

4.3)

a) Slika v prilogi

b)  $p = \text{"AN"}$

Algoritem mora vrniti začetne indekse, kje se začne podniz. To pa naredi tako, da gre v drevo.

V drevesu pogleda prvo črko niza "p" in gre v pravilno poddrevo. V tem poddrevesu bo vrnil indeks začetka podniza + vse indekse, ki so mu vrnil njegovi poddrevesi.

Ko najde ujemajoč niz v poddrevesu, mora iti do vseh končnih listov. Sproti šteje globino. Z globino vemo, koliko je dolg string. To pomeni da je začetni indeks podniza =  $\text{len(string)} - \text{globina}$  (od kjer se je razcepil).

Postopek iskanja označen na sliki.

c) Najdaljši podniz je "ANA".

Če imamo podano besedilo dolžine m, lahko poiščemo najdaljši podniz v  $O(m^2)$  -> dve zanki. V prvi povečujemo štart, drugi pa konec.

Lahko naredimo tudi v  $O(n)$  -> primer v C med prilogami

d)

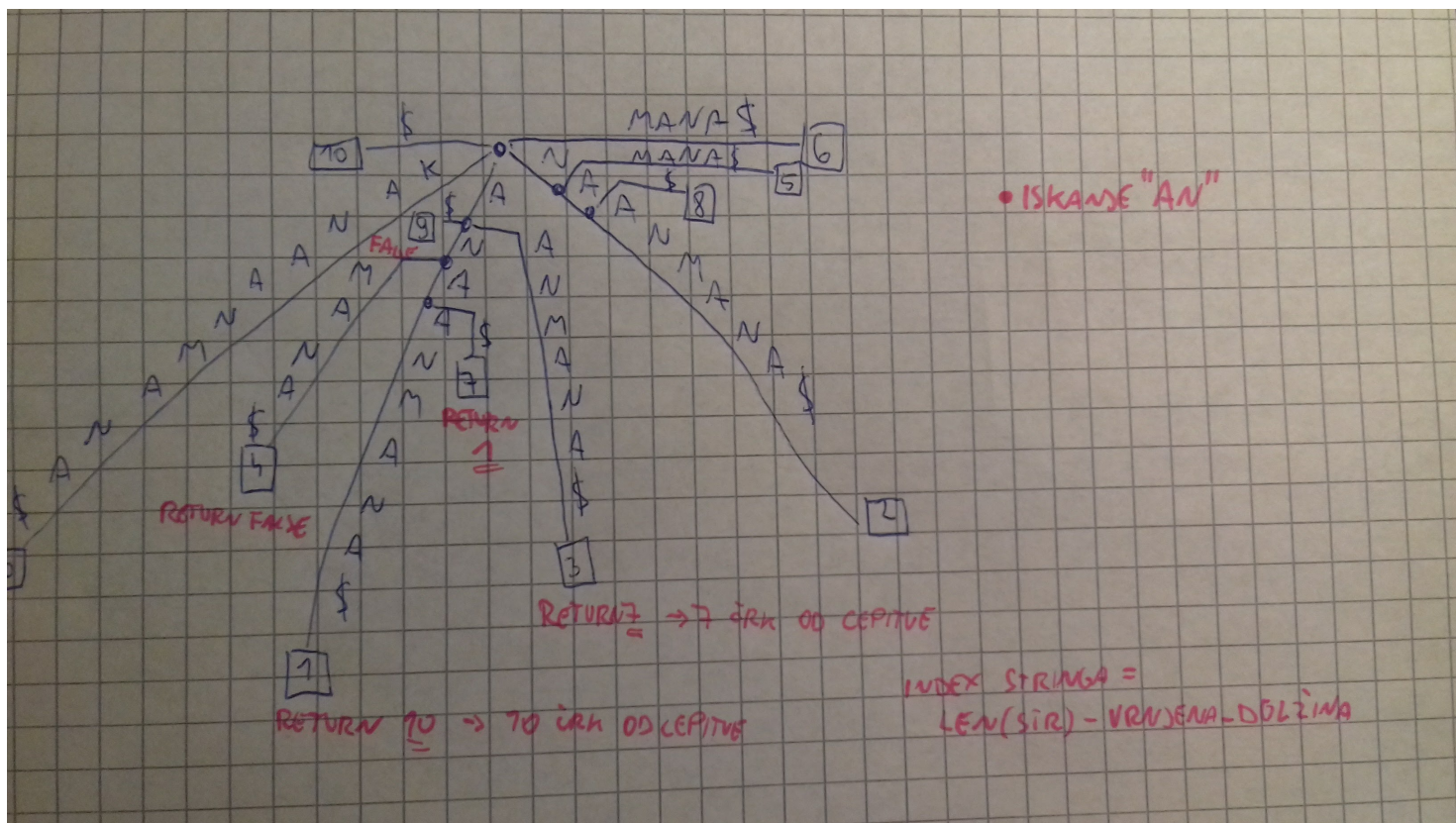
Prav tako tudi tukaj štejemo globino. Štejemo globino, kje nazadnje se je veja razvejala.

Moramo v vsako poddrevo in sproti štejemo globino. Ko se razveja, si globino zapomnimo in gremo v tisto poddrevo.

Ko se vračamo, vračamo max globino.

e) Poddrevesa v ST2 so lahko globlje, kot v ST1, če je c sorodna črka v poddrevesu ST1 (če se pojavlja v ST1)

f) Vsa poddrevesa v ST2 ki so tuja črki c (imajo isto začetno črko) ostanejo na istih mestih, medtem ko vsa ostala poddrevesa spadajo pod črko c in so za 1 krajša



```

44 void improved(long dolzina){
45     // vsota elementov od zadnjega "preloma" (točke, ko je tekoča vsota padla
46     // pod 0) do trenutnega elementa
47     long tekocaVsota = 0;
48
49     // število elementov, ki tvorijo trenutno tekočo vsoto
50     // (tj. število elementov od zadnjega "preloma" do trenutnega elementa)
51     long tekocaDolzina = 0;
52
53     int najZacetek = 0;
54     long najVsota = LLONG_MIN;
55     int najDolzina = 0;
56
57     for (int i = 0; i < dolzina; i++) {
58         int element;
59         scanf("%d", &element);
60
61         if (tekocaVsota < 0) { // prelom!
62             tekocaVsota = 0;
63             tekocaDolzina = 0;
64         }
65
66         tekocaVsota += element;
67         tekocaDolzina++;
68
69         if (tekocaVsota > najVsota) {
70             najVsota = tekocaVsota;
71             najDolzina = tekocaDolzina;
72             najZacetek = i - najDolzina + 1;
73         }
74     }
75
76     printf("%d %d %lld\n", najZacetek, najDolzina, najVsota);
77 }

```