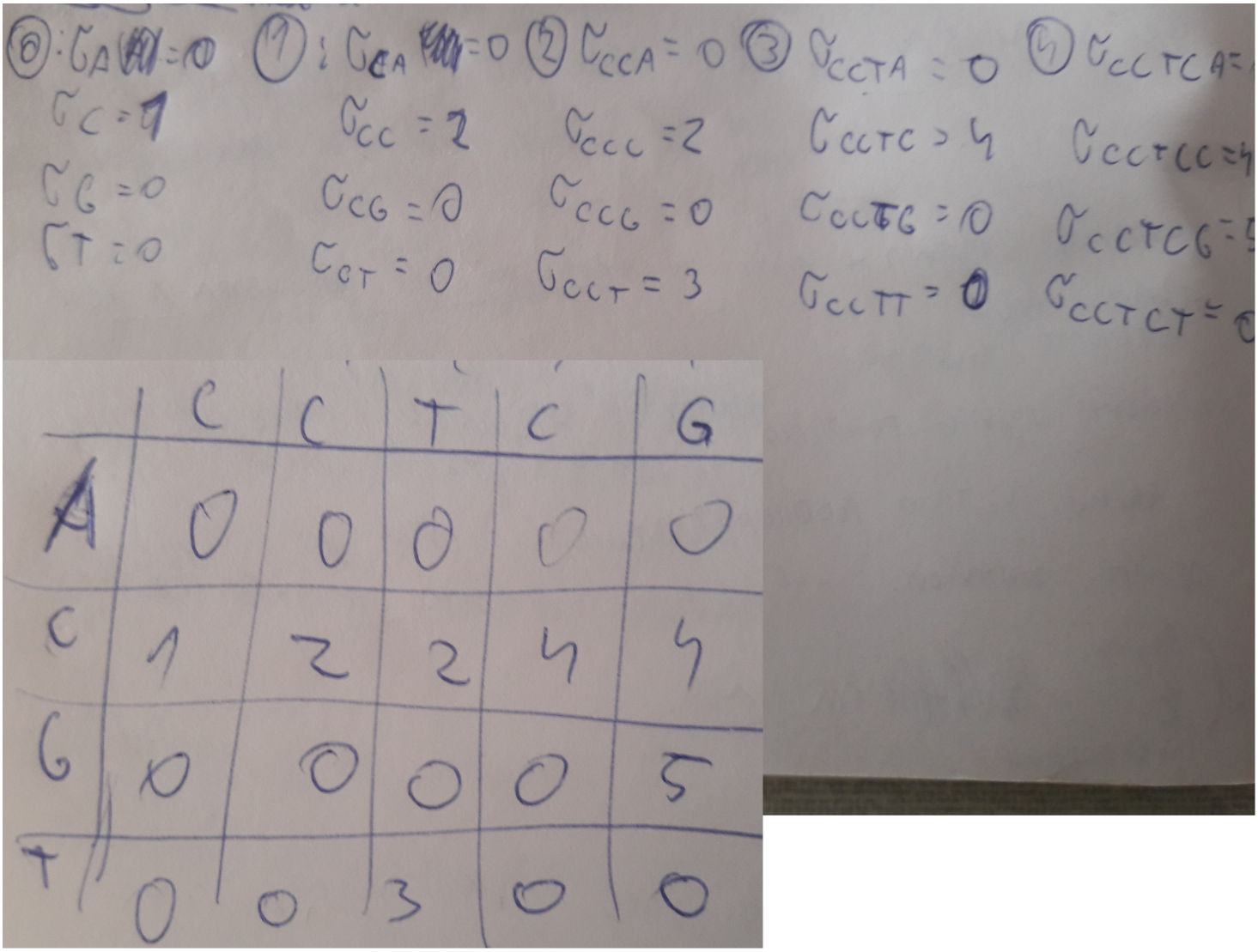
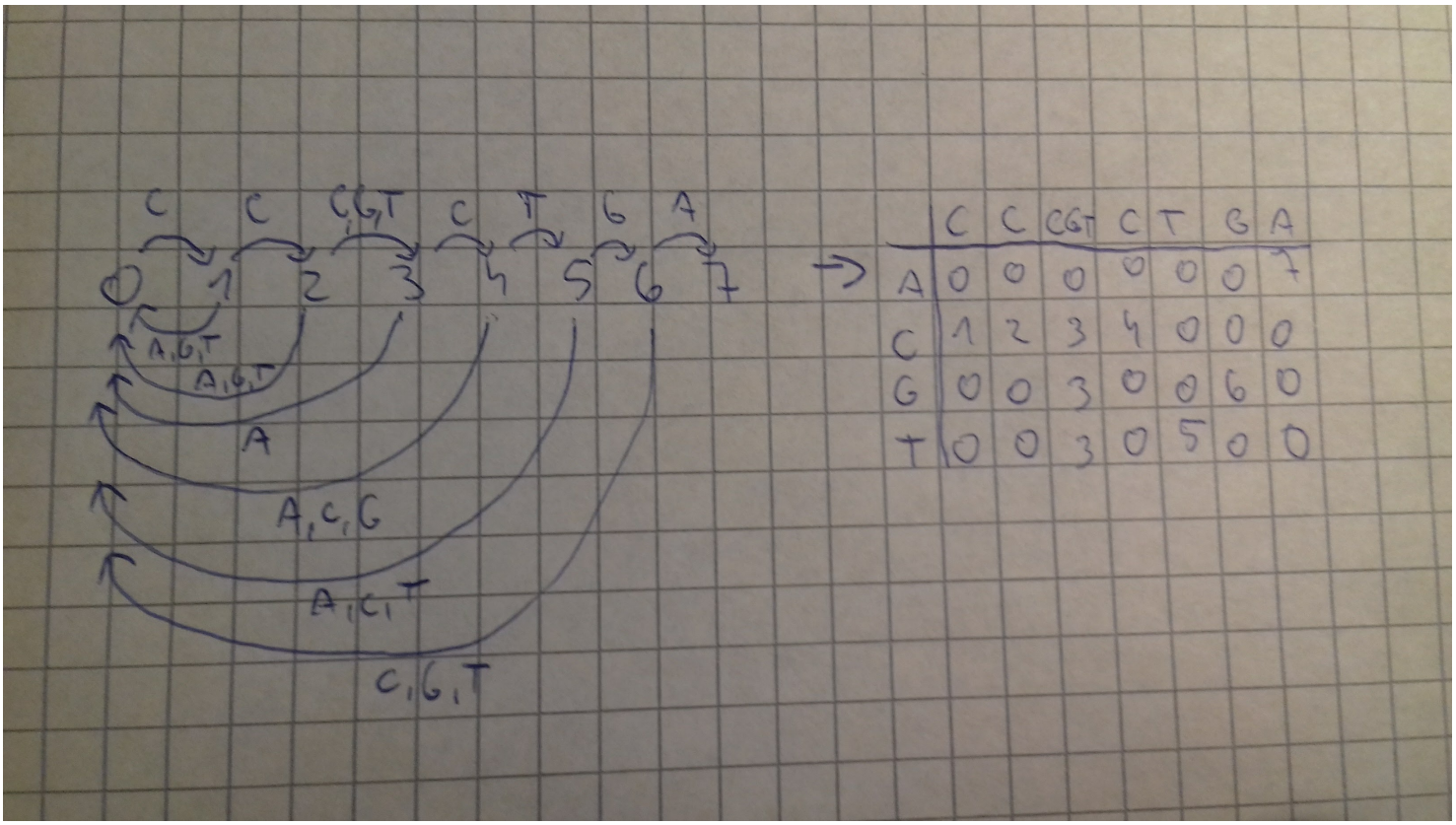
```
4.1)
a) Slika v prilogi
    Če najdemo vzorec, naj se avtomat postavi v stanje m -1, saj je že naslednja črka lahko ponovno cel vzorec.
b)  i) Slika v prilogi
    ii) a)Narišemo tabelo NRDFA
        b)Narišemo prazno tabela DFA
        c)Označiš začetno stanje v DFA(isto kot NDFA)
        d)Za stolpec poiščemo stanja za vse črke možne abecede.
           Če zgeneriramo novo DFA stanje, moramo ta korak ponoviti (situacija 0333 na sliki)
        e)Končno stanje v NDFA je končno stanje v DFA
    iii) KMP zgenerira DFA iz niza, ne iz že narejenega NDFA, tako da ne. Tu delamo od začetka.
c)Kot poslušalec predmeta Podatkovo Rudarjenje moram tu pri prostoru omeniti matrično faktorizacijo.
    To bi uporabili, da stisnemo matriko na dve manjši. - Prostorska kompleksnost.
    Če imamo matriko M, se ta razbije na dve manjši matriki W in H, ki sta pa dimenzij W = n * f in H = f*m
    Predlagam matrično faktorizacijo z majhnim f, saj potrebujemo približen rezultat in ne zelo natančen.
    Iz teh dveh matrik lahko nazaj rekonstruiramo z veliko natančnostjo veliko matriko M in za njuno hranjenje
    porabimo veliko manj prostora. Če se ta zadeva uprablja v bioinformatiki, se v matriki zagotovo nahajajo vzorci,
    katere lahko matrična faktorizacija dobro stisne.
    Časovna kompleksnost:
    Algoritem je lahko tak, da gre skozi vse stolpce matrike(če vrstica >> stolpec -> da se niz nahaja v stolpcu) in
    preiskujemo diagonalo po indeksih. Iteriramo skozi len(tekst)-len(vzorec) polj in štejemo pojavitev na m indeksih.
    O(n**2)
    Lahko bi pa zgradili prefiksno drevo ali prefiksno polje za niz te matrike.
    Potem v tem drevesu gledamo vsa poddrevesa, kjer se začnejo na i-to črko iskanega niza.
    Koda nekaj takega:
    build_prefix_tree(niz)
    max_diag = 0
    for crka in abeceda:
        obisci_poddrevo(crka)
        max_diag += poglej_sosednje crke()
```



	C	C	T	C	G
A	0	0	0	0	0
C	1	2	2	4	4
G	0	0	0	0	5
T	0	0	3	0	0



	C	C	CGT	C	T	G	A
A	0	0	0	0	0	0	7
C	1	2	3	4	0	0	0
G	0	0	3	0	0	6	0
T	0	0	3	0	5	0	0