K24 - KMP, okresy, szablony

06 June 2025 17:06

w # t

w#t

Prefikso-sufiks słowa – słowo, które znajduje się na początku i końcu rozpatrywanego słowa

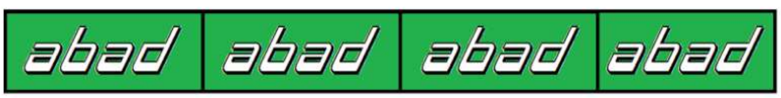
$abc#abobcac

0 0 0 1 2 1 2 3 1 0

Okres słowa - słowo, które przyłożone obok siebie pewną ilość razy utworzy nam rozpatrywane słowo (np. 'abc' jest okresem słowa 'abcabca', słowo 'abcd' jest okresem słowa 'abcdab', a słowo 'ab' nie jest okresem słowa 'bab')

Pierwiastek słowa – idealnie pasujący (inaczej: pełny) okres słowa (np. podsłowo 'aba' jest pierwiastkiem słowa 'abaabaaba', ale podsłowo 'abcd' nie jest pierwiastkiem słowa 'abcdab').

Pierwiastek pierwotny słowa – najkrótszy ze wszystkich pierwiastków słowa



Szablon – słowo, którego wystąpieniami można pokryć cały tekst


```
P[1] = P[0] = 0;
for(int i=2;i<n;i++) {
       int prefiks = P[i-1];
       while (prefiks > 0 && s[prefiks + 1] != s[i])
               prefiks = P[prefiks];
       if (s[prefiks + 1] == s[i])
               prefiks++;
       P[i] = prefiks;
}
```

Szukanie wzorca w tekście

```cpp
void f(const string &s, const string &w) {
    //S jest postaci $wzorzec#tekst
    P[1] = P[0] = 0;
    for(int i = 2;i < s.length();++i) {
        int t = P[i-1];
        while(t > 0 && s[t+1] != s[i])
            t = P[t];
        if(s[t+1] == s[i])
            t++;
        P[i] = t;
        if(P[i] == w.length()) {
            cout << "Mamy wzorzec, start w " << i-w.length()-w.length() << endl;
        }
    }
}
f("$aba#abacabacdabadcaadabacaba", "aba");
```

Lemat: Jeżeli pewne słowo T ma swój prefikso-sufiks S, a on ma własny prefikso-sufiks Z, Z jest też prefikso-sufiksem T.

abacabd ... abacabac abd

1 2 3 4 5 6 2 4 5 6 7

aa a aa b

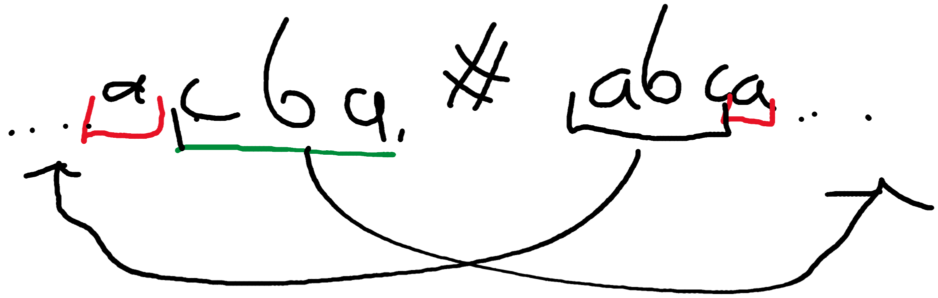
Na pierwszy rzut oka wydaje się, że funkcja ta działa w $O(n^2)$, ponieważ mamy tu pętlę w pętli. Możemy jednak zauważyć, że dla jednej iteracji głównej pętli zmienna „prefiks" rośnie co najwyżej o 1, a pętla while może ją tylko zmniejszać. Ponieważ nie możemy odjąć więcej niż dodaliśmy, odjęć pętli while również nie będzie więcej niż n. Wyznaczanie funkcji prefiksowej działa więc w zamortyzowanym czasie $O(n)$.

abaa ba

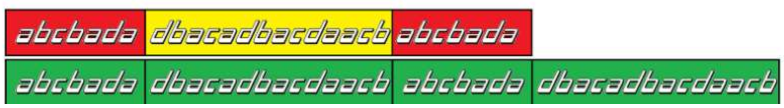
aaba # abaa ba   t=r#s

abcacba

a b c a c b a



Prefikso-sufiksom jednoznacznie odpowiadają okresy słowa. Zauważmy, że gdy słowo długości n ma prefiksosufiks o długości k, to ma też okres długości $n – k$.



Z tego, w szczególności, wynika, że najkrótszy okres odpowiada najdłuższemu prefikso-sufiksowi. Wprowadźmy jeszcze jedno narzędzie, które pomoże nam badać okresy słów.

Lemat o okresowości: Jeżeli słowo ma dwa okresy długości p i q, to $NWD(p, q)$ (największy wspólny dzielnik) także jest okresem tego słowa.

```
for (i=1 i<n i++)
  if (P[i]==0) continue;
  j=i
  while (P[P[j]] != 0)
    j=P[j]
  res += i-P[j]
```

```
for (i=1 i<n i++)
  if (P[i]==0) continue;
  if (P[P[i]]==0)
    MinP[i] = P[i]
  else
```

MinP[i] = P[i]

else

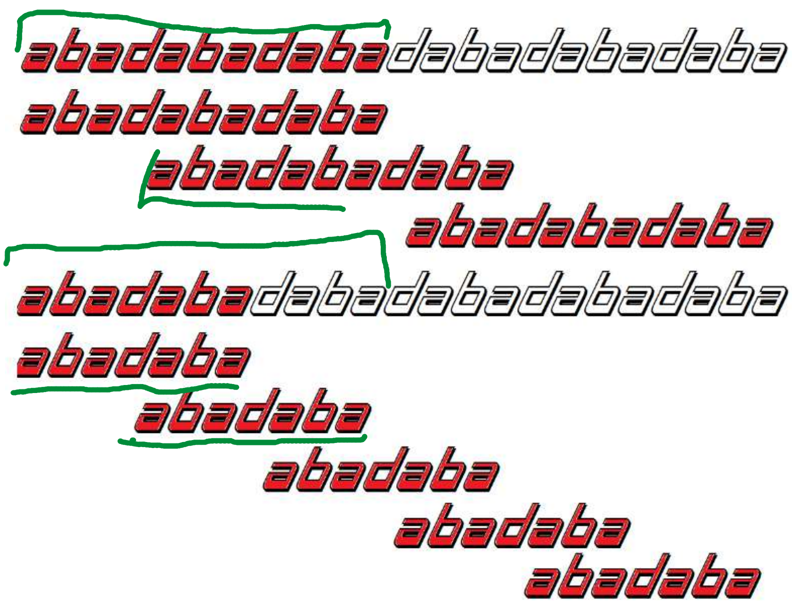
MinP[i] = MinP[P[i]]

res += i - MinP[i]

)

Na pierwszy rzut oka widać, że szablon jest zawsze prefikso-sufiksem danego słowa. Dla każdego prefikso-sufiksu możemy więc sprawdzić, czy jest szablonem w czasie liniowym, znajdując jego wystąpienia w słowie przy pomocy algorytmu KMP. Niestety, kandydatów na szablon mamy pesyimstycznie $O(n)$, a KMP potrafi sprawdzić każdy z nich w czasie $O(n)$, więc całkowita złożoność wynosiłaby $O(n^2)$. Spróbujmy poszukać czegoś lepszego.

Lemat o szablonach: Jeżeli p jest szablonem słowa s, zaś q jest jego prefikso-sufiksem, to jeżeli $p/2 \le q \le p$, to również q jest szablonem s.



$q \to 2q \quad O(n \lg n)$