

Implementacja LCP w $\mathcal{O}(n)$ przy niższym narzucie pamięciowym

Marcin Witkowski

Wstęp

Oryginalny algorytm autorstwa *Kasai et al.* przedstawiony na wykładzie działa w $\mathcal{O}(n)$ aczkolwiek wykorzystuje średnio $13n$ bajtów pamięci podczas swoich obliczeń. Celem tego projektu była implementacja dwóch wersji algorytmu wykorzystujących odpowiednio $9n$ i $(6 + \delta)n$ pamięci używając trików z *Two Space Saving Tricks for Linear Time LCP Array Computation*. Wy tłumaczenie algorytmów można znaleźć w kodzie źródłowym `src/algorithms.rs`.

Testy

Testy można uruchomić używając komendy `$ cargo test --lib --release`. Zostały one zrealizowane przez testowanie odpowiedzi między algorytmami przy użyciu 10000 losowych słów długości 1000.

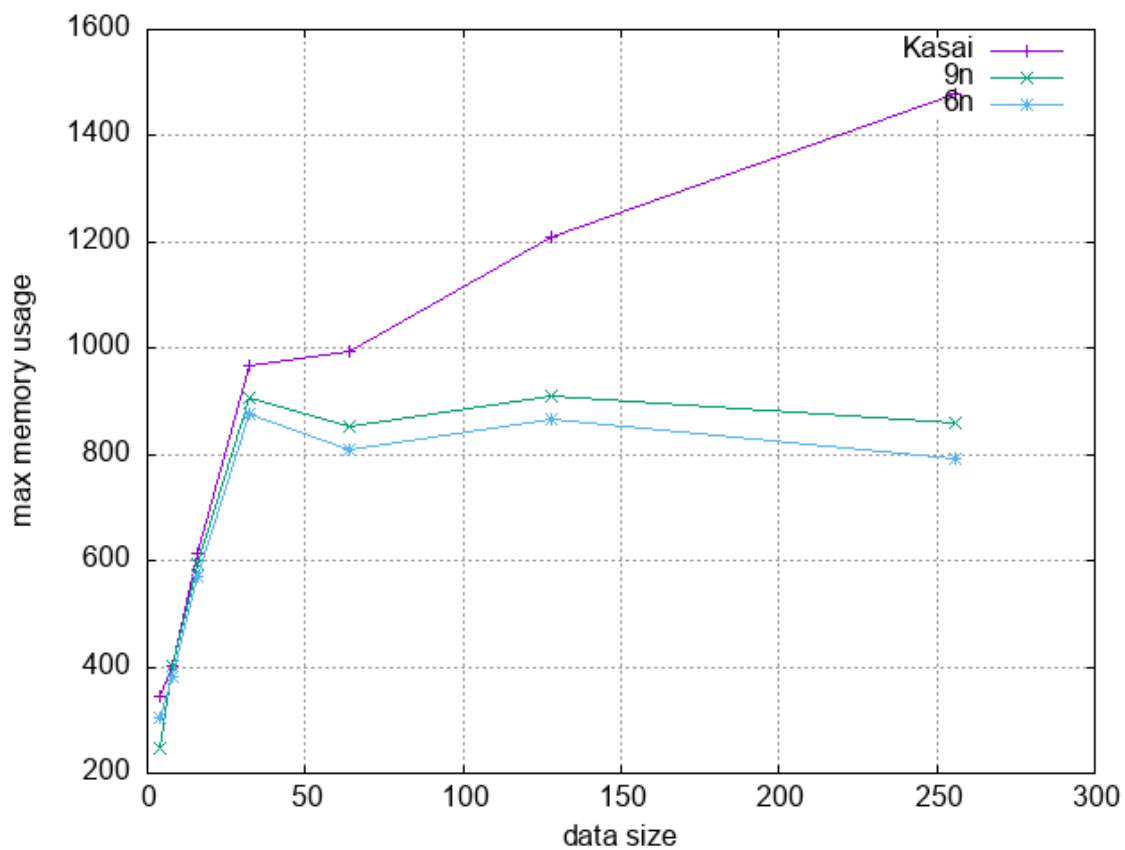
Benchmarki

Pamięciowe

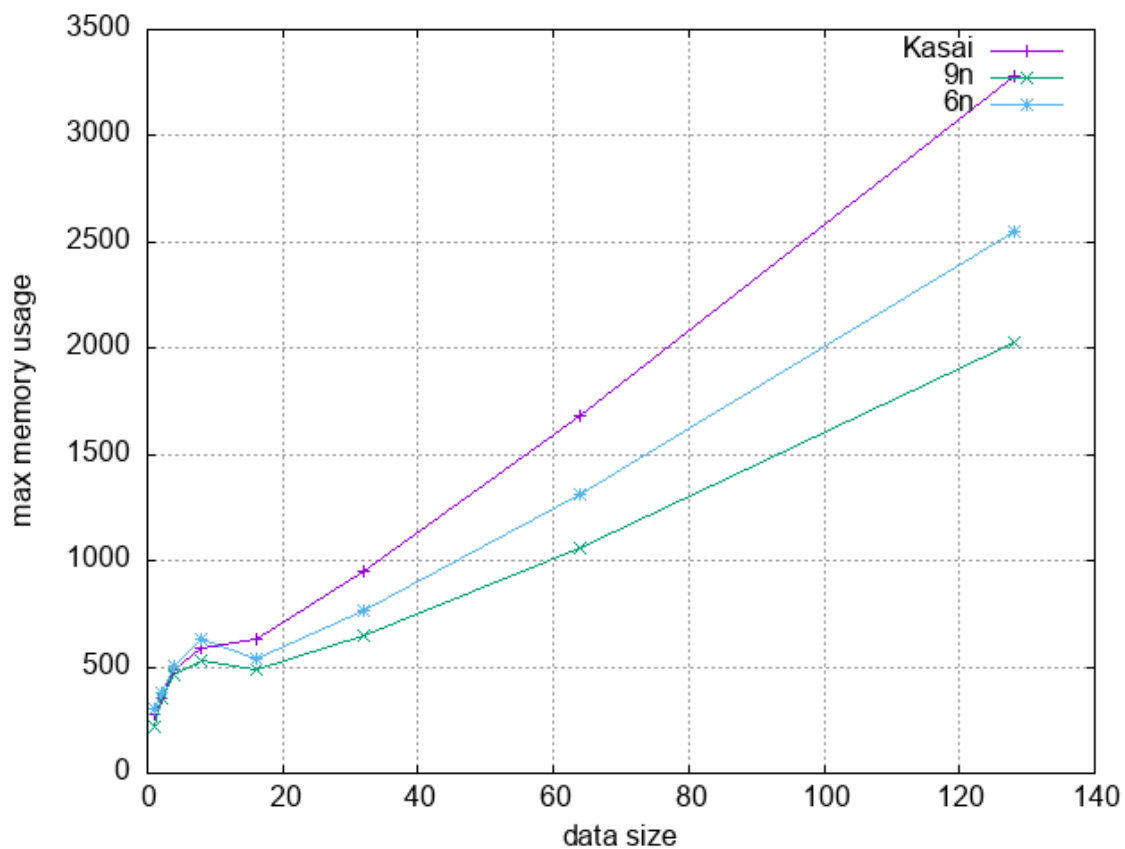
Na tekście naturalnym oraz na słowach fibonacciego, algorytm wykorzystujący $(6n + \delta)$ faktycznie jest najlepszy. W przypadku słów losowych mamy natomiast sporo wartości dla których $\text{BWT}[k - 1] \neq \text{BWT}[k]$, $k = \text{RSA}[i]$, przez co tablica pomocniczna `ssa` rośnie do sporych rozmiarów w związku z czym musimy skopiować prawie całą tablicę SA. W każdym jednak przypadku obie implementacje były sporo lepsze niż niemodyfikowany algorytm Kasai-a. Benchmarki można uruchomić samemu korzystając z komendy `$ cargo bench`.

Złożonościowe

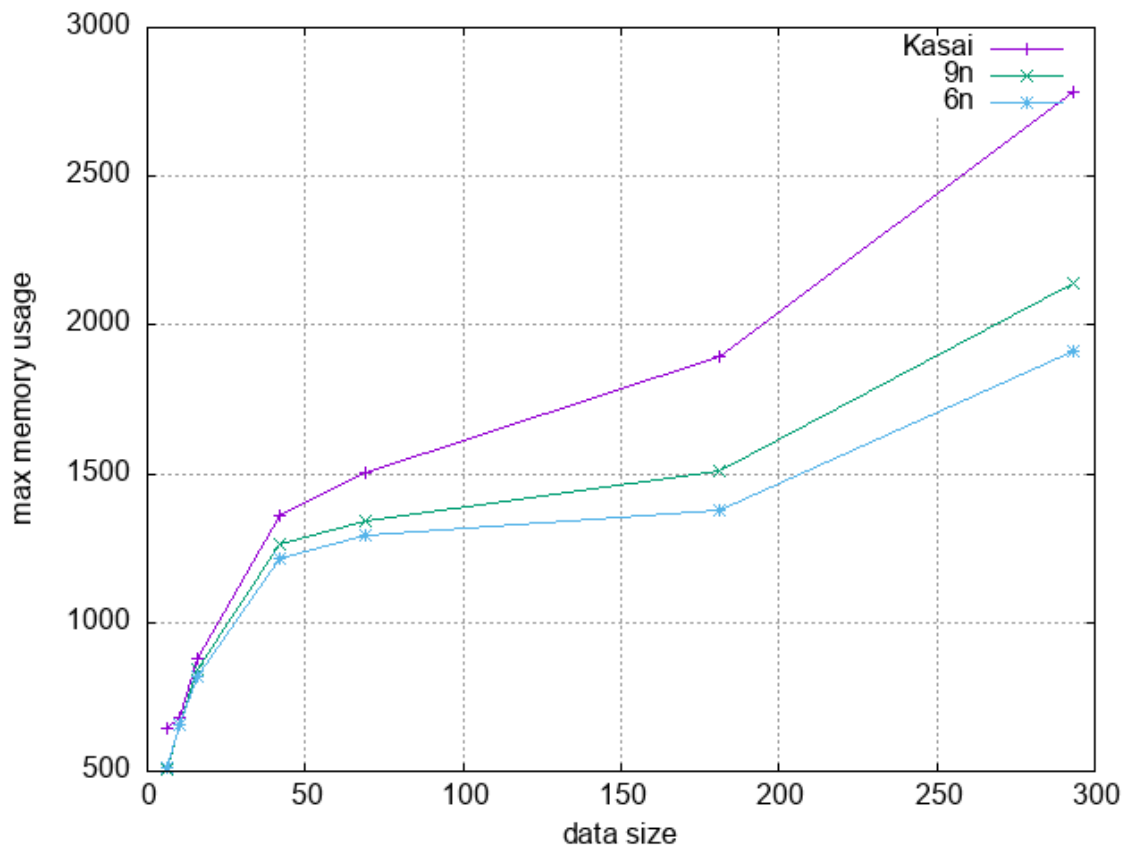
Wszystkie wykresy jak i więcej danych można znaleźć w katalogu `charts`, a dostęp do wszystkich wykresów można uzyskać korzystając z `charts/report/index.html`. Podsumowując, wszystkie algorytmy działają w czasie $\mathcal{O}(n)$, przy czym oryginalny algorytm Kasai-a był najszybszy, następnie $9n$ i algorytm działający w pamięci $(6 + \delta)n$ jest ostatni. Do testowania złożoności zostały użyte słowa losowe oraz słowa fibonacciego odpowiedniej długości.



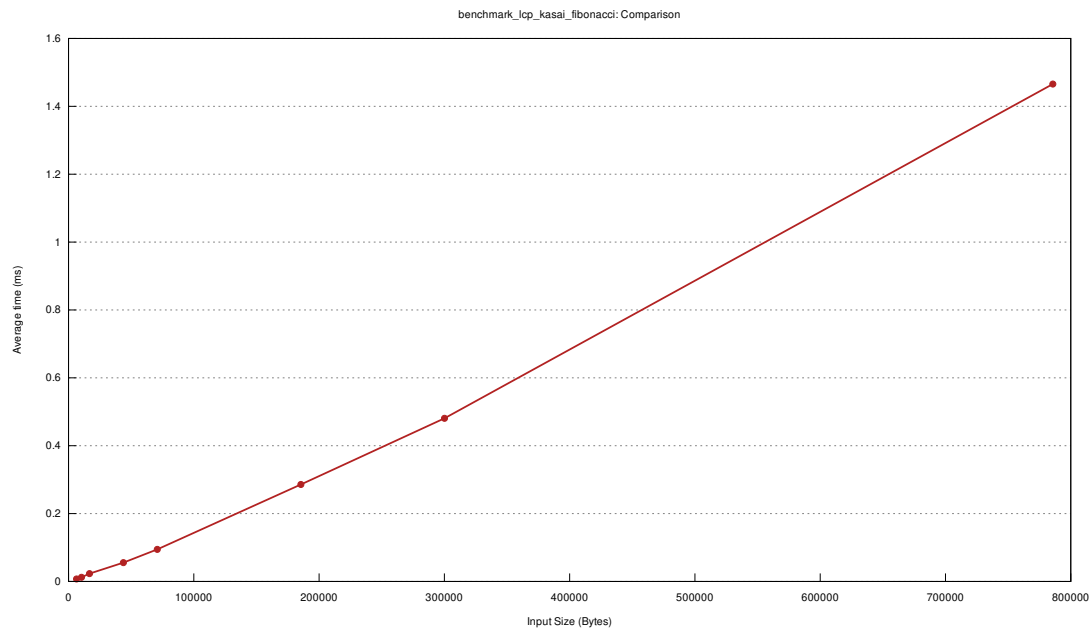
Rysunek 1: Memory benchmark on Lorem Ipsum



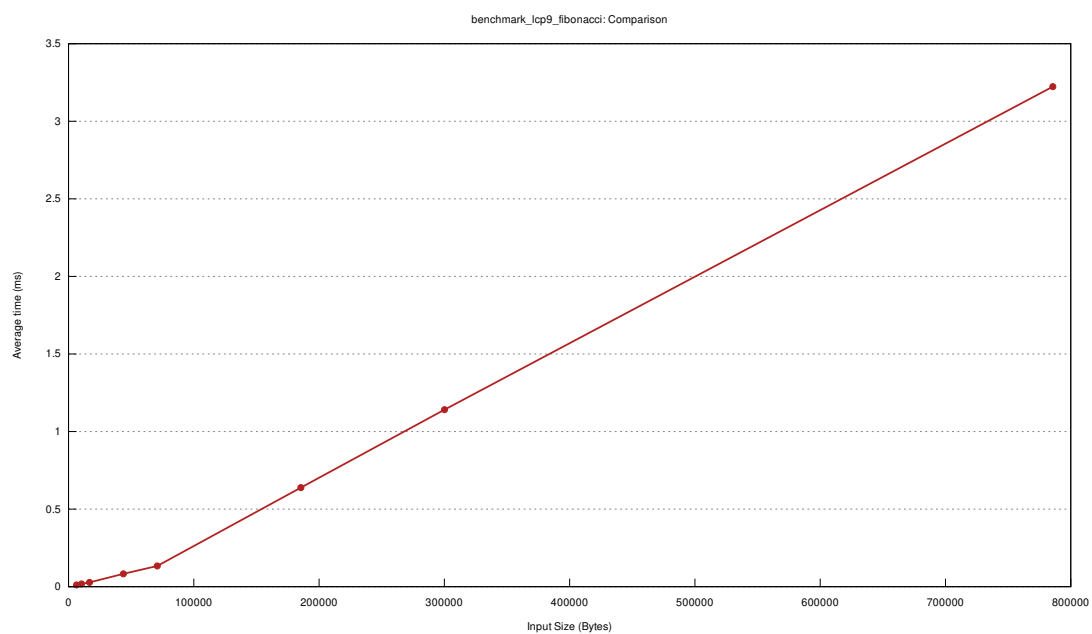
Rysunek 2: Memory benchmark on random words



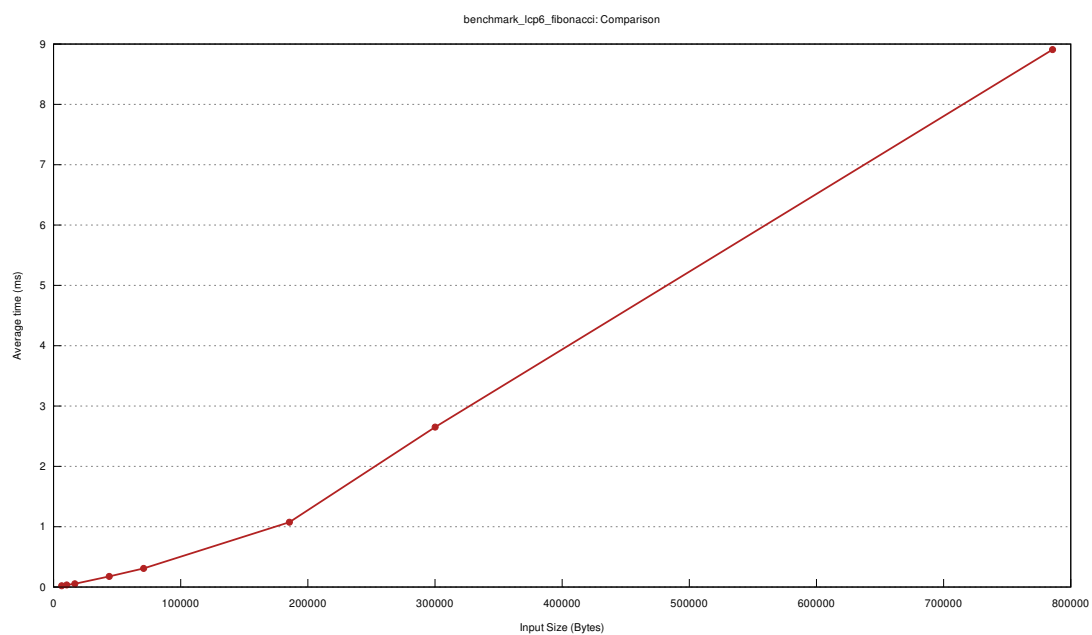
Rysunek 3: Memory benchmark on fibonacci words



Rysunek 4: Kasai LCP on Fibonacci words



Rysunek 5: $9n$ LCP on Fibonacci words



Rysunek 6: $(6 + \delta)n$ LCP on Fibonacci words