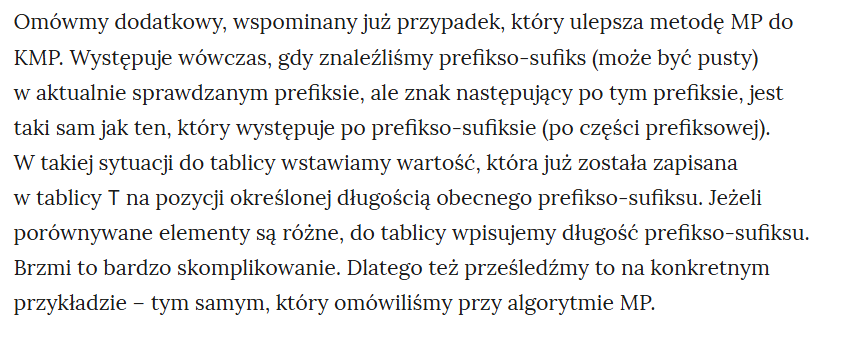
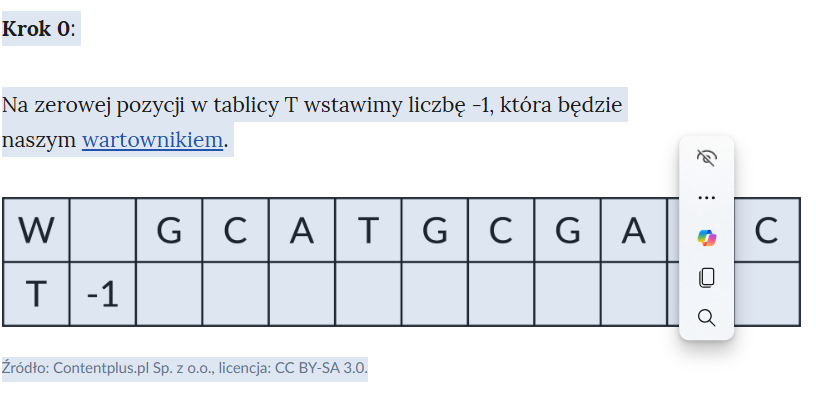
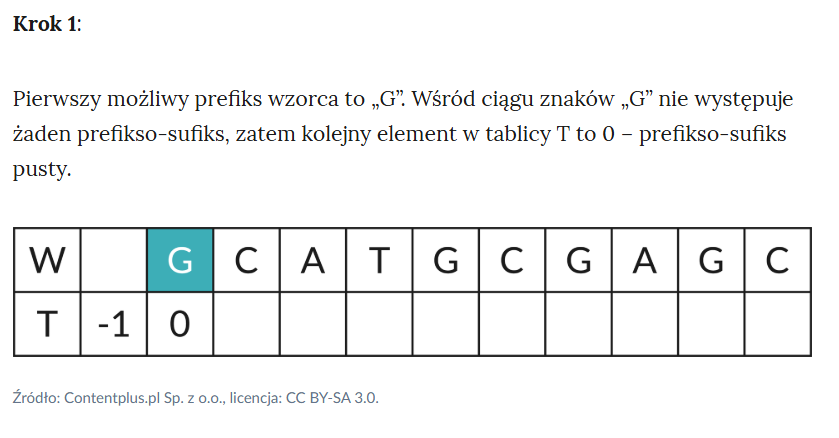
Algorytm KMP (Knutha – Morrisa – Pratta)

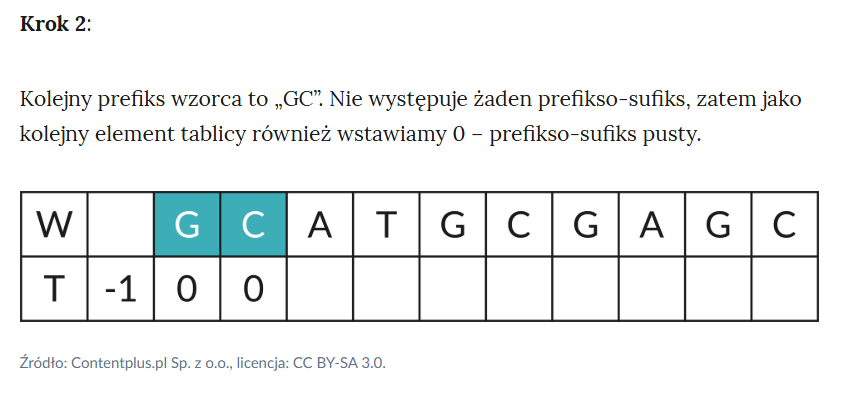
**T –** łańcuch znakowy

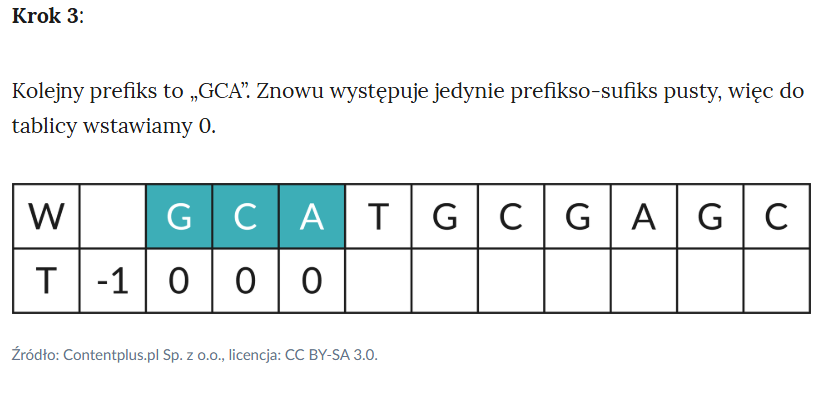
**w** – wzorzec

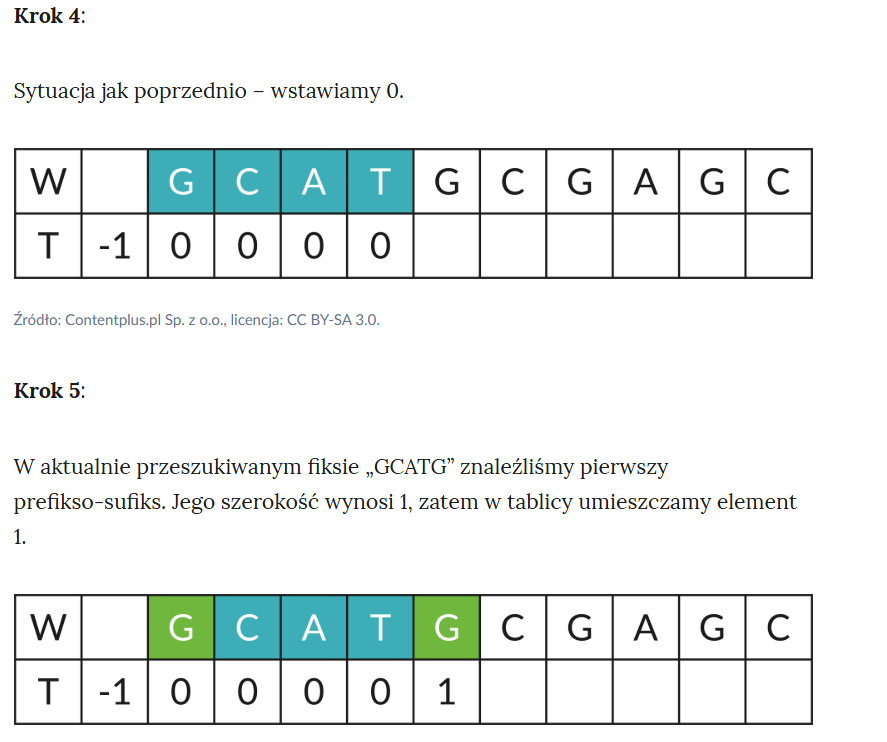
* Gdy znaleźliśmy potencjalne dopasowanie, które okazuje się błędne, algorytm KMP pomija sprawdzanie dopasowania na tych następnych pozycjach, które na pewno okażą się błędne.
* Pomocna jest analiza wstępna wzorca, która polega na utworzeniu tablicy częściowych dopasowani.
* prefix – składająca się z k znaków przednia część łańcucha znaków
* sufix – składająca się z k znaków końcowa część łańcucha
* prefixo-sufix – składająca się k znaków cześć łańcucha, która występuje zarówno z przodu jak i z tyłu
* szerokość prefixo-sufixu - długość prefixu lub sufixu, z którego składa się pefixo-sufix
* Budowa tablicy częściowych dopasowani w algorytmie **MP** polega na wyznaczeniu maksymalnego prefixo-sufixu dla każdego możliwego prefixu we wzorcu:

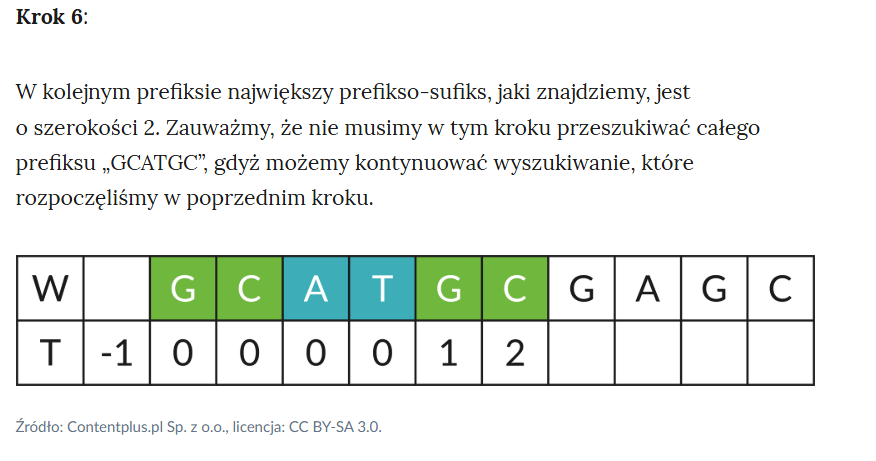


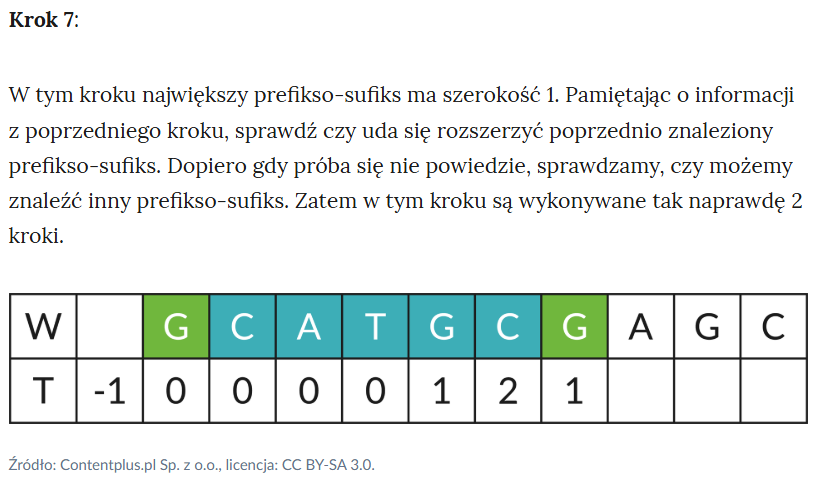


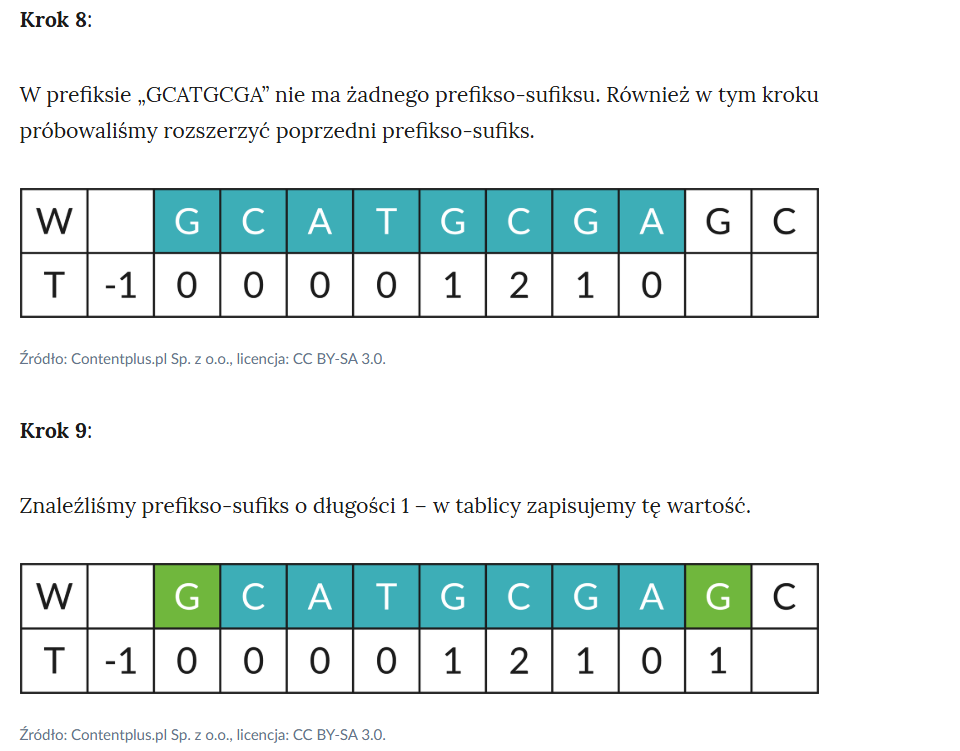


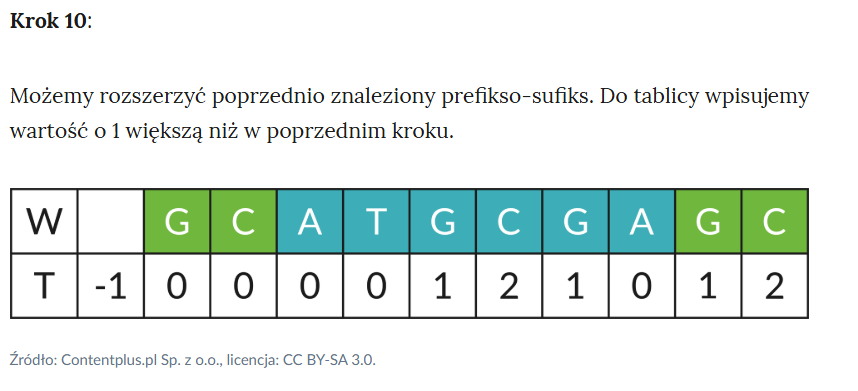


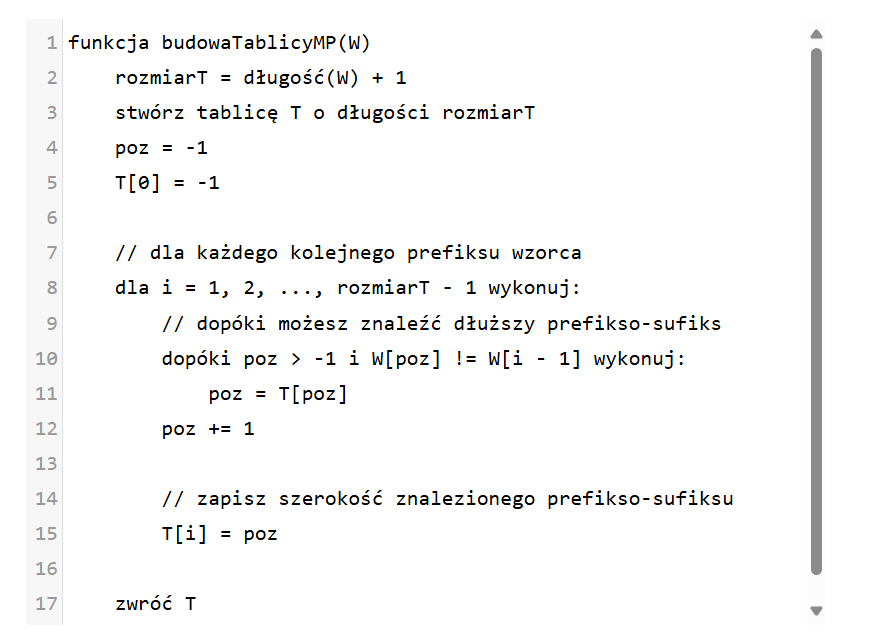


****

****

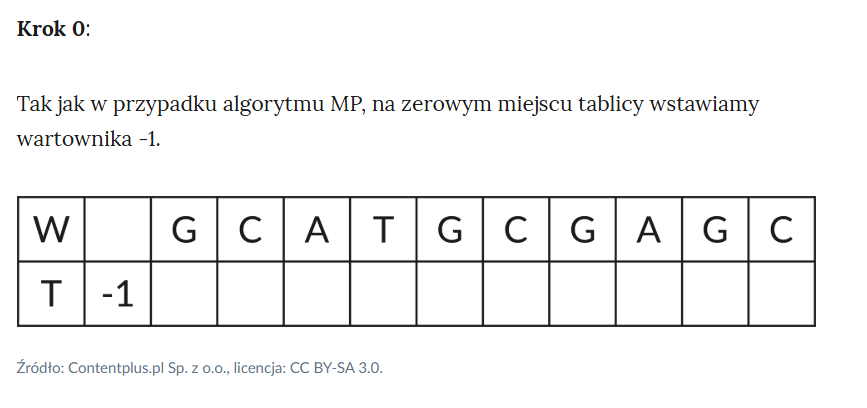
****

****

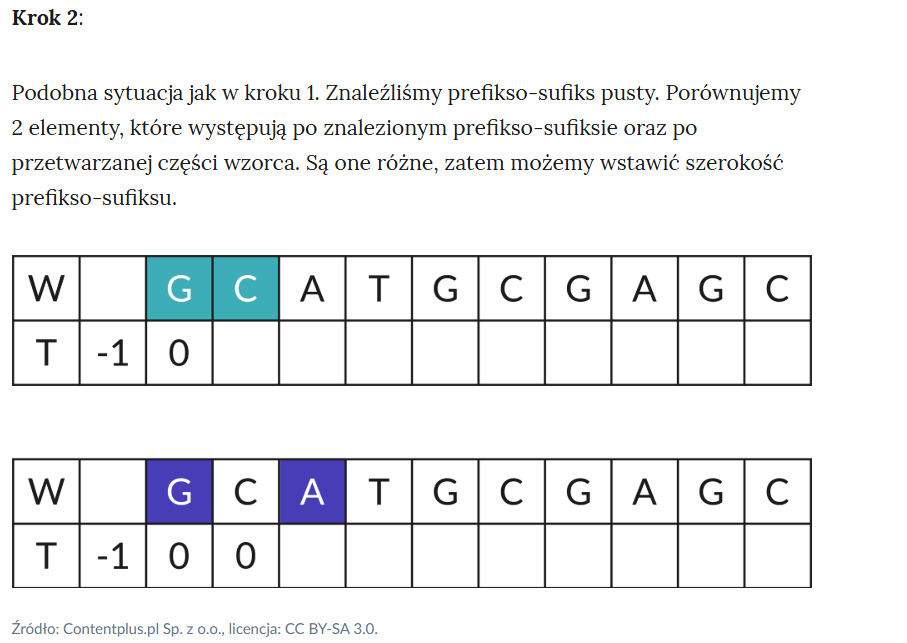
****

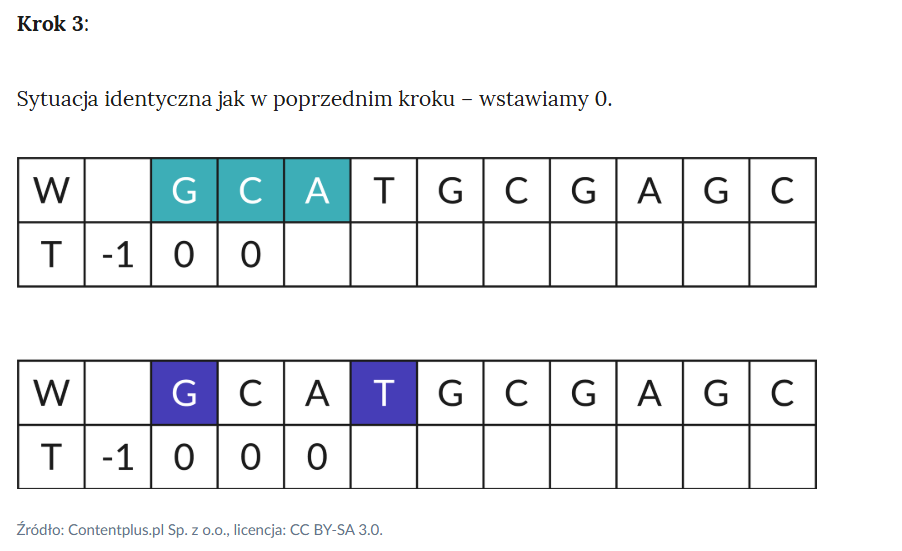
# Usprawnienie

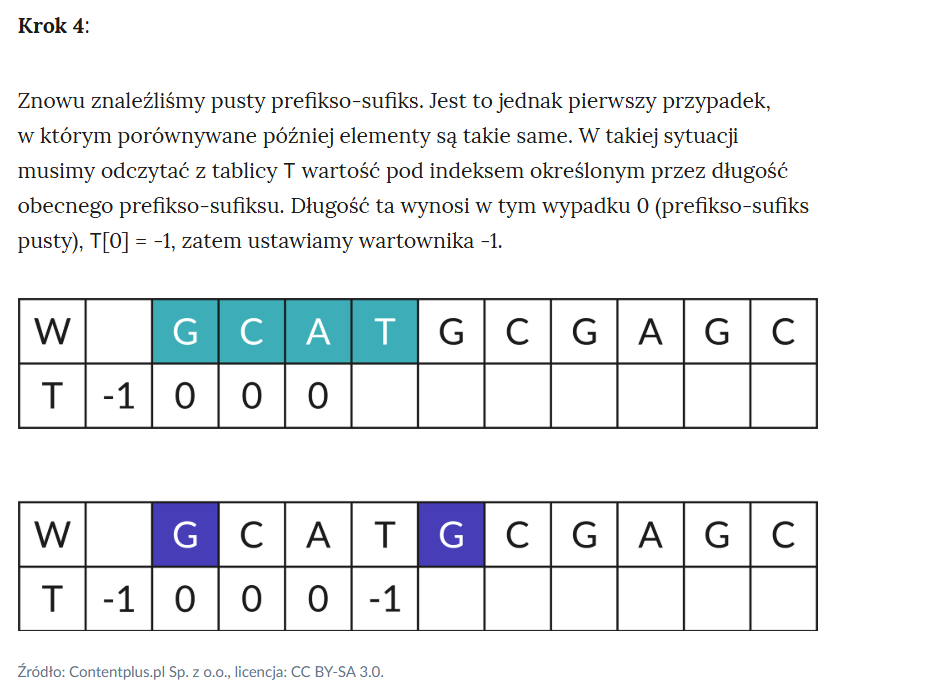
# 

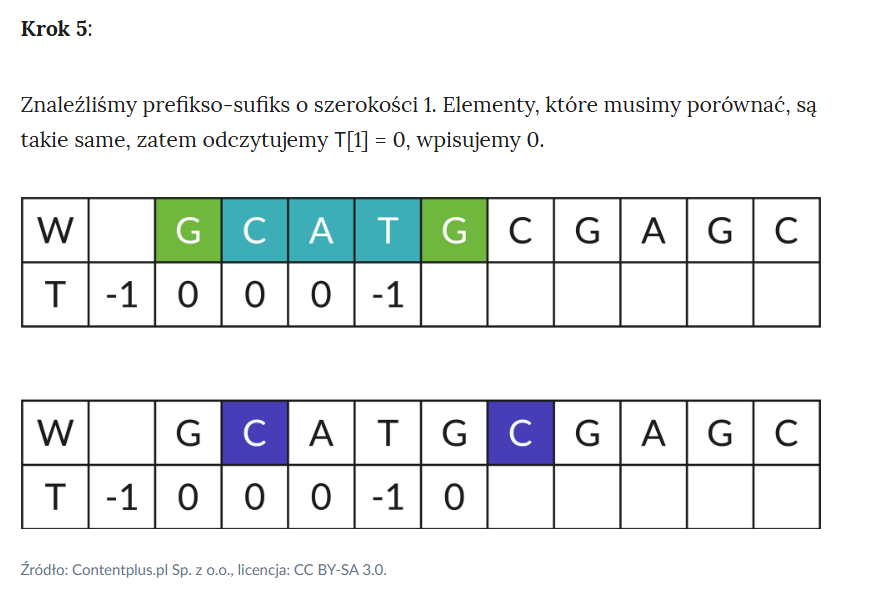


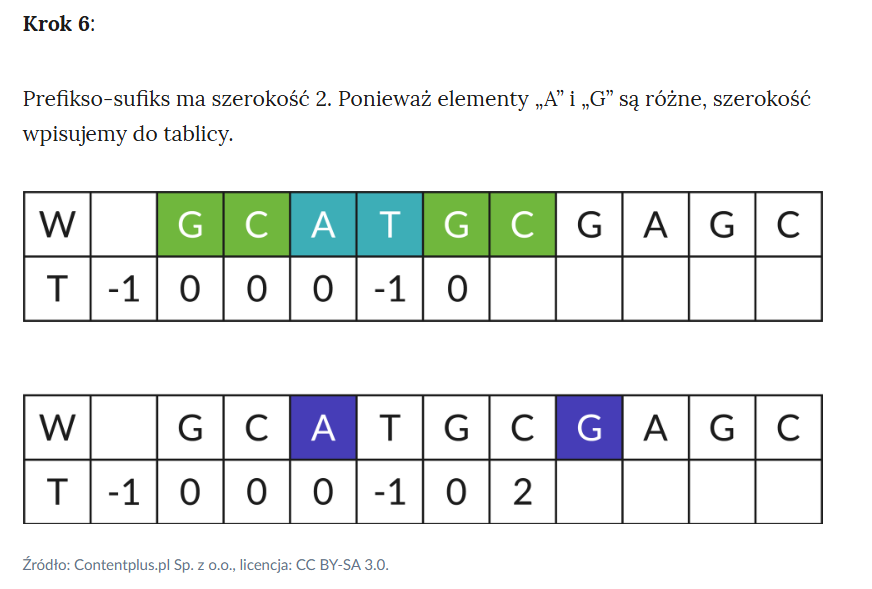
# 

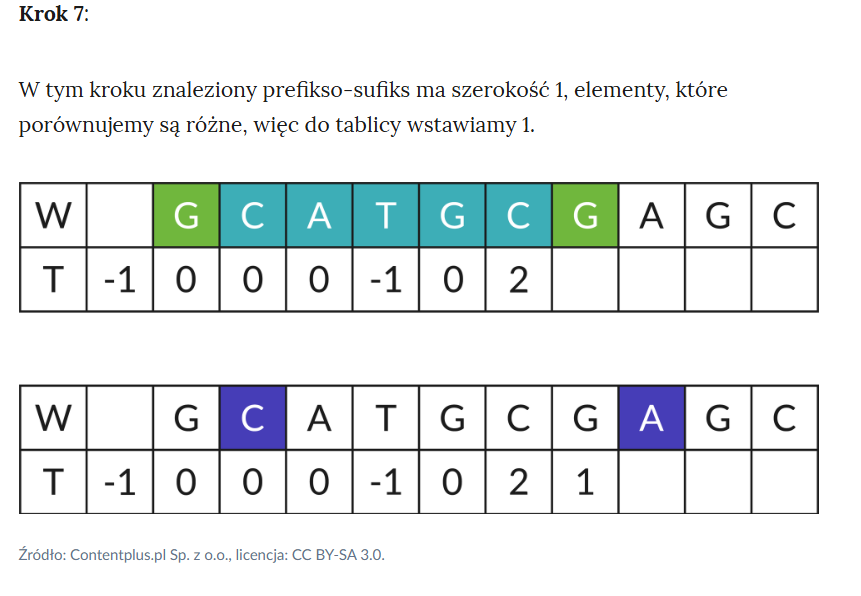


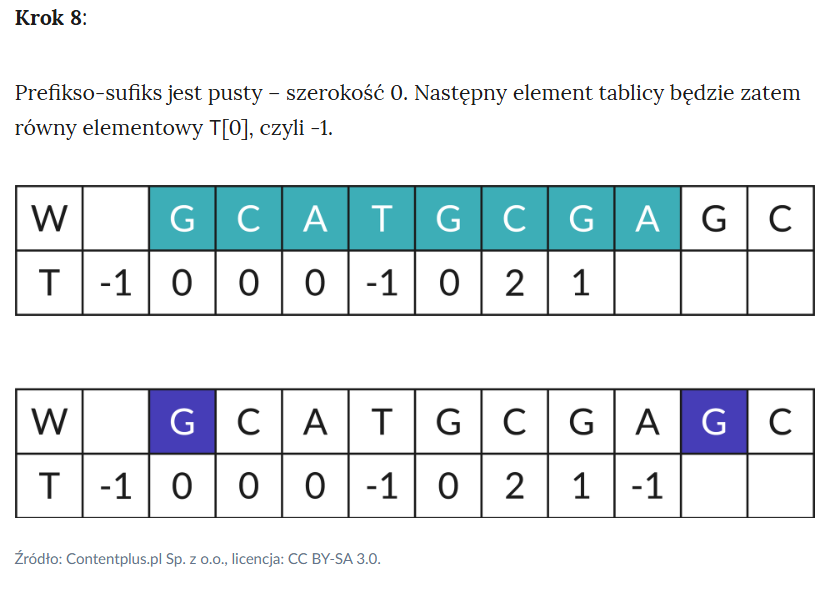


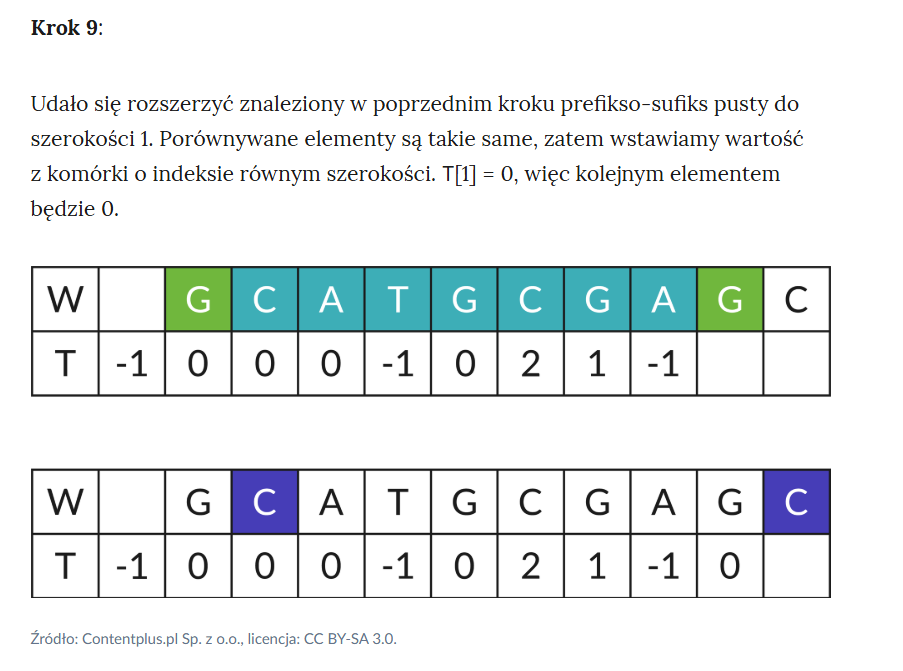


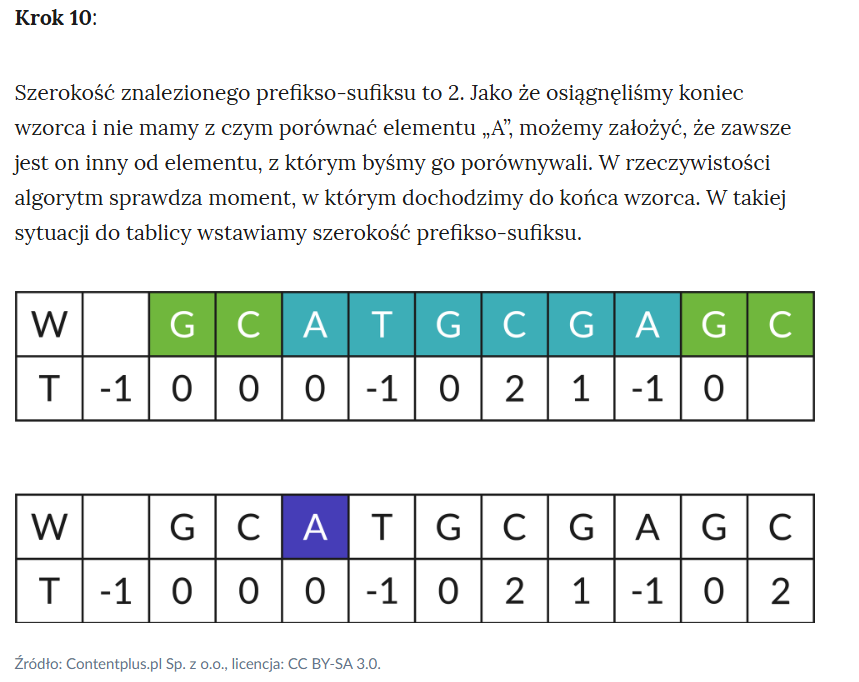












**Pseudokod algorytmu:**

funkcja budowaTablicyKMP(W)

rozmiarT = długość(W) + 1

stwórz tablicę T o długości rozmiarT

poz = -1

T[0] = -1

*// dla każdego kolejnego prefiksu wzorca*

dla i = 1, 2, ..., rozmiarT - 1 wykonuj:

*// dopóki możesz znaleźć dłuższy prefikso-sufiks*

dopóki poz > -1 i W[poz] != W[i - 1] wykonuj:

poz = T[poz]

poz += 1

*// usprawnienie*

*// jeżeli element występujący po znalezionym*

*// prefikso-sufiksie jest taki sam jak po prefiksie*

*jeżeli i == rozmiarT - 1 LUB W[i] != W[poz]*

*// zapisz szerokość znalezionego prefikso-sufiksu*

T[i] = poz

w przeciwnym wypadku

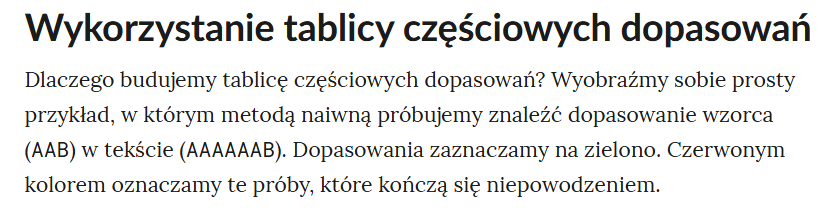
*// kolejny element tablicy będzie tym samym,*

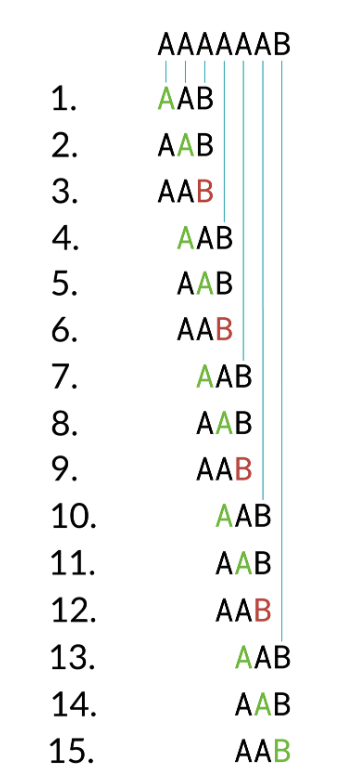
*// co element na pozycji równej szerokości*

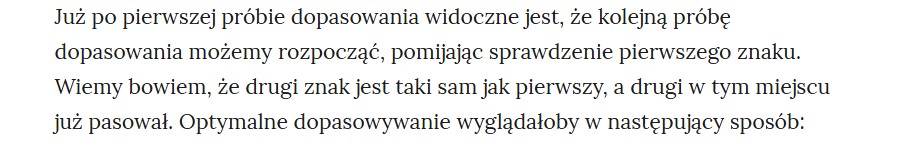
*// prefikso-sufiksu*

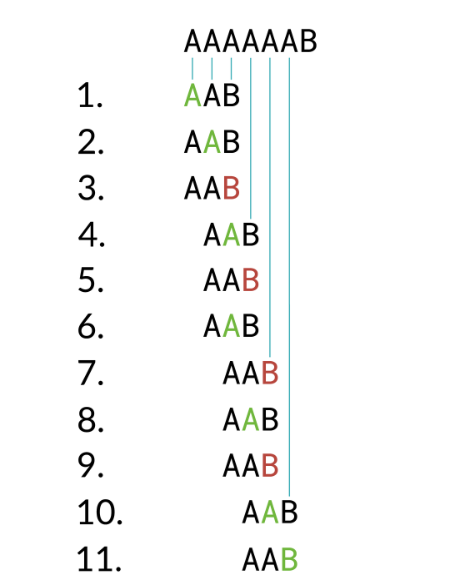
T[i] = T[poz]

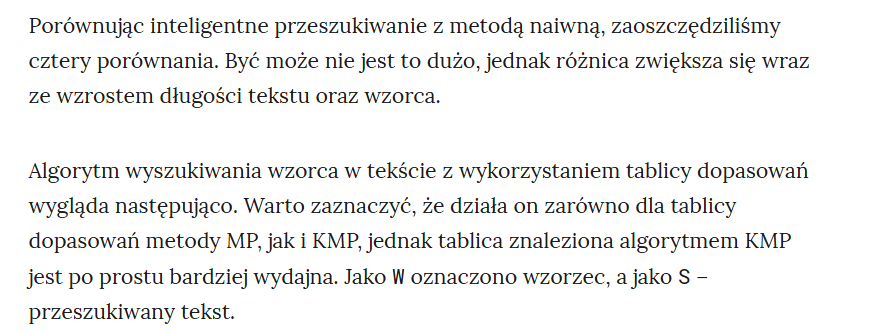
zwróć T











Pseudokod:

funkcja szukajAlgorytmemKMP(W, S)

T = budowaTablicyKMP(W)

b = 0

dla i = 0, 1, ..., długość(S) - 1 wykonuj:

dopóki b > -1 i W[b] != S[i] wykonuj:

b = T[b]

b += 1

jeżeli b == długość(W):

zwróć i - długość(W) + 1

zwróć -1