VAJE 7.

1. Preverite, ali so naslednje matrike aditivne, z uporabo pogoja štirih točk. V primeru, ko je matrika aditivna konstruirajte drevo z uporabo algoritma ADITIVNA FILOGENIJA.

(a)

	A	B	C	D
A	0	10	5	10
B	10	0	9	5
C	5	9	0	8
D	10	5	8	0

(b)

	A	B	C	D	E
A	0	2	7	7	12
B	2	0	7	7	12
C	7	7	0	4	11
D	7	7	4	0	11
E	12	12	11	11	0

Ali je matrika aditivna, lahko še bolj učinkovito preverimo s t.i. pogojem štirih točk:

pogoj štirih točk:

Izmed treh vsot $D_{ij} + D_{k\ell}, D_{ik} + D_{j\ell}, D_{i\ell} + D_{jk}$ sta dve enaki, tretja pa je manjša (ali enaka) od njiju.

Izrek

$n \times n$ matrika D je aditivna natanko tedaj, ko je pogoj štirih točk izpolnjen za vse štiri različne elemente $1 \leq i, j, k, \ell \leq n$.

Zgled: Naj bo D spet naslednja matrika:

	A	B	C	D
A	0	2	4	4
B	2	0	4	4
C	4	4	0	2
D	4	4	2	0

Pogoj je potrebno preveriti za eno samo četverico, $(i, j, k, \ell) = (A, B, C, D)$.

"Izmed treh vsot $D_{AB} + D_{CD}, D_{AC} + D_{BD}, D_{AD} + D_{BC}$ sta dve enaki, tretja pa je manjša (ali enaka) od njiju."

$D_{AB} + D_{CD} = 2 + 2 = 4,$

$D_{AC} + D_{BD} = 4 + 4 = 8,$

$D_{AD} + D_{BC} = 4 + 4 = 8.$

Pogoj je izpolnjen. Matrika je torej aditivna.

Zgled: Naj bo sedaj D naslednja matrika:

	A	B	C	D
A	0	2	2	2
B	2	0	3	2
C	2	3	0	2
D	2	2	2	0

Spet preverimo pogoj le za $(i, j, k, \ell) = (A, B, C, D)$.

"Izmed treh vsot $D_{AB} + D_{CD}, D_{AC} + D_{BD}, D_{AD} + D_{BC}$ sta dve enaki, tretja pa je manjša (ali enaka) od njiju."

$D_{AB} + D_{CD} = 2 + 2 = 4,$

$D_{AC} + D_{BD} = 2 + 2 = 4,$

$D_{AD} + D_{BC} = 2 + 3 = 5.$

Pogoj ni izpolnjen. Matrika torej ni aditivna.

(a)

	A	B	C	D
A	0	10	5	10
B	10	0	9	5
C	5	9	0	8
D	10	5	8	0

• Predp.: ni aditivna.

Gledamo 3 pare razdalj: ABCD

$D_{AB} + D_{CD} = 10 + 8 = 18$

$D_{AC} + D_{BD} = 5 + 5 = 10$

$D_{AD} + D_{BC} = 10 + 9 = 19$

• Izmed treh vsot dve nista enaki, kljub temu da je tretja manjša/enaka od obeh.

• Ni aditivna!

(b)

	A	B	C	D	E
A	0	2	7	7	12
B	2	0	7	7	12
C	7	7	0	4	11
D	7	7	4	0	11
E	12	12	11	11	0

• Predp.: ni aditivna

Sistem diagonal ne deluje! Krneki!

Gledamo ABCD, ABCE, ACDE, ABDE, BCDE (vsak se mora 4x pojavit)

$D_{AB} + D_{CD} = 2 + 4 = 6$

$D_{AC} + D_{BD} = 7 + 7 = 14$

$D_{AD} + D_{BC} = 7 + 7 = 14$

$D_{AB} + D_{CE} = 2 + 11 = 13$

$D_{AC} + D_{BE} = 7 + 12 = 19$

$D_{AE} + D_{BC} = 12 + 7 = 19$

$D_{BC} + D_{DE} = 7 + 11 = 18$

$D_{BD} + D_{CE} = 7 + 11 = 18$

$D_{CD} + D_{BE} = 4 + 12 = 16$

$D_{AC} + D_{DE} = 7 + 11 = 18$

$D_{AD} + D_{CE} = 7 + 11 = 18$

$D_{AE} + D_{DC} = 12 + 4 = 16$

$D_{AB} + D_{DE} = 18$

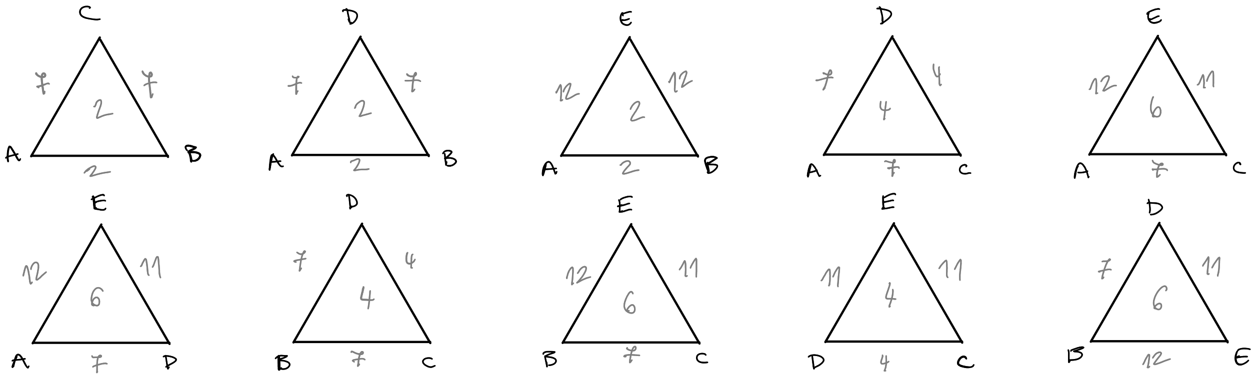
$D_{AD} + D_{BE} = 7 + 12 = 19$

$D_{AE} + D_{BD} = 12 + 7 = 19$

$\Rightarrow$ Je aditivna!

1. Izberemo trojico z najkrajšo razdaljo
2. Določimo $\delta = 1/2 \times$ razdalja iz 1.
3. Odštejemo 2δ vsem razdaljam v matriki, razen diagonalnim + znebimo se ene instance (črke)

1.



sredina = Σ najkrajši dve − tretja

min = 2 in se pojavi 3x ---> naključno izberemo eno trojico, npr. ABC.

2. $\delta = 1/2 \cdot 2 = 1$

Vsem odštejemo 2δ (razen diag.):

3.

	A	B	C	D	E
A	0	0	5	5	10
B	0	0	5	5	10
C	5	5	0	2	9
D	5	5	2	0	9
E	10	10	9	9	0

$\Rightarrow$

Izločimo eno instanco glede na trojico ABC (preverimo če drži):

$$D_{ik} \leq \underline{D_{ij}} + \underline{D_{jk}}$$

$$\Downarrow$$

$$D_{AC} \leq D_{AB} + D_{BC}$$

$$5 \leq \underline{0} + 5$$

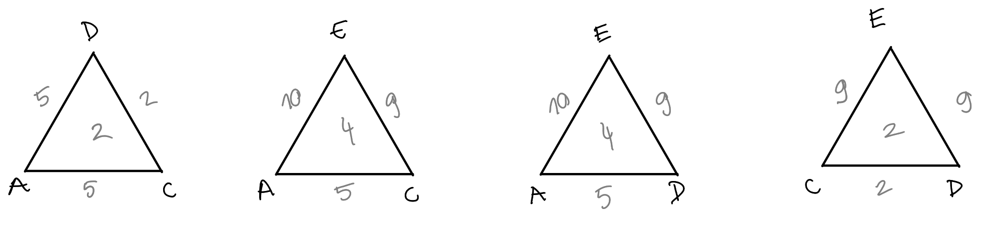
odstranimo vmesnega; za vmesnega vzami tistega, ki je v Δ med večjima vrednostima.

odstranimo B

	A	C	D	E
A	0	5	5	10
C	5	0	2	9
D	5	2	0	9
E	10	9	9	0

Ponovimo korake...

1.



min = 2, spet naključno izberemo ACD.

2. $\delta = 1/2 \cdot 2 = 1$

$$D_{AD} = D_{AC} + D_{CD}$$

$$3 \leq 3 + 0 \Rightarrow$$ odstranimo C:

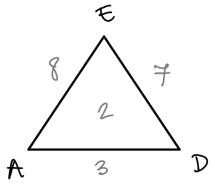
3.

	A	C	D	E
A	0	3	3	8
C	3	0	0	7
D	3	0	0	7
E	8	7	7	0

$\Rightarrow$

	A	D	E
A	0	3	8
D	3	0	7
E	8	7	0

1.



$\Rightarrow$

2. $\delta = \frac{1}{2} \cdot 2 = 2$

3.

	A	D	E
A	0	1	6
D	1	0	5
E	6	5	0

$D_{AE} \leq D_{AD} + D_{DE}$

$6 \leq 1 + 5$

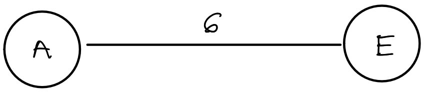
$\Downarrow$

odstranimo D.

	A	E
A	0	6
E	6	0

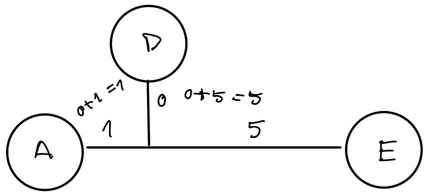
Nimamo več trojic. Začnemo graditi drevo:

Začnemo z zadnjim parom:

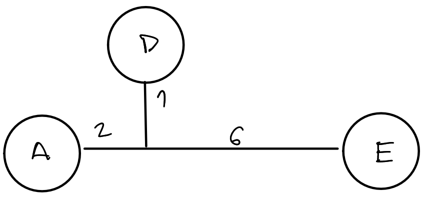


$\Rightarrow$

Dodamo D; upoštevamo razdalje $D_{AD} = 1$ in $D_{DE} = 5$:



Dodamo δ vsaki povezavi: $\delta = 1$

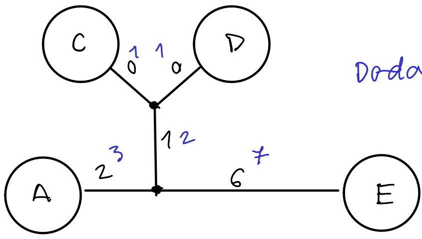


$\Rightarrow$

Vrnemo C; upoštevamo $D_{AC} = 3$, $D_{CD} = 0$, $D_{CE} = 7$

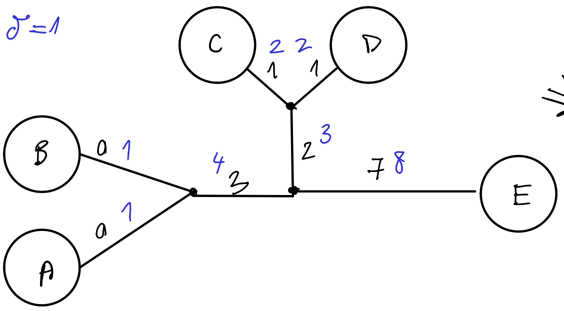
naredimo razvejitev

Dodamo $\delta = 1$



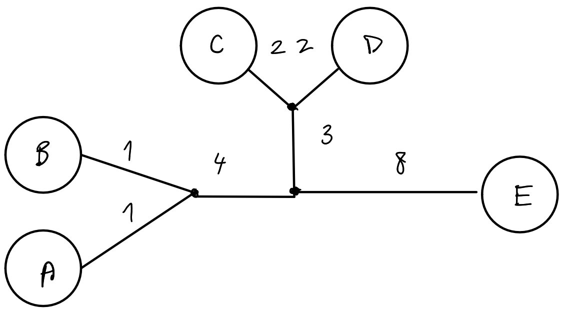
Vrnemo še B; upoštevamo razdalje:
$D_{AB} = 0$, $D_{CB} = 5$, $D_{BD} = 5$, $D_{BE} = 10$

Dodamo $\delta = 1$



$\Rightarrow$

Končno drevo (brez korena):



2. Zgradite POPOLNO FILOGENIJO za naslednjo matriko:

	1	2	3	4	5	6	7	8
A	0	0	1	1	1	0	1	0
B	0	1	1	1	0	0	0	0
C	1	0	0	0	0	1	0	1
D	0	0	1	1	0	0	1	0
E	1	0	0	0	0	0	0	0

Imamo 2 načina, da preverimo ali sploh imamo popolno filogenijo:

I) Primerjamo pare stolpcev, in sicer, za vsak par mora veljati eden izmed pogojev:
   a) enice v enem morajo biti podmn. enic v drugem (npr. $S_8 \subseteq S_6$ ali pa $S_5 \subseteq S_4$)
   b) enice med dvema stolpcema morajo biti na popolnoma različnih mestih (npr. $S_1 \cap S_2 = \emptyset$)

Ta metoda je $O(nm^2)$, obstaja še metoda kazalcev, ki je $O(nm)$:

II) 1. Stolpce razvrstimo nepadajoče glede na število enic (to je $O(nm)$)
   Dodamo stolpec S_0 s samimi enicami (na začetek)
   Za vsako enico v matriki narišemo kazalec na prejšnjo enico v isti vrstici
   2. Izgradimo filogenetsko drevo, če je lastnost disjunktnosti/vsebovanosti izpolnjena
   3. Izboljšamo, označimo drevo

Kako to izgleda za našo matriko:

1.

	1	2	3	4	5	6	7	8
A	0	0	1	1	1	0	1	0
B	0	1	1	1	0	0	0	0
C	1	0	0	0	0	1	0	1
D	0	0	1	1	0	0	1	0
E	1	0	0	0	0	0	0	0
Σ	2	1	3	3	1	1	2	1

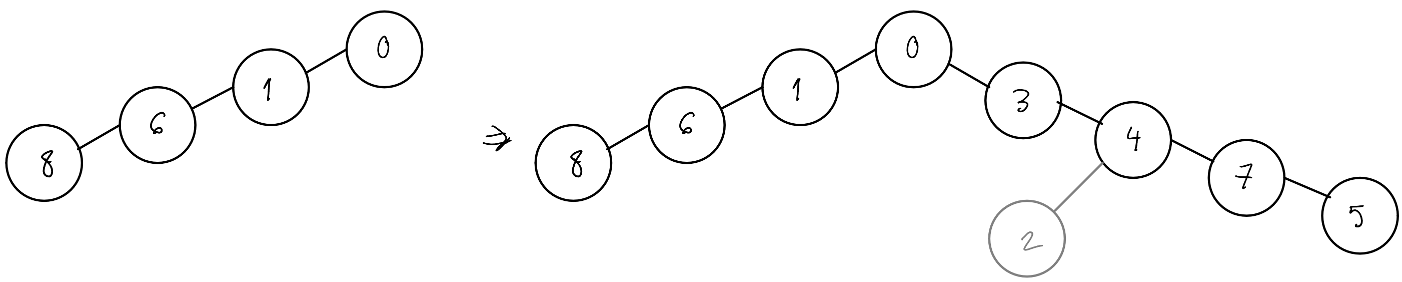
$\Rightarrow$

	S_0	S_3	S_4	S_1	S_7	S_2	S_5	S_6	S_8
A	1 ←	1 ←	1 ←	0	1 ←	0	1	0	0
B	1 ←	1 ←	1 ←	0	0	1	0	0	0
C	1 ←	0	0	1 ←	0	0	0	1 ←	1
D	1 ←	1 ←	1 ←	0	1	0	0	0	0
E	1 ←	0	0	1	0	0	0	0	0

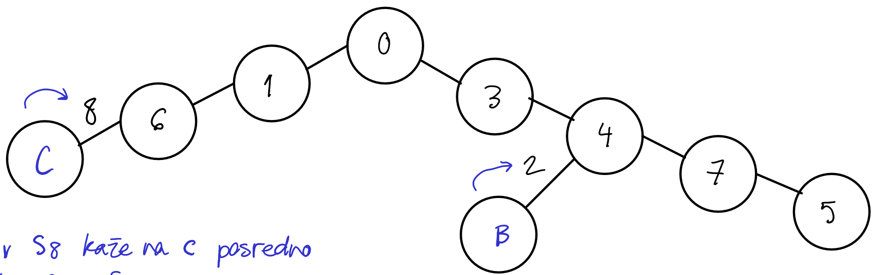
Vse enice enega stolpca morajo kazati na enice v ISTI drug stolpec (npr. v S_7 obe kažeta na S_4).

Pogoj je izpolnjen. Začnemo z drevesom.

2. Začnemo z najbolj oddaljenim stolpcem. Kazalci nakazujejo na povezavo med stolpci:

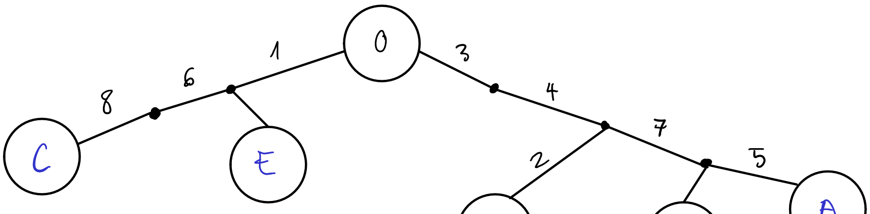


3. Vstavljamo znake v drevo, tako da premaknemo imena stolpcev na povezave:



enica v S_8 kaže na C posredno preko S_6 in S_1

enica v S_2 kaže na B posredno preko S_4 in S_3



ena enica v S1 kaže na C, druga pa na E

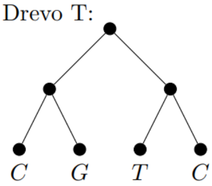
Dobili smo popolno filogenijo, kjer so povezave označene z "znaki", listi pa s predmeti
(motor, št. koles) (avto, bicikelj...)

3. Rešite mali problem varčnosti za naslednje podatke:

(a) Cene mutacij:

δ	A	C	G	T
A	0	3	4	9
C	3	0	2	4
G	4	2	0	4
T	9	4	4	0

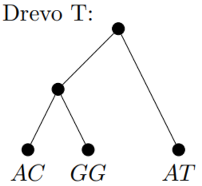
Drevo T:



(b) Cene mutacij:

δ	A	C	G	T
A	0	1	3	3
C	1	0	3	3
G	3	3	0	1
T	3	3	1	0

Drevo T:



Filogenetsko drevo v tej metodi temelji na poravnavi zaporedij. Iščemo drevo z najmanjšo stopnjo evolucije (najmanj evolucijskih dogodkov). Vsak dogodek ima neko ceno. Iščemo evolucijski scenarij z najmanjšo skupno ceno ⇒ najbolj varčno drevo.

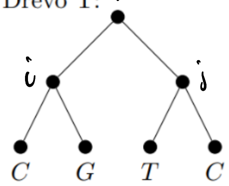
Veliki problem varčnosti: iščemo drevo in zaporedja za notranje točke drevesa (NP!).

Mali problem varčnosti: drevo je znano, iščemo le zaporedja za notranje točke.

(a) Cene mutacij:

δ	A	C	G	T
A	0	3	4	9
C	3	0	2	4
G	4	2	0	4
T	9	4	4	0

Drevo T: k



Vsakemu listu določimo tabelo:

C:

A	C	G	T
∞	0	∞	∞

G:

A	C	G	T
∞	∞	0	∞

T:

A	C	G	T
∞	∞	∞	0

0 če je enak znak, ∞ sicer.

Za vsako notranjo točko tudi določimo tabelo:

i)

C:	∞	0	∞	∞
A:	0	3	4	9
	∞	3	∞	∞

G:	∞	∞	0	∞
A:	0	3	4	9
	∞	∞	4	∞

3 + 4 = 7

C:	∞	0	∞	∞
C:	3	0	2	4
	∞	0	∞	∞

G:	∞	∞	0	∞
C:	3	0	2	4
	∞	∞	2	∞

0 + 2 = 2

C:	∞	0	∞	∞
G:	4	2	0	4
	∞	2	∞	∞

G:	∞	∞	0	∞
G:	4	2	0	4
	∞	∞	0	∞

2 + 0 = 2

C:	∞	0	∞	∞
T:	9	4	4	0
	∞	4	∞	∞

G:	∞	∞	0	∞
T:	9	4	4	0
	∞	∞	4	∞

4 + 4 = 8

Vzamemo vrstico iz matrike in prištejemo razlike (prištejemo eno drugi) Vzamemo dve najmanjši vrednosti. Če je več enakih, izberemo naključno.

i:

A	C	G	T
7	2	2	8

j:

A	C	G	T
12	4	6	4

k:

A	C	G	T
13	6	8	10

Za k vzamemo zdaj tabeli za i in j ...

j)

T:	∞	∞	∞	0
A:	0	3	4	9
	∞	∞	∞	9

C:	∞	0	∞	∞
A:	0	3	4	9
	∞	3	∞	∞

Za C so iste vrednosti kot prej

9 + 3 = 12

T:	∞	∞	∞	0
C:	3	0	2	4
	∞	∞	∞	4

C:	∞	0	∞	∞
C:	3	0	2	4
	∞	0	∞	∞

4 + 0 = 4

T:	∞	∞	∞	0
G:	4	2	0	4
	∞	∞	∞	4

C:	∞	0	∞	∞
G:	4	2	0	4
	∞	2	∞	∞

4 + 2 = 6

T:	∞	∞	∞	0
T:	9	4	4	0
	∞	∞	∞	0

C:	∞	0	∞	∞
T:	9	4	4	0
	∞	4	∞	∞

0 + 4 = 4

k)

i:	7	2	2	8
A:	0	3	4	9
	7	5	6	17

j:	12	4	6	4
A:	0	3	4	9
	12	7	10	13

$5 + 7 = 13$

i:	7	2	2	8
C:	3	0	2	4
	10	2	4	12

j:	12	4	6	4
C:	3	0	2	4
	15	4	8	8

$2 + 4 = 6$

i:	7	2	2	8
G:	4	2	0	4
	11	4	2	12

j:	12	4	6	4
G:	4	2	0	4
	16	6	6	8

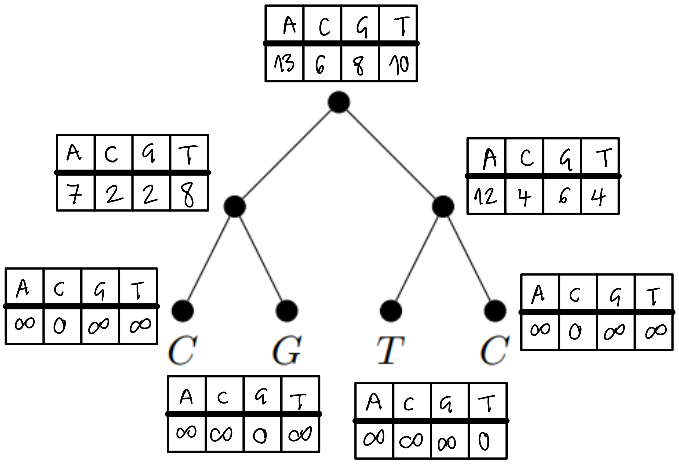
$2 + 6 = 8$

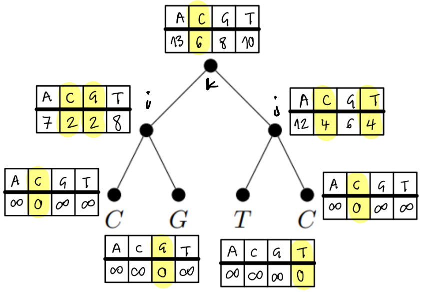
i:	7	2	2	8
T:	9	4	4	0
	16	6	6	8

j:	12	4	6	4
T:	9	4	4	0
	21	8	10	4

$6 + 4 = 10$

Vsaki točki moramo določiti en znak. Vzamemo tistega, ki ima najmanjšo vrednost v tabeli



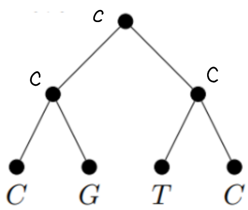


k: očitno določimo C.

i: izberemo C, saj je starš točke tudi C (izbirali smo med C in G).

j: spet izberemo C.

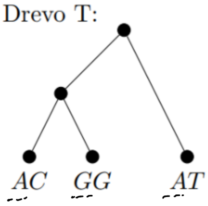
Končno drevo:



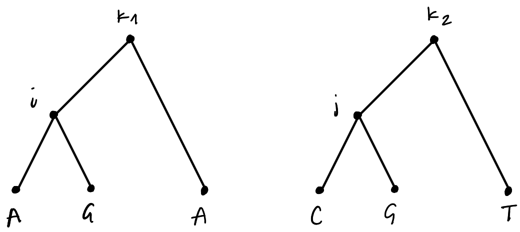
(b) Cene mutacij:

δ	A	C	G	T
A	0	1	3	3
C	1	0	3	3
G	3	3	0	1
T	3	3	1	0

Drevo T:



V tem primeru ne moremo imeti dva znaka v listih, zato naredimo več dreves:



Zdaj spet za vsako drevo določimo znake za vozlišča:

	A	C	G	T
C:	∞	0	∞	∞
G:	∞	∞	0	∞
T:	∞	∞	∞	0
A:	0	∞	∞	∞
i:	3	4	3	4
j:	4	3	3	4

i)

A:	0	∞	∞	∞	G:	∞	∞	0	∞	
A:	0	1	3	3		0	1	3	3	
	0	∞	∞	∞		∞	∞	3	∞	= 3

A:	0	∞	∞	∞	G:	∞	∞	0	∞	
C:	1	0	3	3		1	0	3	3	
	1	∞	∞	∞		∞	∞	3	∞	= 4

A:	0	∞	∞	∞	G:	∞	∞	0	∞	
G:	3	3	0	1		3	3	0	1	
	3	∞	∞	∞		∞	∞	0	∞	= 3

A:	0	∞	∞	∞	G:	∞	∞	0	∞	
T:	3	3	1	0		3	3	1	0	
	3	∞	∞	∞		∞	∞	1	∞	= 4

j)

C:	∞	0	∞	∞	G:	∞	∞	0	∞	
A:	0	1	3	3		0	1	3	3	
	∞	1	∞	∞		∞	∞	3	∞	= 4

C:	∞	0	∞	∞	G:	∞	∞	0	∞	
C:	1	0	3	3		1	0	3	3	
	∞	0	∞	∞		∞	∞	3	∞	= 3

C:	∞	0	∞	∞	G:	∞	∞	0	∞	
G:	3	3	0	1		3	3	0	1	
	∞	3	∞	∞		∞	∞	0	∞	= 3

C:	∞	0	∞	∞	G:	∞	∞	0	∞	
T:	3	3	1	0		3	3	1	0	
	∞	3	∞	∞		∞	∞	1	∞	= 4

k1)

i:	3	4	3	4
A:	0	1	3	3
	3	5	6	7

A:	0	∞	∞	∞
	0	1	3	3
	0			

= 3

i:	3	4	3	4
C:	1	0	3	3
	4	4	6	7

A:	0	∞	∞	∞
	1	0	3	3
	1			

= 5

i:	3	4	3	4
G:	3	3	0	1
	6	7	3	5

A:	0	∞	∞	∞
	3	3	0	1
	3			

= 6

i:	3	4	3	4
T:	3	3	1	0
	6	7	4	4

A:	0	∞	∞	∞
	3	3	1	0
	3			

= 7

k2)

j:	4	3	3	4
A:	0	1	3	3
	4	4	6	7

T:	∞	∞	∞	0
	0	1	3	3
				3

= 7

j:	4	3	3	4
C:	1	0	3	3
	5	3	6	7

T:	∞	∞	∞	0
	1	0	3	3
				3

= 6

j:	4	3	3	4
G:	3	3	0	1
	7	6	3	5

T:	∞	∞	∞	0
	3	3	0	1
				1

= 4

j:	4	3	3	4
T:	3	3	1	0
	7	6	4	4

T:	∞	∞	∞	0
	3	3	1	0
				0

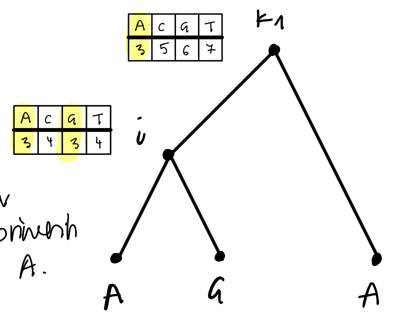
= 4

k1:

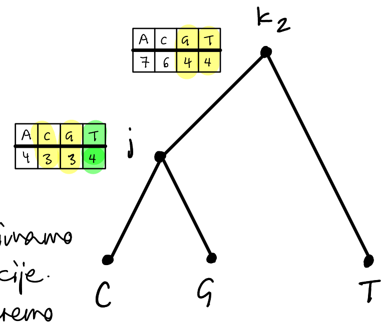
A	C	G	T
3	5	6	7

k2:

A	C	G	T
7	6	4	4



Tu je v obeh primerih vreda A.



Tu pa imamo dve opcije. Če izberemo G, izberemo G dvakrat.

Če se odločimo za T, pogledamo nazaj v tabelo za k2: min se pojavi za par T-G ali T-T, zato imamo 3 izbire za točke.

Na koncu združimo drevesa nazaj skupaj: