 196ms	35ms
Python	159ms	157 988ms	566ms
PyPy	69ms	36 666ms	163ms

Czas zmierzony przy pomocy narzędzia time:

Sposób implementacji Kod Test "art" Test "kruszwil" Test "AAB"

C++	19ms	4 934ms	56ms
Python	58ms	31 509ms	132ms
PyPy	83ms	6 758ms	106ms