# V A J E 3. – Kodiranje

1. Ponazorite delovanje KMP algoritma tako, da poiščete vzorec $p$ =ACACA v besedilu $t$ =ACACTACGACACACA.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| q | 0 | 1 | 2 | 3 | 4 | 2 | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 3 | 4 |
| i | 1 | 2 | 3 | 4 | 5 | 5 | 5 | 6 | 7 | 8 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 14 | 15 |
| p[q+1] | A | C | A | C | A | A | A | A | C | A | A | A | C | A | C | A | E | C | A |
| t[i] | A | C | A | C | T | T | T | A | C | G | G | A | C | A | C | A | C | C | A |
| match | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

i=n STOP

| p | A | C | A | C | A |
|---|---|---|---|---|---|
| π | 0 | 0 | 1 | 2 | 3 |
| | 1 | 2 | 3 | 4 | 5 |

q>0 in znaka zaporedij nista enaka (while loop): vzami za q = π[q]
π[4] je 2. Spet preveri pogoj.
q=0 zato se while preneha. Znaka nista enaka, zato gre v naslednjo iteracijo for zanke ⇒ poveča i, a ne tudi q.

```
def KMP(p, t):
    n = |t|, m = |p|, r = []
    π = ConstructPi(p)
    q = 0
    for i = 1 .. n do
        while q > 0 and p[q+1] ≠ t[i] do
            q = π[q]
        if p[q+1] = t[i] then
            q++
        if q = m then
            r += r[i-q+1]
    return r
```

q = m zato v r doda indeks znaka, ki je začel ujemanje (t.j. v tem primeru $i-q+1 = 13-5+1 = 9$

žakaj ne 14? Takoj ko se poveča q se to preveri, preden se poveča i.

2. Zgradite priponsko drevo za tekst $t$ =ATTTGCCG.

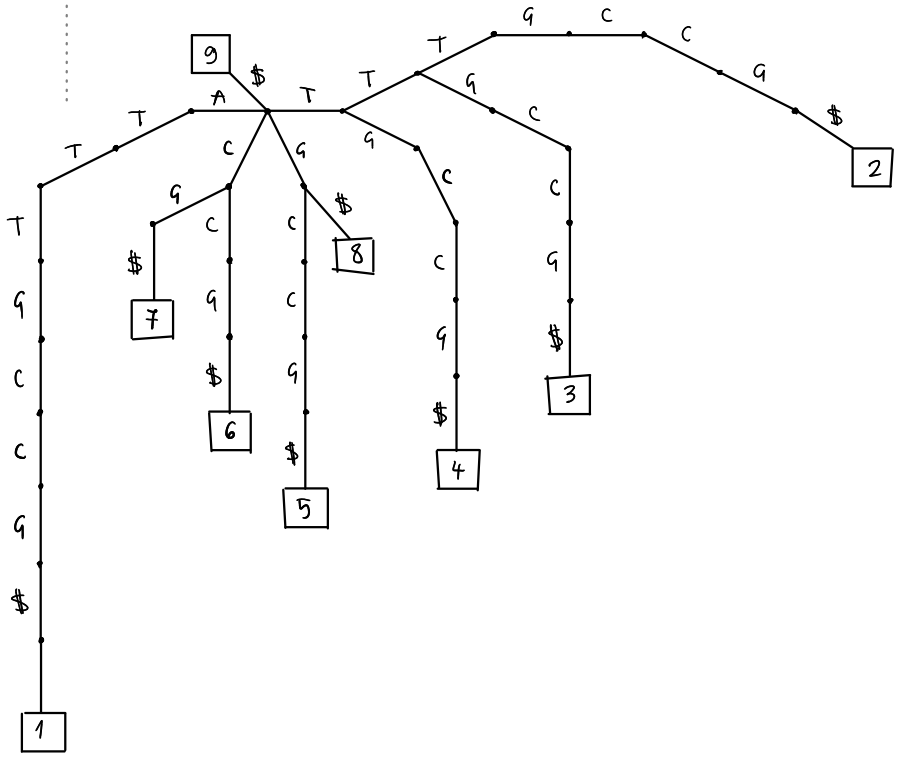
   (a) Ali tekst vsebuje vzorec $p$ =TT? Kje?

   (b) Ali tekst vsebuje vzorec $p$ =CGT? Kje?

1. Izpišeš vse pripone besedila: (|t|+1)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | A | T | T | T | G | C | C | G | $ |
| 2 | T | T | T | G | C | C | G | $ | |
| 3 | T | T | G | C | C | G | $ | | |
| 4 | T | G | C | C | G | $ | | | |
| 5 | G | C | C | G | $ | | | | |
| 6 | C | C | G | $ | | | | | |
| 7 | C | G | $ | | | | | | |
| 8 | G | $ | | | | | | | |
| 9 | $ | | | | | | | | |

2. Izgradiš drevo v vrstnem redu (abecednem):




a) Spustimo se po drevesu do listov; če je razvejitev, vzamemo za rezultat vse možne liste.

Za TT je to indeks 2 in 3.

b) Za CGT je to indeks ... ni ga.

3. Zgradite priponsko polje za tekst iz naloge 2 in odgovorite na enaki vprašanji kot v nalogi 2.

Indekse listov drevesa zapišemo v polje po vrstnem redu: Za iskanje vzorca uporabimo bisekcijo.

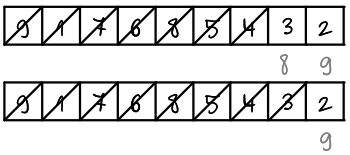
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | 1 | 7 | 6 | 8 | 5 | 4 | 3 | 2 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

a) "TT" 1. Srednji indeks je 5, list/pripona 8 = "G$". Primerjamo vzorec: G$ < TT ⇒ glej desno.

2. Srednji indeks je 4 (vzamemo levega): TG... < TT ⇒ desno.

G je "manjši" od T

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ~~9~~ | ~~1~~ | ~~7~~ | ~~6~~ | ~~8~~ | 5 | 4 | 3 | 2 |
| | | | | 5 | 6 | 7 | 8 | 9 |

3. 

Srednji indeks je 3: TTG... = TT ⇒ ujemanje, ampak glej naprej; desno.

Indeks 2: TTT... = TT ⇒ ujemanje! Zmanjkalo polja.

b) Analogno.

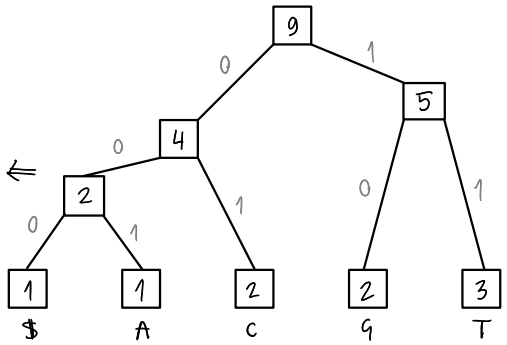
---

4. Dano imamo besedilo $t$ =ATTTGCCG.

(a) Zakodirajte besedilo z uporabo Huffmanovega algoritma.
(b) Zakodirajte besedilo z uporabo Burrows-Wheeler kompresije.
(c) Iz dobljene kode pridobite originalen text $t$.

a) $t_A = 1$, $t_C = 2$, $t_G = 2$, $t_T = 3$, $t_\$ = 1$

Iz frekvenc izgradimo drevo.




Kode:

| | |
|---|---|
| A | 001 |
| C | 01 |
| G | 10 |
| T | 11 |
| $ | 000 |

Št. bitov po kodiranju: $t_A \times |HE(A)| + \dots$
$= t_A \times 3 + t_C \times 2 + t_G \times 2 + t_T \times 2 + t_\$ \times 3 = 3 + 4 + 4 + 6 + 3$
$= 20$-bitov

---

b) ① BWT ② MTF ③ HE

Pripone razvrstimo pa abecednem vrstnem redu. Zadnji stolpec je rezultat naše transformacije:

| | | |
|---|---|---|
| 9 | $ ... | G |
| 1 | A ... | $ |
| 6 | C C ... | G |
| 7 | C G ... | C |
| 8 | G $ ... | C |
| 5 | G C ... | T |
| 4 | T G ... | T |
| 3 | T T G ... | T |
| 2 | T T T ... | T |

① 

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | A | T | T | T | G | C | C | G | $ |
| 2 | T | T | T | G | C | C | G | $ | T |
| 3 | T | T | G | C | C | G | $ | A | T |
| 4 | T | G | C | C | G | $ | A | T | T |
| 5 | G | C | C | G | $ | A | T | T | T |
| 6 | C | C | G | $ | A | T | T | T | G |
| 7 | C | G | $ | A | T | T | T | G | C |
| 8 | G | $ | A | T | T | T | G | C | C |
| 9 | $ | A | T | T | T | G | C | C | G |

⇒ BWT(t) = G $ G C C T T T

Zdaj je v lepši obliki za stiskanje.

---

Po vrsti BWT(t) vzemamo znake in jih dajemo na vrh, ostale premaknemo dol, da zapolnimo luknjo.

② 

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | $ | G | $ | G | C | C | T | T | T |
| 2 | A | $ | G | $ | G | : | C | : | : |
| 3 | C | A | A | A | $ | | G | | |
| 4 | G | C | C | C | A | | $ | | |
| 5 | T | T | T | T | T | | A | | |
| | | 4 | 2 | 2 | 4 | 1 | 5 | 1 | 1 |

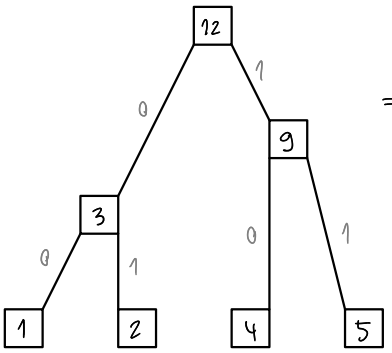
Vrstica iz katere smo vzeli znak.

⇒ MTF(BWT(t)) = 422411511

Znebili smo se enega znaka ⇒ krajše kodiranje!

---

③ $t_1 = 3$, $t_2 = 2$, $t_4 = 2$, $t_5 = 1$




⇒

| Kode: | |
|---|---|
| 1 | 00 |
| 2 | 01 |
| 4 | 10 |
| 5 | 11 |

Št. bitov po kodiranju: $(3+2+2+1) \times 2 =$
$= 8 \times 2 = 16$-bitov

4 manj kot v (a)!

---

c) encoded(t) = 1 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0
4 2 2 4 1 5 1 1

Uporabimo MTF tabelo

Ali pa pač preberemo prvo vrstico.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | $ | G | $ | G | C | C | T | T | T |
| 2 | A | $ | G | $ | G | G | C | C | C |
| 3 | C | A | A | A | $ | $ | G | G | G |
| 4 | G | C | C | C | A | A | $ | $ | $ |
| 5 | T | T | T | T | T | T | A | A | A |
| | | 4 | 2 | 2 | 4 | 1 | 5 | 1 | 1 |

Pove kaj izberemo v prejšnjem stolpcu

⇒ dobimo G $ G C C T T T.

⇒ Iz tabele po vrsti glede na številke vzemamo prve znake besede:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $ | T | T | T | T | G | C | C | G |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| A | T | T | T | G | C | C | G | $ |

Originalno besedilo.

| | | |
|---|---|---|
| 9 | $ ... | G |
| 1 | A ... | $ |
| 6 | C C ... | G |
| 7 | C G ... | C |
| 8 | G $ ... | C |
| 5 | G C ... | T |
| 4 | T G ... | T |
| 3 | T T G ... | T |
| 2 | T T T ... | T |