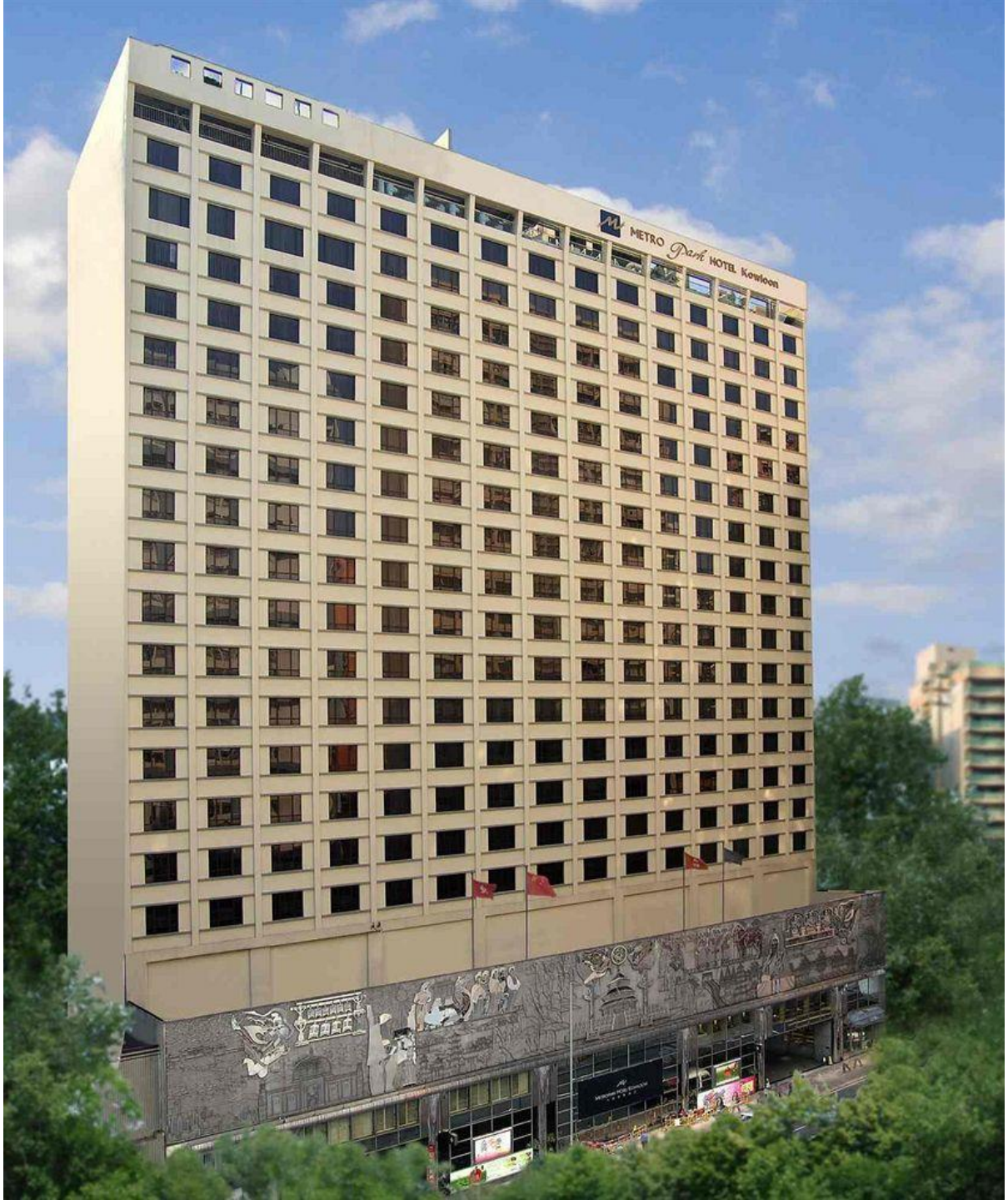
Koja životinja nam je donela SARS?

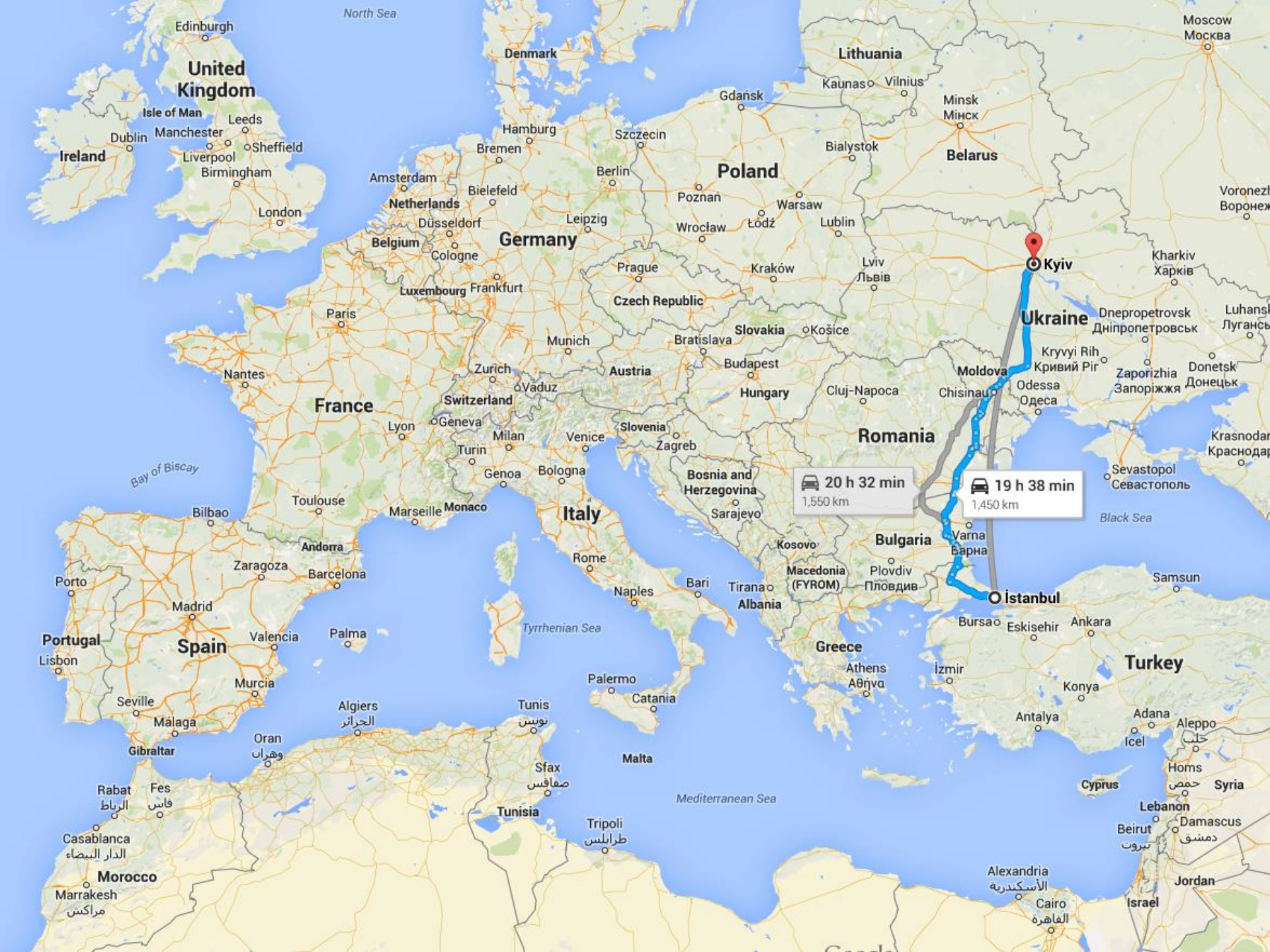
Rekonstrukcija evolutivnih stabala

Bioinformatics Algorithms:
an Active Learning Approach
~Poglavlje 7~

Pregled

- **Izbijanje epidemije**
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- Neighbour-Joining algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

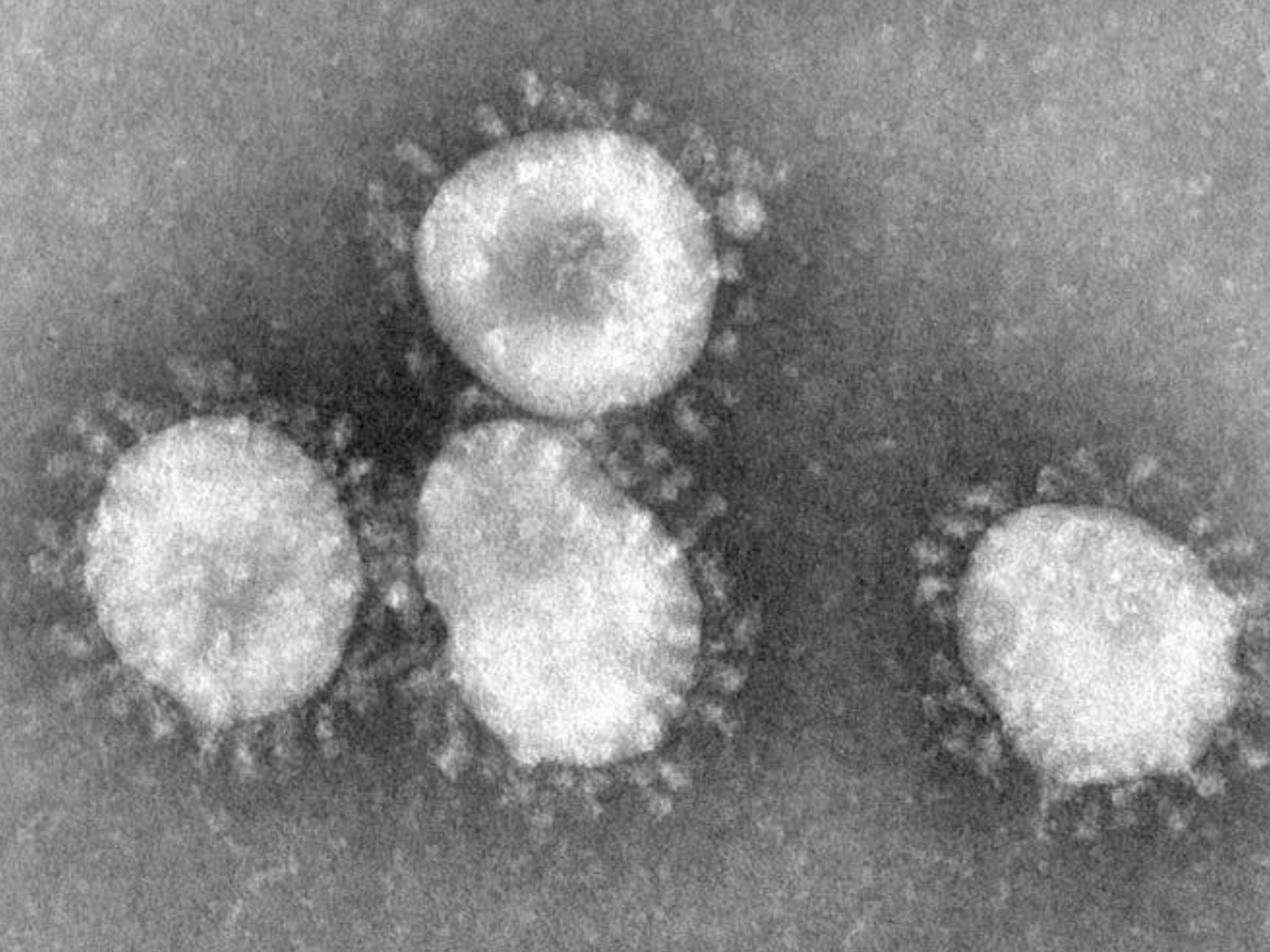

METRO Park HOTEL Kowloon


20 h 32 min
1,550 km
19 h 38 min
1,450 km
Kyiv
İstanbul
North Sea
Bay of Biscay
Tyrrhenian Sea
Mediterranean Sea
Black Sea
United Kingdom
Ireland
Isle of Man
Edinburgh
Dublin
Manchester
Leeds
Liverpool
Sheffield
Birmingham
London
Denmark
Netherlands
Amsterdam
Hamburg
Bremen
Belgium
Düsseldorf
Cologne
Germany
Luxembourg
Frankfurt
Paris
France
Nantes
Switzerland
Zurich
Vaduz
Geneva
Lyon
Milan
Turin
Venice
Genoa
Bologna
Monaco
Marseille
Toulouse
Bilbao
Andorra
Zaragoza
Barcelona
Porto
Madrid
Portugal
Lisbon
Spain
Valencia
Palma
Murcia
Seville
Málaga
Gibraltar
Oran
Algiers
Tunis
Sfax
Tunisia
Malta
Rabat
Fes
Casablanca
Morocco
Marrakesh
Tripoli
Palermo
Catania
Italy
Rome
Naples
Bari
Albania
Tirana
Kosovo
Macedonia (FYROM)
Bosnia and Herzegovina
Sarajevo
Slovenia
Zagreb
Austria
Munich
Czech Republic
Prague
Slovakia
Bratislava
Košice
Hungary
Budapest
Poland
Gdańsk
Szczecin
Berlin
Poznań
Warsaw
Wrocław
Łódź
Lublin
Kraków
Belarus
Minsk
Bialystok
Lithuania
Vilnius
Kaunas
Moscow
Ukraine
Lviv
Kharkiv
Dnepropetrovsk
Kryvyi Rih
Zaporizhia
Donetsk
Moldova
Chisinau
Odessa
Cluj-Napoca
Romania
Bulgaria
Varna
Plovdiv
Greece
Athens
Turkey
İzmir
Bursa
Eskisehir
Ankara
Konya
Antalya
Adana
Aleppo
Icel
Samsun
Sevastopol
Krasnodar
Cyprus
Syria
Lebanon
Beirut
Damascus
Jordan
Israel
Alexandria
Cairo


Ireland
1 case
China
Canada
29 cases
United States
1 case
Guangdong
Province
Metropole
Hotel
Hong Kong
Special
Administrative
Region
195 cases
Philippine
Sea
Vietnam
58 cases
Bay
of
Bengal
Arabian
Sea
South
China
Sea
Singapore
71 cases
Indian

Širenje ?????

Širenje SARS-a



28261 aauaauacug cgucuugguu cacagcucuc acucagcaug gcaaggagga acuuagauuc
28321 ccucgaggcc agggcguucc aaucaacacc aauagugguc cagaugacca aauuggcuac
28381 uaccgaagag cuacccgacg aguucguggu ggugacggca aaaugaaaga gcucagcccc
28441 agaugguacu ucuauuaccu aggaacuggc ccagaagcuu cacuucccua cggcgcuaac
28501 aaagaaggca ucguaugggu ugcaacugag ggagccuuga auacacccaa agaccacauu
28561 ggcacccgca auccuaauaa caaugcugcc accgugcuac aacuuccuca aggaacaaca
28621 uugccaaaag gcuucuacgc agagggaagc agaggcggca gucaagccuc uucucgcucc
28681 ucaucacgua gucgcgguaa uucaagaaau ucaacuccug gcagcaguag gggaaauucu
28741 ccugcucgaa uggcuagcgg agguggugaa acugcccucg cgcuauugcu gcuagacaga
28801 uugaaccagc uugagagcaa aguuucuggu aaaggccaac aacaacaagg ccaaacuguc
28861 acuaagaaau cugcugcuga ggcaucuaaa aagccucgcc aaaaacguac ugccacaaaa
28921 caguacaacg ucacucaagc auuugggaga cgugguccag aacaaaccca aggaaauuuc
28981 ggggaccaag accuaaucag acaaggaacu gauuacaaac auuggccgca aauugcacaa
29041 uuugcuccaa gugccucugc auucuuugga augucacgca uuggcaugga agucacaccu
29101 ucgggaacau ggcugacuua ucauggagcc auuaaauugg augacaaaga uccacaauuc
29161 aaagacaacg ucauacugcu gaacaagcac auugacgcau acaaaacauu cccaccaaca
29221 gagccuaaaa aggacaaaaa gaaaaagacu gaugaagcuc agccuuugcc gcagagacaa
29281 aagaagcagc ccacugugac ucuucuuccu gcggcugaca uggaugauuu cuccagacaa
29341 cuucaaaauu ccaugagugg agcuucugcu gauucaacuc aggcauaaac acucaugaug
29401 accacacaag gcagaugggc uauguaaacg uuuucgcaau uccguuuacg auacauaguc
29461 uacucuugug cagaaugaau ucucguaacu aaacagcaca aguagguuua guuaacuuua
29521 aucucacaua gcaaucuuua aucaaugugu aacauuaggg aggacuugaa agagccacca
29581 cauuuucauc gaggccacgc ggaguacgau cgaggguaca gugaauaaug cuagggagag
29641 cugccuauau ggaagagccc uaauguguaa aauuaauuuu aguagugcua uccccaugug
29701 auuuuaauag cuucuuagga gaaugacaaa aaaaaaaaaa aaaaaaaaaa a

28261 aauaauacug cgucuugguu cacagcucuc acucagcaug gcaaggagga acuuagauuc
28321 ccucgaggcc agggcguucc aaucaacacc aaCaguggUc cagaugacca aauuggcuac
28381 uacAgaagag cuacccgacg aguucguggu ggugacggca aaaugaaaga gcucagcccc
28441 agaugguacu ucuauuaccu aggaacuggc ccagaagcuu cacuucccua cggcgcuaac
28501 aaagaaggca ucguaugggu ugcaacugag ggagccuuga auacCcccaa agaccacauu
28561 ggcacccgca auccuaauaa caaugcugcc accgugcuac aacuuccuca aggaacaaca
28621 uugccaaaag gcuucuacgc agagggaagc agaggcggca gucaagccuc uucucgcucc
28681 ucaucacgua gucgcgguaa uucaagaaau ucaacuccug gcagcaguag gggaaauucu
28741 ccugcucgaa uggcuagcgg agguggugaa acugcccucg cgcuauugcu gcuagacaga
28801 uugaaccagc uugagagcaa aguuucuggu aaaggccaac aacaacaagg ccaaacuguc
28861 acuaagaaau cugcugcuga ggcaucuaaa aagccucgcc aaaaacguac ugccacaaaa
28921 caguacaacg ucacucaagc auuugggaga cgugguccag aacaaaccca aggaaauuuc
28981 ggggaccaag acUuaaucag acaaggaacu gauuacaaac auuggccgca aauugcacaa
29041 uuugcuccaa gugccucugc auucuuugga augucacgca uuggcaugga agucacaccu
29101 ucgggaacau ggcugacuua ucauggagcc auuaaauugg augacaaaga uccacaauuc
29161 aaagacaacg ucauacugcu gaacaagcac auugacgcau acaUaacauu cccaccaaca
29221 gaUccuaaaa aggacaaaaa gaaaaagacu gaugaagcuc agccuuugcc gcagagacaa
29281 aagaagcagc ccacugugac ucuucuuccu gcggcugaca uggaugauuu cuccagacaa
29341 cuucaaaauu ccaugagugg agcuucugcu gauucaacuc aggcauaaac acucaugaug
29401 accacacaag gcagaugggc uauguaaacg uuuucgcaau uccguuuacg auacauaguc
29461 uacucuugug cagaaugaau ucucguaacu aaacagcaca aguagguuua guuaacuuua
29521 aucucacaua gcaaucuuua aucaaGgugu aacauuaggg aggacuugaa agagccacca
29581 cauuAucauc gaggccacgc ggaguacgau cgaggguaca gugaauaaug cuagggagag
29641 cugccuauau ggaagagccc uaauguguaa aauuaauuuu aguagugcua uccccaugug
29701 auuuuaauag cuucuuagga gaaugacaaa aaaaaaaaaa aaaaaaaaaa a

28261 aauaauacug cgucuugguu cacagcucuc acucagcaug gcaaggagga acuuagauuc
28321 ccucgaggcc agggcguucc aaucaacacc aaCaguggUc cagaugacca aauuggcuac
28381 uacAgaagag cuacccgacg aguucguggu ggugacggca aaaugaaaga gcuGagcccc
28441 agaugguacu ucuauuaccu aggaacuggc ccagaagcuu cacuucccua cggcgcuaac
28501 aaagaaggca ucguauUggu ugcaacugag ggagccuuga auacCcccaa agaccacauu
28561 ggcacccgca auccuaauaa caaugcugcc accgugcuac aacuuccuca aggaacaaca
28621 uugccaaaag gcuucAacgc agagggaagc agaggcggca gucaagccuc uucucgcucc
28681 ucaucacgua gucgcgguaa uucaagaaau ucaacuccug gcagcaguag gggaaauucu
28741 cGugcucgaa uggcuagcgg agguggugaa acugcccucg cgcuauugcu gcuagacaga
28801 uugaaccagc uugagagcaa aguuucuggu aaaggccaac aacaacaagg ccaGacuguc
28861 acuaagaaau cugcugcuga ggcaucuCaa aagccucgcc aaaaacguac ugccacaaaa
28921 caguacaacg ucacucaagc auuugggaga cgugguccag aacaaaccca aggaaauuuc
28981 ggggaccaag acUuaaucag acaaggaacu gauuacaaac auuggccgca aauugcacaa
29041 uuugcuccaa gugccucugc auucuuugga auguUacgca uuggcaugga agucacaccu
29101 ucgggaacau ggcugacuua ucauggagcc auuaaauugg augacaaaga uccacaauuc
29161 aaagacaacg ucauacugcu gaacaagcac auugacgcau acaUaacauu cccaccaaca
29221 gaUccuaaaa aggacaaaaa gaaaaagacu gaugaagcuc agccuuugcc gcagagacaa
29281 aagaagcagc ccacugugac ucuucuuccu gcggcugaca uggaugauuu cuccagacaa
29341 cuucaaaauu ccaugagugg aAcuucugcu gauucaacuc agUcauaaac acucaugaug
29401 accaAacaag gcagaugggc uauguaaacg uuuucgcaau uccguuuacg auacauaguc
29461 uacucuugug cagaaugaau ucucguaacu aaacagcaca aguagguuua guuaacAuua
29521 aucucacaua gcaaucuuua aucaaGgugu aacauuaggg aggacuugaa agagccacca
29581 cauuAucauc gaggccacgc ggaguacgau cgaggguaca gugaauaaug cuagggagag
29641 cugccuauau ggaagagccc uaauguguaa aauuaauuuu aguagugcua uccccaugug
29701 auuuuaauag cuucuuagga gaaugacaaa aaaaaaaaaa aaaaaaaaaa a

28261 Cauaauacug cgucuugguu cacaUcucuc acucagcaug gcaaggagga acuuagauuc
28321 ccucgaggcc agggcguucc aaucaacacc aaCagugguc cagaugacca aauuggcuac
28381 uacAgaagag cuacccgacg aguucguggu ggugacggca aaaugaaaga gcuGagcccc
28441 agaugguacu ucuauuaccu aggaacuggc ccagaagcuu cacuucccua cggcgcuaac
28501 aaagaaggca ucguauUggu ugcaacugag ggagccuuga auacCcccaa agaccacauu
28561 ggcacccgca auccuaauaa caaugcugcc accgugcuac aacuuccuca aggaacaaca
28621 uugccaaaag gcuucAacgc agagggaagc agaggcggca gucaagccuc uucucgcucc
28681 ucaucacgua gucgcgguaa uucaagUaau ucaacuccug gcaAcaguag gggaaauucu
28741 cGugcucgaa uggcuagcgg agguggugaa acugcccucg cgcuauugcu gcuagacaga
28801 uugaaccagc uugagagcaa aguuucuggu aaaggccaac aacaacaagg ccaGacuguc
28861 acuaagaaau cugcugcuga ggcaucuCaa aagccucgcc aaaaacguac ugccacaaaa
28921 caguacaacg ucacucaagc auuugggaga cgugguccag aacaaaccca aggaaauuuc
28981 ggggaccaag acUuaaucag acaaggaacu gauuacaaac auuggccgca aauugcacaa
29041 uuuAcuccaa gugccucugc auucuuugga auguUacgca uuggcaugga agucacaccu
29101 ucgggaacau ggcugacuua ucauggagcc auuaaauugg augacaaaga uccacaauuc
29161 aaagacaacg ucauacugcu gaacaagcac auugacgcau acaUaacauu cccaccaaca
29221 gaUccuaaaa aggGcaaaaa gaaaaagacu gaugaagcuc agccuuugcc gcagagacaa
29281 aagaagcagc ccacugugac ucuucuuccu gcggcugaca uggaugauuu cuccaUacaa
29341 cuucaaaauu ccaugagugg aAcuucugcu gauucaacuc agUcauaaac acucaugaug
29401 accaAacaag gcagaugggc uauguaaacg uuuucgcaau uccguuuacg auacauaguc
29461 uacucuugug cagaaugaau ucucguaacu aaacagcaca aguagguuua guuaacAuua
29521 aucucacaua gcaaucuuua aucaaGgugu aacauuaggg aggacuugaa agagccacca
29581 cauuAucauc gaggccacgc ggaguacgau cgaggguaca gugaauaaug cuagggagag
29641 cugccuauau ggaagagccc uaauguguaa aauuaauuuu aUuagugcua uccccaugug
29701 auuuuaauag cuucuuagga gaaugacaaa aaaaaaaaaa aaaaaaaaaa a

28261 Cauaauacug cgucuugguu cacaUcucuc acucagcaug gcaaggagga acuuagauuc
28321 ccucgaggcc agggcguCcc aaucaacacc aaCagugguc cagaugacca aauuggcuac
28381 uacAgaagag cuacccgacg aguucguggu ggugacggca aaaugaaaga gcuGagcccc
28441 agaugguacu ucuauuaccu aggaacuggc ccagaagcuu cacuucccua cggcgcuaac
28501 aaagaaggca ucguauUggu ugcaacuUag ggagccuuga auacCGccaa agaccacauu
28561 ggcacccgca auccuaauaa caaugcugcc accgugcuac aacuuccuca aggaacaaca
28621 uugccaaaag gcuucAacgc agagggaagc agaggcggca gucaagccuc uucucgcucc
28681 ucaucacgua gucgcgguaa uucaagUaau ucaacuccug gcaAcaguag gggaaauucu
28741 cGugcucgaa uggcuagcgg agguggugaa acugcccucg cgcuauugcu gcuagacaga
28801 uugaaccagc uugagagcaa aguuucuggu aaaggccaac aacaacaagg ccaGacuguc
28861 acuaagaaau cugcugcuga ggcaucuCaa aagccucgcc aaaaacguac ugccacaaaa
28921 caguacaacg ucacucCagc auuugggaga cgugguccag aacaaaccca aggaaauuuc
28981 ggggaccaag acUuaaucag acaaggaacu gauuacaaac auuggccgca aauugcacaa
29041 uuuAcuccaC gugccucugc auucuuugga auguUacgca uuggcaugga agucacaccu
29101 ucgggaacau ggcugacuua ucauggagcc auuaaauugg augacaaaga uccacaauuc
29161 aaagacaacg ucauacugcu gaacaagcac auugacgcau acaUaaGauu cccaccaaca
29221 gaUccuaaaa aggGcaaaaa gaaaaagacu gaugaagcuc agccuuugcc gcagagacaa
29281 aagaagcagc ccacugugac ucuucuuccu gcggcugaca uggaugauuu cuccaUacaa
29341 cuucaaaauu ccaugagugg aAcuucugcu gauucaacuc agUcauaaac acucaugaug
29401 accaAacaag gcagaugggc uauguaaacg uuuucgcaau uccguuuacg auacauaguc
29461 uacucuugug cagaaugaau ucucguaacu aaacagcaca aguagguuua guuaacAuua
29521 aucucacaua gcaaucuuua aucaaGgugu aacauuaggg aggacuugaa agagccacca
29581 cauuAucauc gaggccacgc ggaguacgau cgaggguaca gugaauaaug cuaAggagag
29641 cugccuauau ggaagagccc uaauAuguaa aauuaauuuu aUuagugcua uccccaugug
29701 auuuuaauag cuucuuagga gaaugacaaa aaaaaaaaaa aaaaaaaaaa a

Pitanja u vezi sa SARS-om

- Koja životinja nam je donela SARS?

Pitanja u vezi sa SARS-om

- Koja životinja nam je donela SARS?
- Kako smo se prvobitno zarazili?

Pitanja u vezi sa SARS-om

- Koja životinja nam je donela SARS?
- Kako smo se prvobitno zarazili?
- Kako se SARS širio po svetu?

Pitanja u vezi sa SARS-om

- Koja životinja nam je donela SARS?
- Kako smo se prvobitno zarazili?
- Kako se SARS širio po svetu?
- Sva ova pitanja spadaju u domen filogenetske analize koja se bavi rekonstrukcijom evolutivnih stabala

must be that one
or other than should be
may living or now

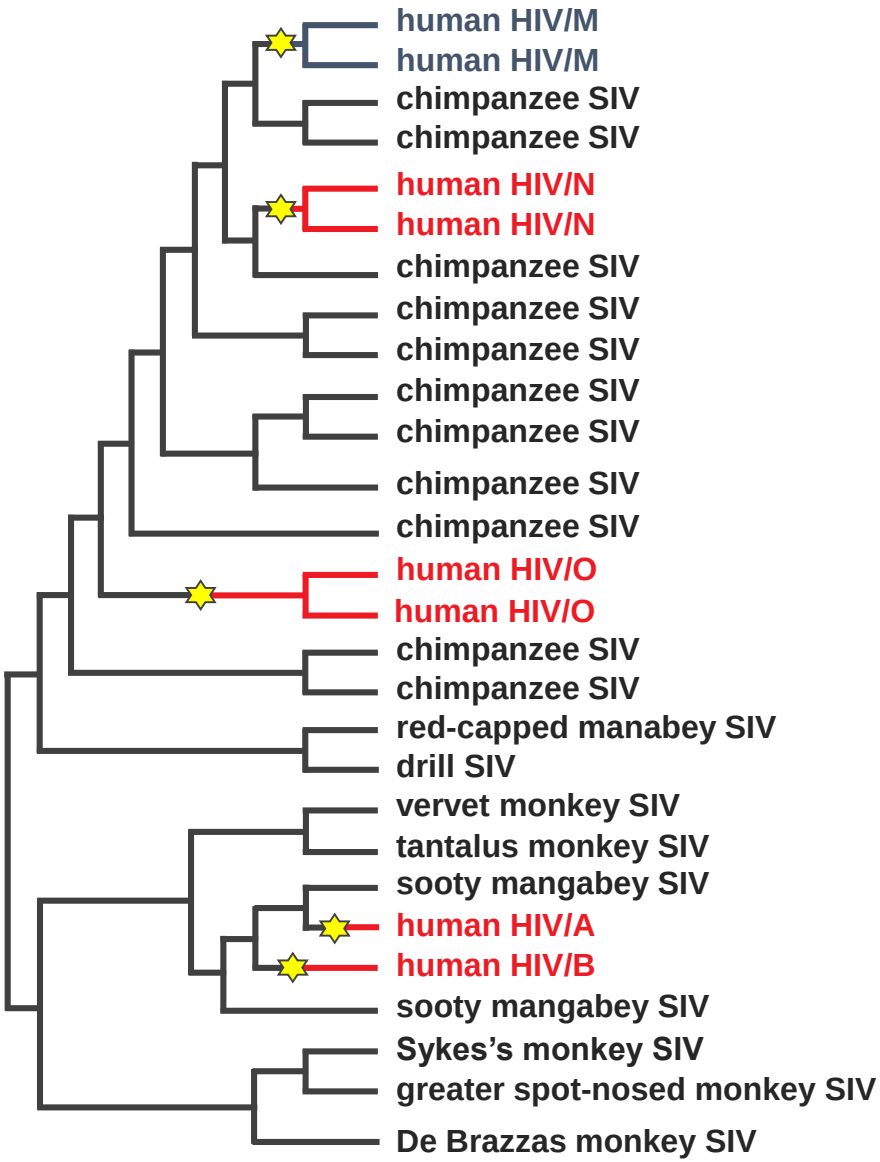
do this & have many species in
some genera (as is). Represent
other state.

Evolutivno stablo za HIV

SIVs (majmuni)

HIV (ljudi)

humana infekcija



Pregled

- Izbijanje epidemije
- **Transformacija matrice rastojanja u evolutivno stablo**
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- Neighbour-Joining algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

Konstrukcija matrice rastojanja

Vrsta	Poravnanje
Šimpanza	ACGTAGGCCT
Človek	ATGTAAGACT
Foka	TCGAGAGCAC
Kit	TCGAAAGCAT

Konstrukcija matrice rastojanja

$D_{i,j}$ = broj različitih simbola u i-tom i j-tom redu višestrukog poravnanja

Vrsta	Poravnanje	Matrica rastojanja			
		Šimpanza	**Čovek**	**Foka**	**Kit**
Šimpanza	ACGTAGGCCT	0	3	6	4
Čovek	ATGTAAGACT	3	0	7	5
Foka	TCGAGAGCAC	6	7	0	2
Kit	TCGAAAGCAT	4	5	2	0

Konstrukcija matrice rastojanja

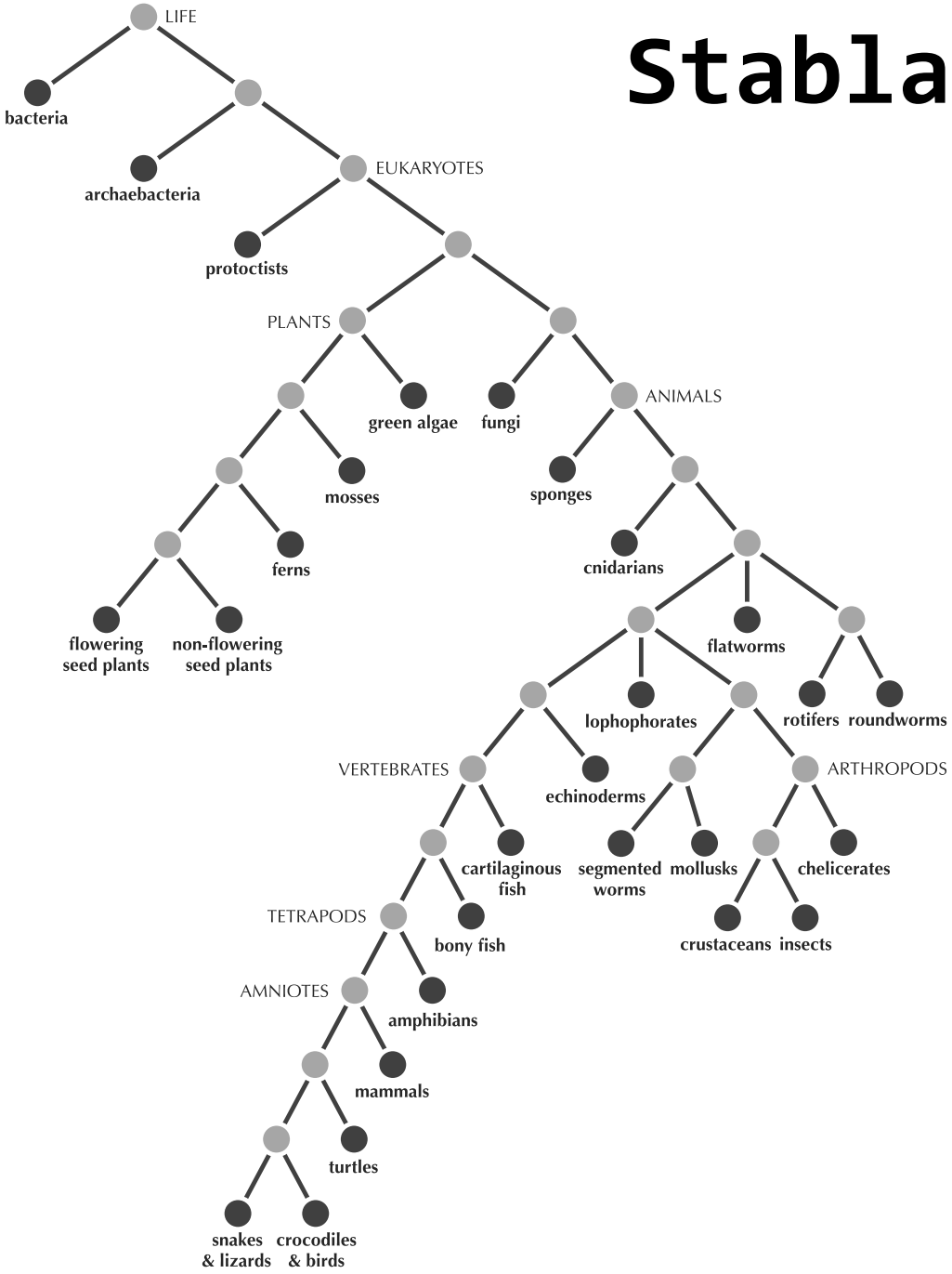
$D_{i,j}$ = broj različitih simbola u i-tom i j-tom redu višestrukog poravnanja

Vrsta	Poravnanje	Matrica rastojanja			
		Šimpanza	**Čovek**	**Foka**	**Kit**
Šimpanza	ACGTAGGCCT	0	**3**	6	4
Čovek	ATGTAAGACT	**3**	0	7	5
Foka	TCGAGAGCAC	6	7	0	2
Kit	TCGAAAGCAT	4	5	2	0

Spike protein

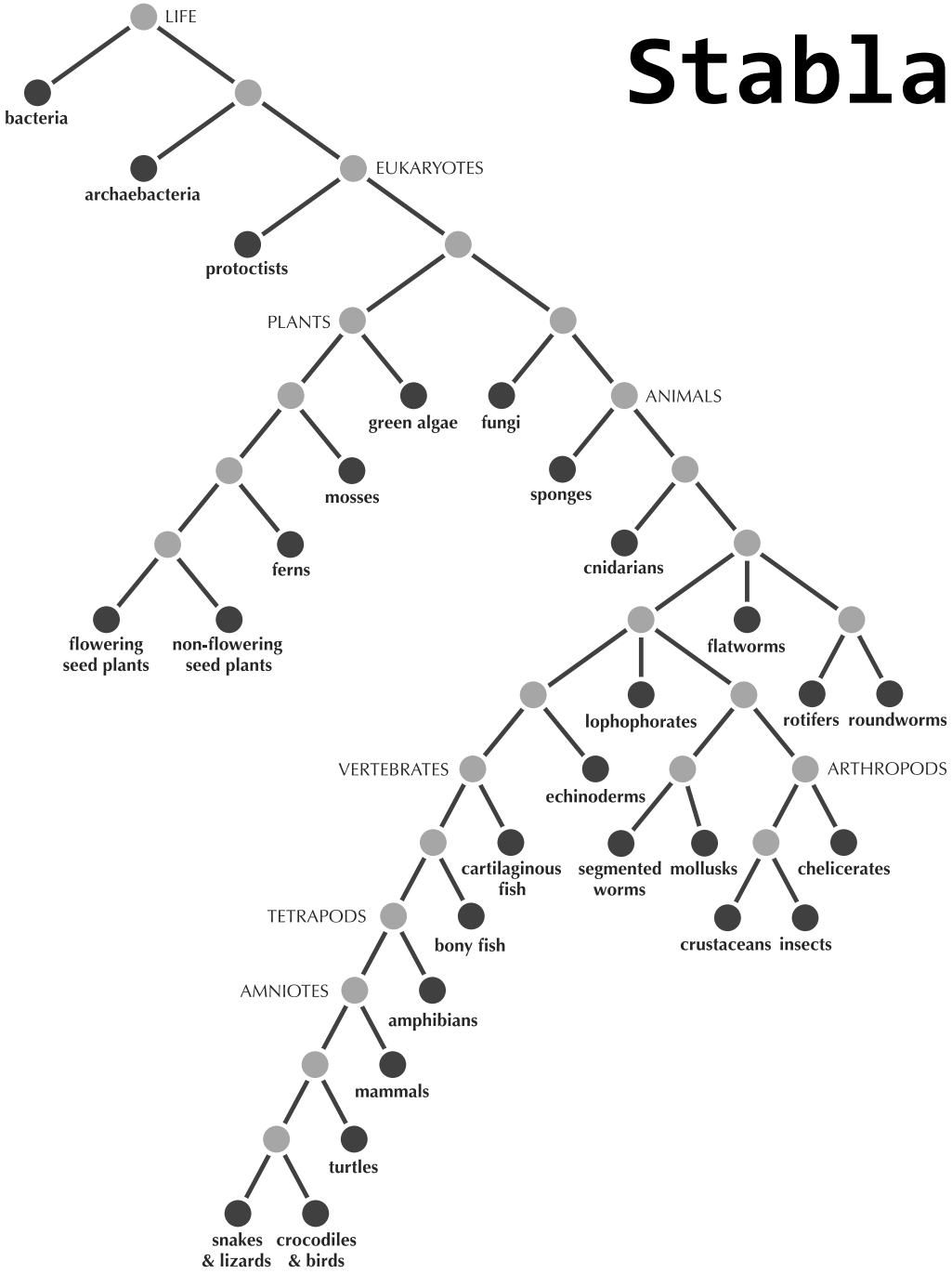
```
   1 mfifllfltl tsgsdldrct tfddvqapny tqhtssmrgv yypdeifrsd tlyltqdlfl
  61 pfysnvtgfh tinhtfgnpv ipfkdgiyfa ateksnvvrg wvfgstmnnk sqsviiinns
 121 tnvviracnf elcdnpffav skpmgtqtht mifdnafnct feyisdafsl dvseksgnfk
 181 hlrefvfknk dgflyvykgy qpidvvrdlp sgfntlkpif klplginitn frailtafsp
 241 aqdiwgtsaa ayfvgylkpt tfmlkydeng titdavdcsq nplaelkcsv ksfeidkgiy
 301 qtsnfrvvps gdvvrfpnit nlcpfgevfn atkfpsvyaw erkkisncva dysvlynstf
 361 fstfkcygvs atklndlcfs nvyadsfvvk gddvrqiapg qtgviadyny klpddfmgcv
 421 lawntrnida tstgnynyky rylrhgklrp ferdisnvpf spdgkpctpp alncywplnd
 481 ygfytttgig yqpyrvvvls fellnapatv cgpklstdli knqcvnfnfn gltgtgvltp
 541 sskrfqpfqq fgrdvsdftd svrdpktsei ldispcafgg vsvitpgtna ssevavlyqd
 601 vnctdvstai hadqltpawr iystgnnvfq tqagcligae hvdtsyecdi pigagicasy
 661 htvsllrsts qksivaytms lgadssiays nntiaiptnf sisittevmp vsmaktsvdc
 721 nmyicgdste canlllqygs fctqlnrals giaaeqdrnt revfaqvkqm yktptlkyfg
 781 gfnfsqilpd plkptkrsfi edllfnkvtl adagfmkqyg eclgdinard licaqkfngl
 841 tvlppllttdd miaaytaalv sgtatagwtf gagaalqipf amqmayrfng igvtqnvlye
 901 nqkqianqfn kaisqiqesl tttstalgkl qdvvnqnaqa lntlvkqlss nfgaissvln
 961 dilsrldkve aevqidrlit grlqslqtyv tqqliraaei rasanlaatk msecvlgqsk
1021 rvdfcgkgyh lmsfpqaaph gvvflhvtyv psqernftta paichegkay fpregvfvfn
1081 gtswfitqrn ffspqiittd ntfvsgncdv vigiinntvy dplqpeldsf keeldkyfkn
1141 htspdvdlgd isginasvvn iqkeidrlne vaknlnesli dlqelgkyeq yikwpwyvwl
1201 gfiagliaiv mvtillccmt sccsclkgac scgscckfde ddsepvlkgv klhyt
```

Stabla



Stablo: povezani aciklički graf.

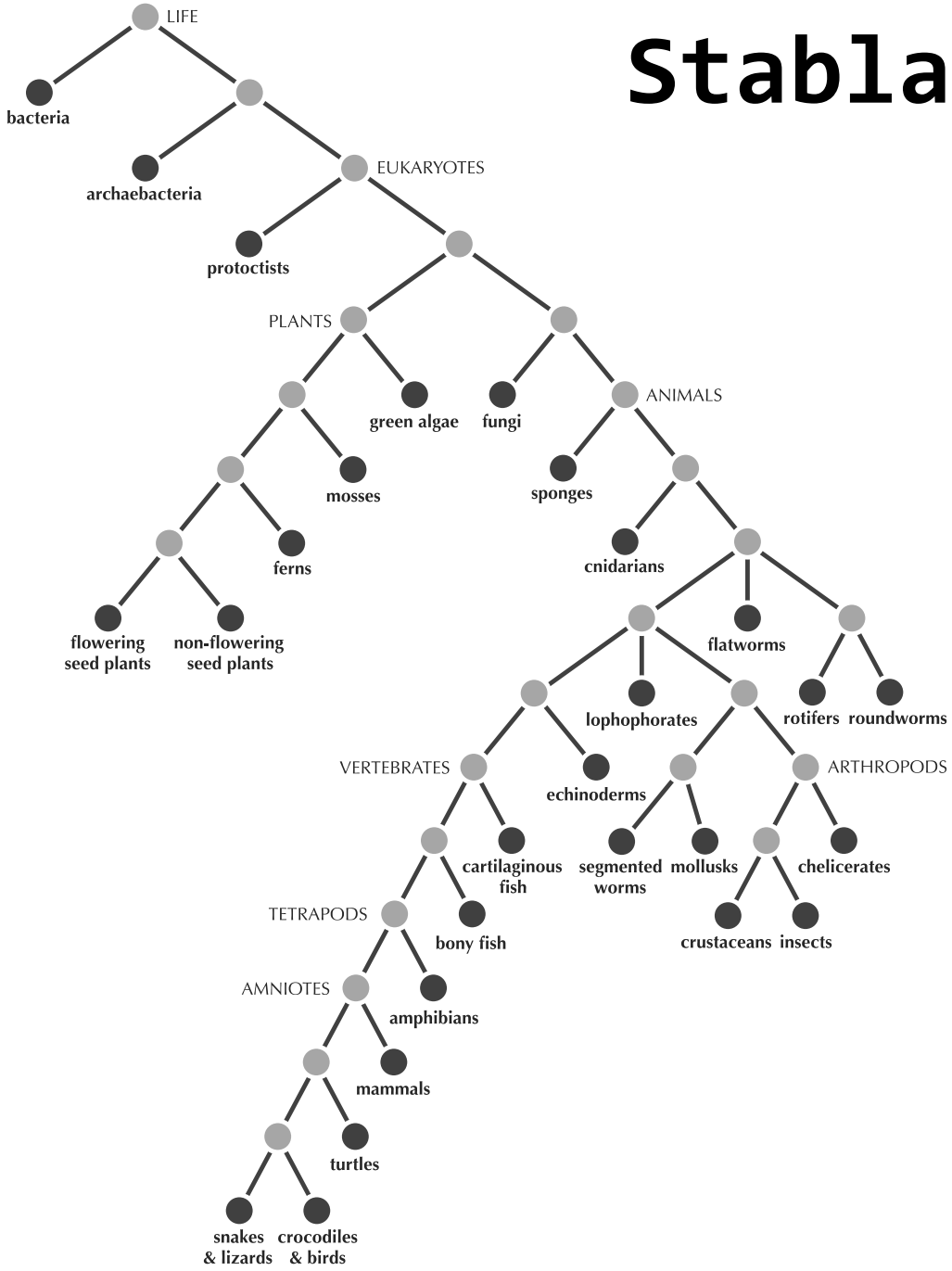
Stabla



Stablo: povezani aciklički graf.

Listovi (*degree*=1): današnje vrste

Stabla

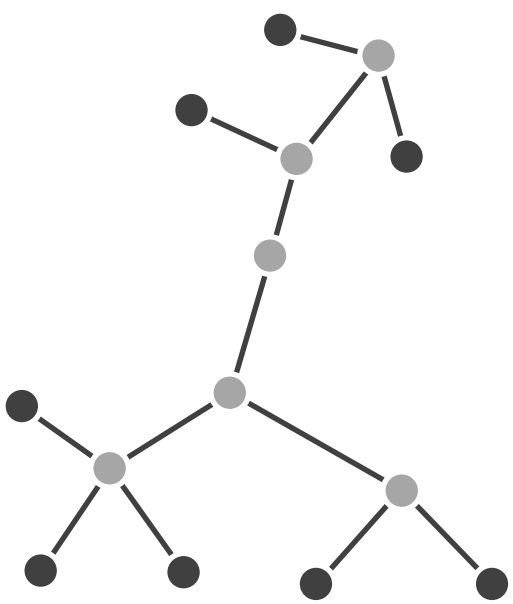


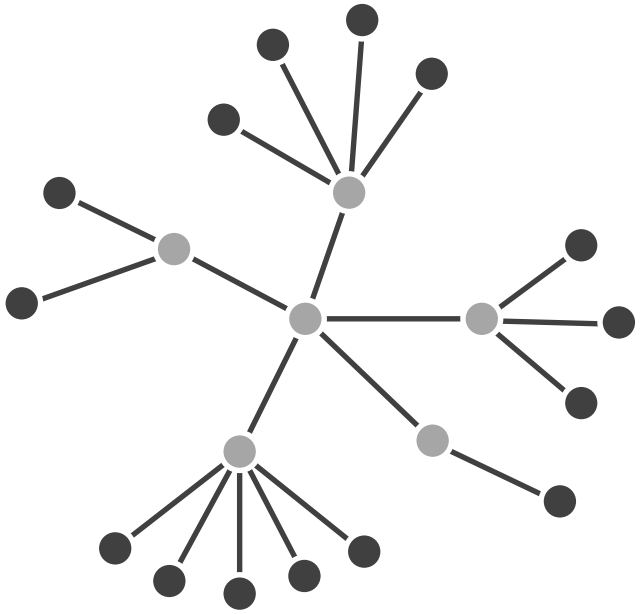
Stablo: povezani aciklički graf.

Listovi (*degree*=1): današnje vrste

Unutrašnji čvorovi (*degree* ≥ 1): izumrle vrste

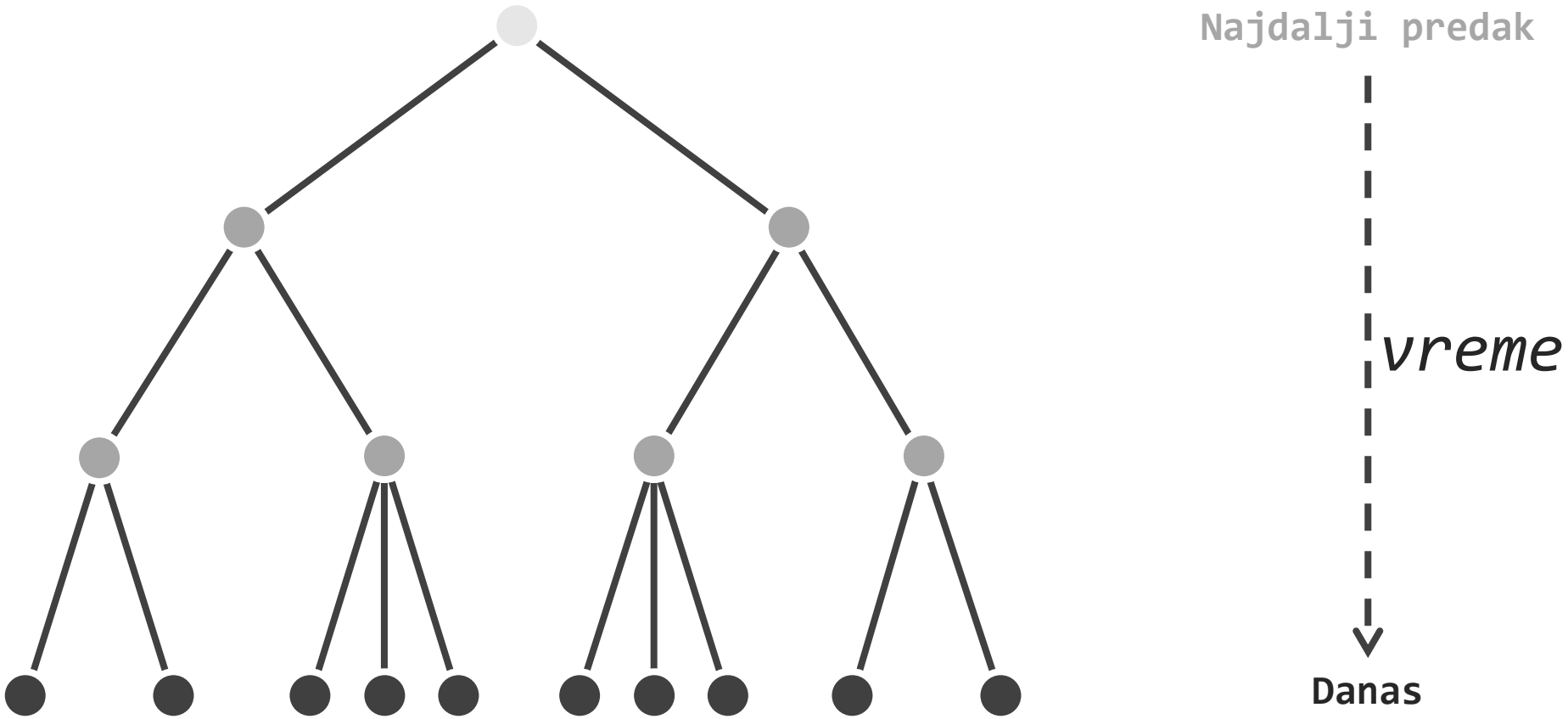
Stabla





Za povezana aciklična stabla se može pokazati:
1. Svako stablo sa bar dva čvora sadrži bar dva lista.
2. Svako stablo sa n čvorova sadrži tačno $n - 1$ grana.

Stabla



Koreni čvor: najdalji zajednički predak

Filogeneza na osnovu rastojanja

Problem filogeneze na osnovu rastojanja:
Konstruisati evolutivno stablo na osnovu matrice rastojanja.

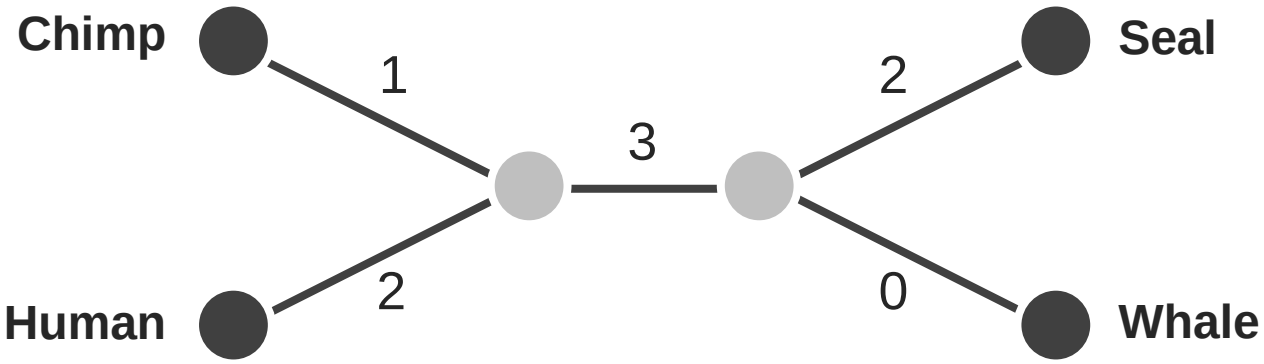
- **Ulaz:** Matrica rastojanja.
- **Izlaz:** Nekoreno stablo koje *odgovara* matrici rastojanja.

Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

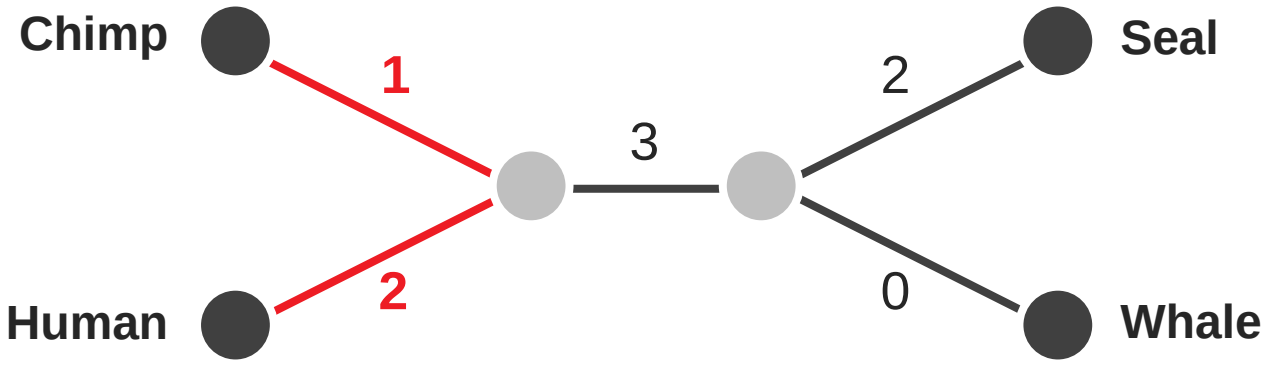
Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



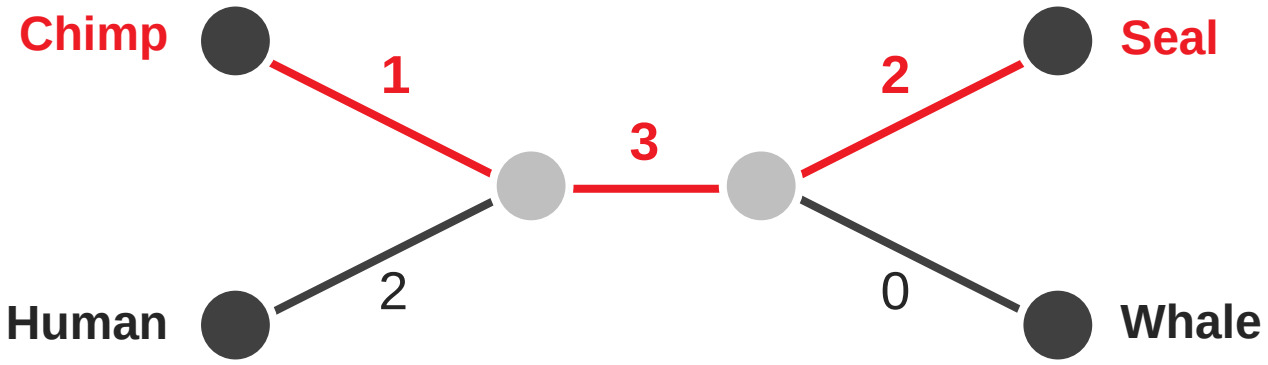
Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	**3**	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



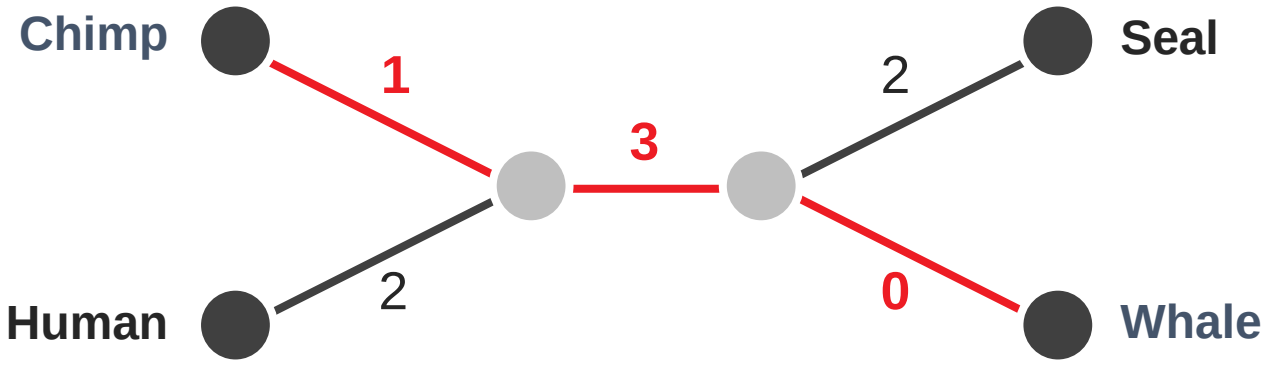
Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	**6**	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



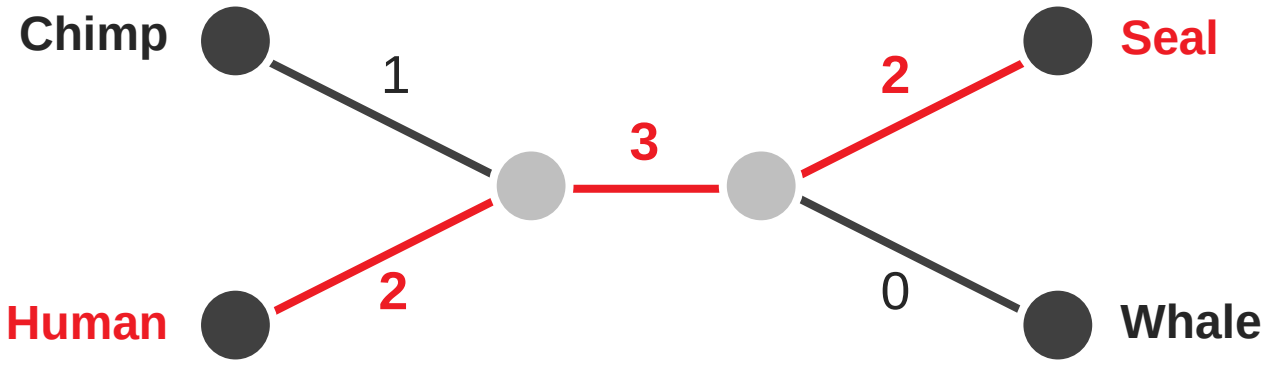
Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	6	**4**
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



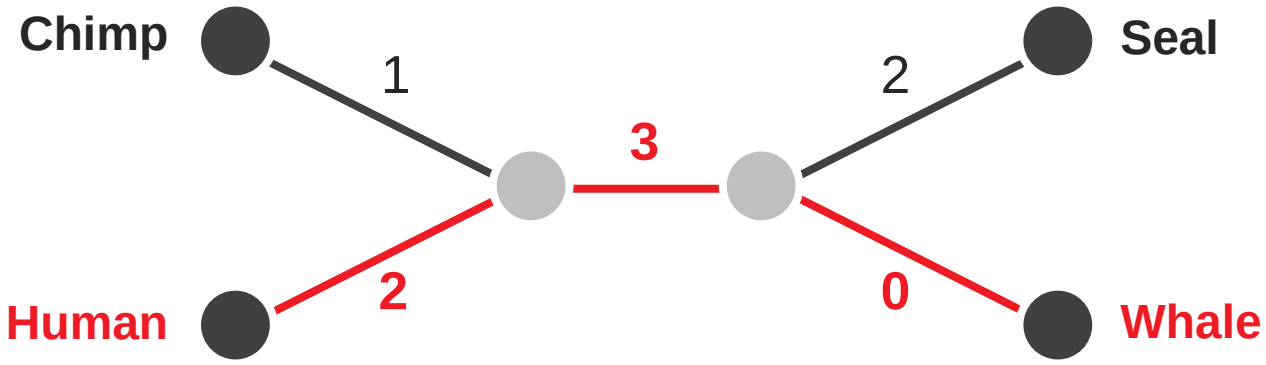
Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	**7**	5
Seal	6	7	0	2
Whale	4	5	2	0



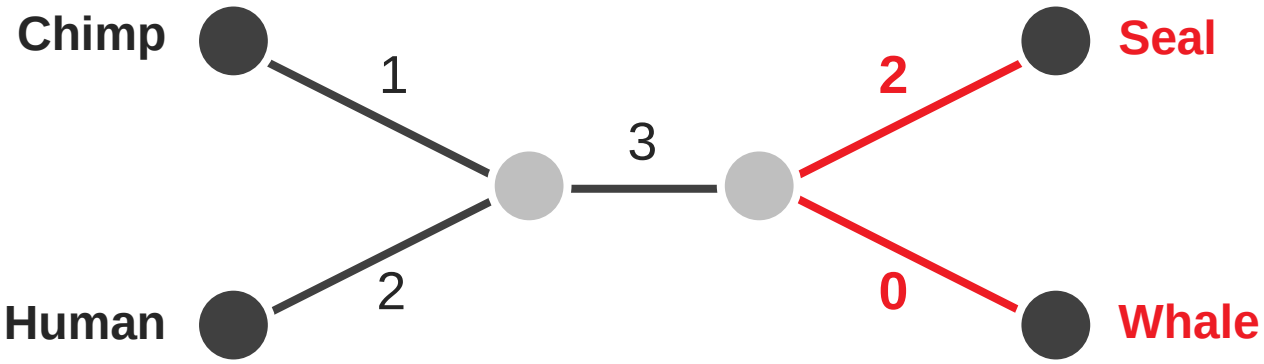
Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	**5**
Seal	6	7	0	2
Whale	4	5	2	0



Kada stablo odgovara matrici?

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	**2**
Whale	4	5	2	0



Filogeneza na osnovu rastojanja

Pokušajmo da konstruišemo stablo koje odgovara sledećoj matrici:

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

Nijedno odgovarajuće stablo

Pokušajmo da konstruišemo stablo koje odgovara sledećoj matrici:

	i	***j***	***k***	***l***
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

Da bi za datu matricu bilo moguće konstruisati odgovarajuće evolutivno stablo, matrica mora biti ***aditivna*** (u nastavku)

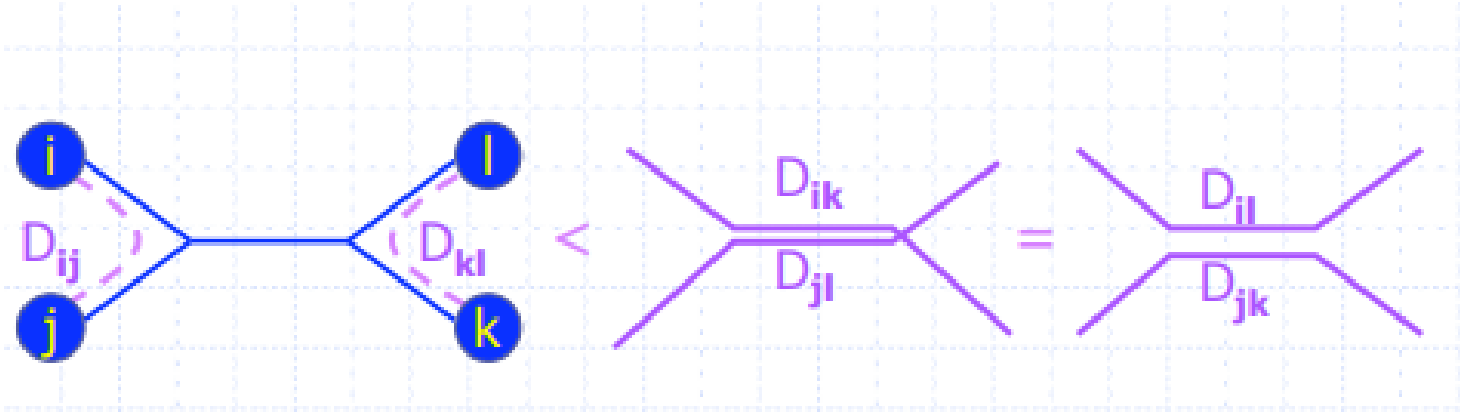
Aditivne matrice

- Definicija: Za dato evolutivno stablo, matricu koja opisuje rastojanja između njegovih listova zovemo *aditivnom matricom*

- Teorema: Matrica *D* je aditivna akko za proizvoljna 4 indeksa u matrici *i,j* i *k,l* važi:

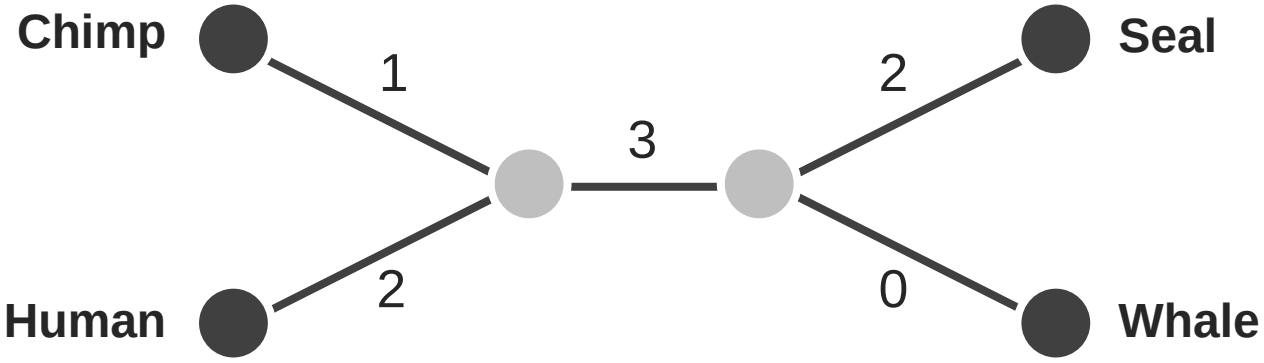
$$Dij + Dkl <= Dik+Djl = Dil + Djk$$

- Ova teorema sugeriše kako da na osnovu matrice povežemo listove u evolutivnom stablu:



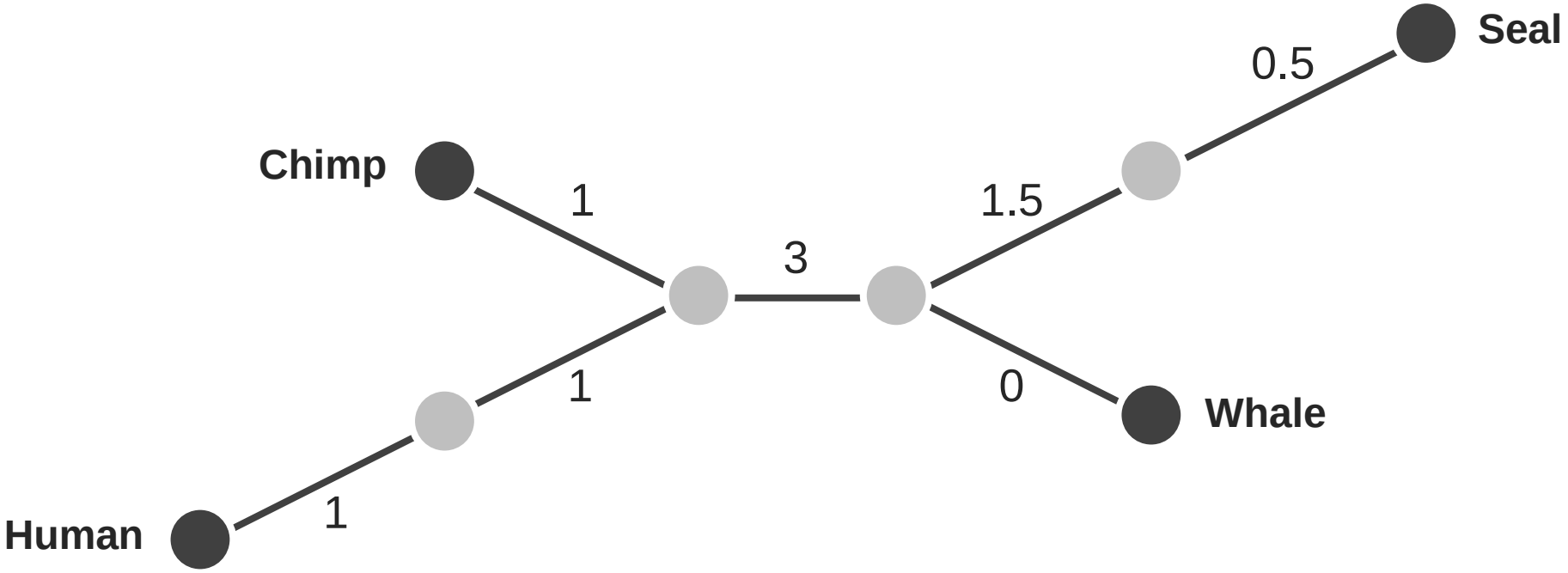
Više od jednog odgovarajućeg stabla

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

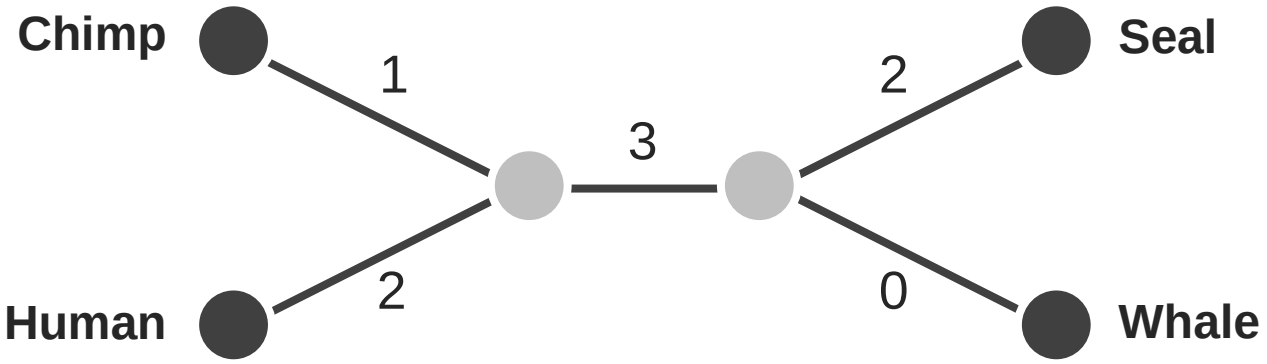


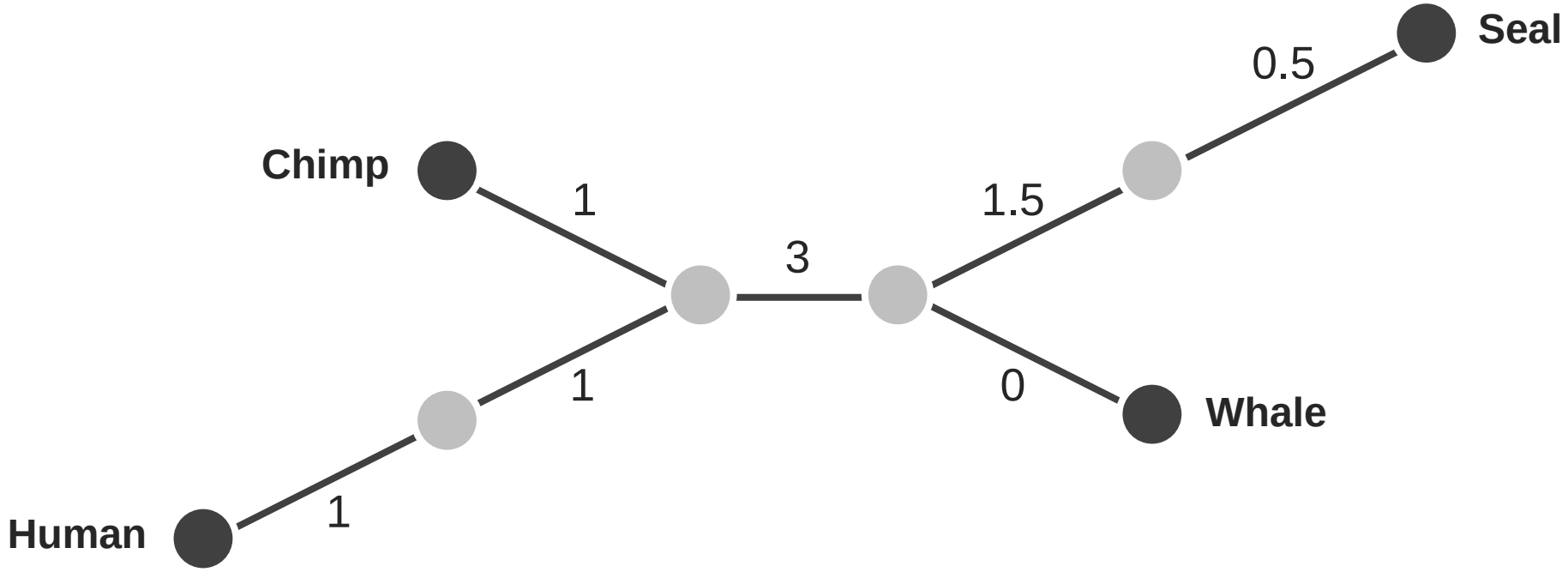
Više od jednog odgovarajućeg stabla

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

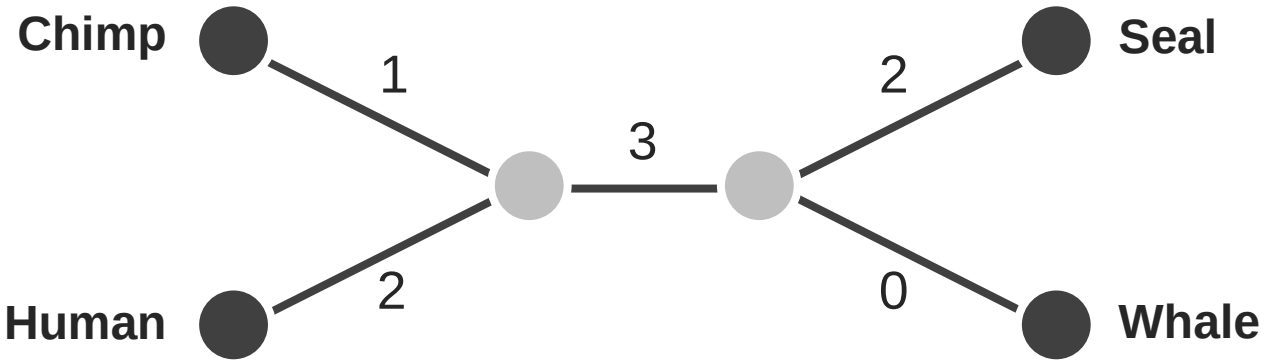


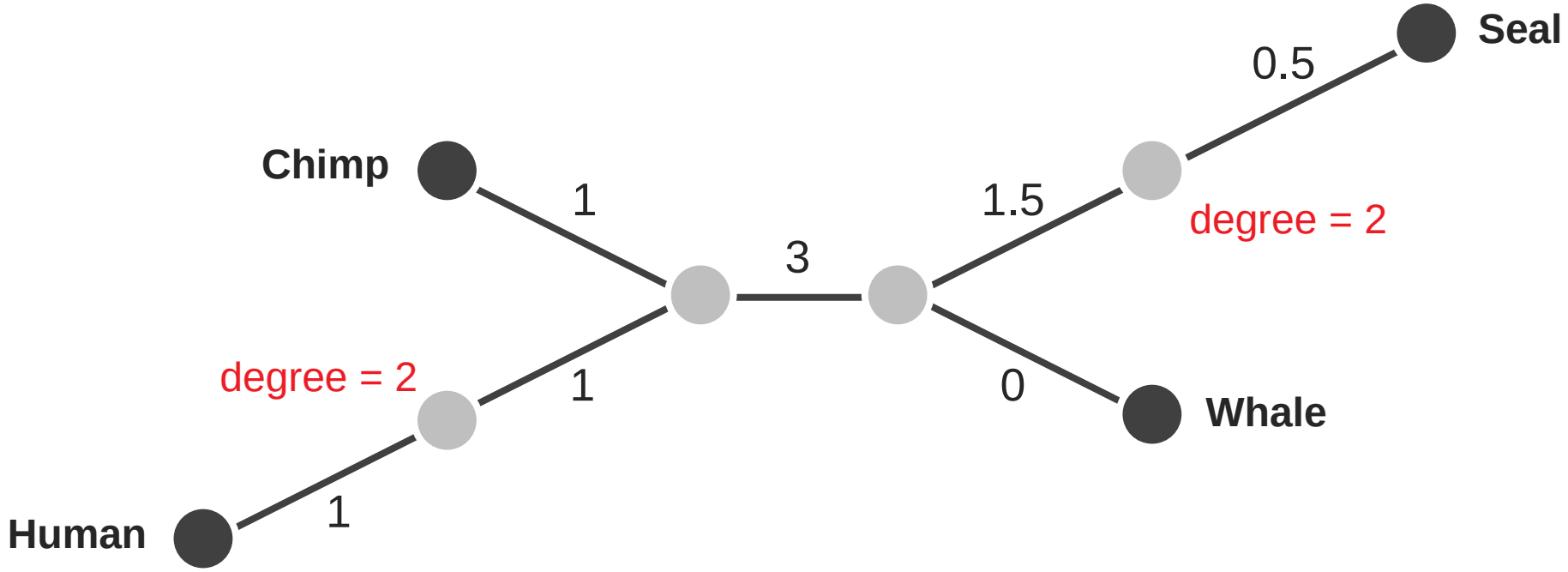
Koje stablo je bolje?



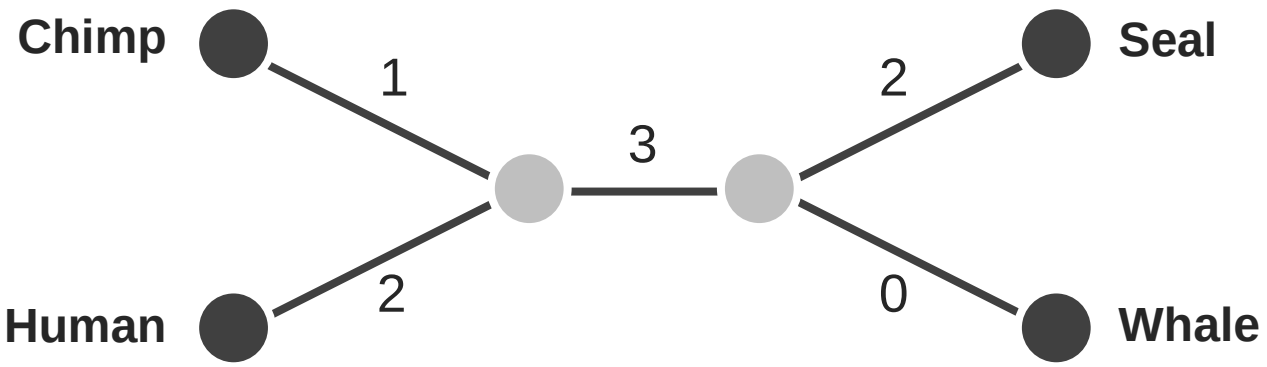


Koje stablo je bolje?



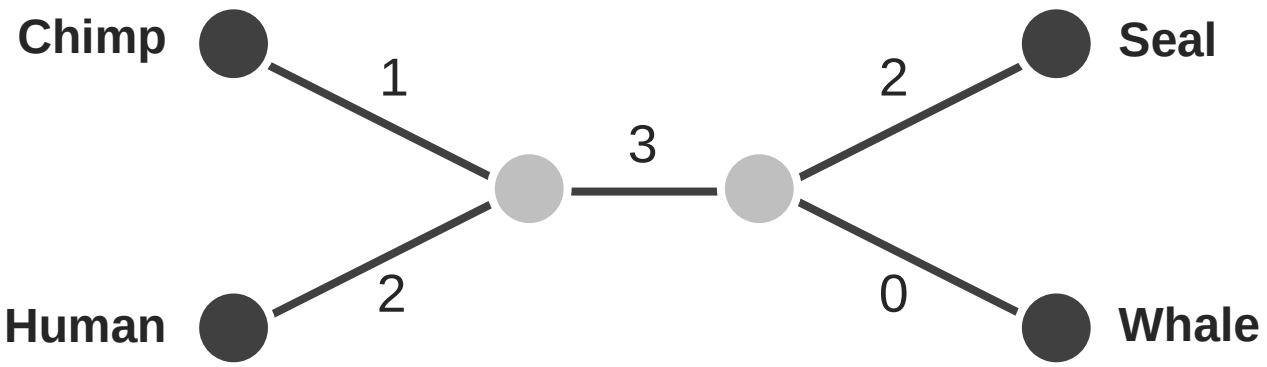


Koje stablo je bolje?



Prosto stablo: stablo bez čvorova stepena >=2.

Koje stablo je bolje?



Prosto stablo: stablo bez čvorova stepena =2.

Teorema: Postoji tačno jedno prosto stablo koje odgovara aditivnoj matrici

Reformulacija

Problem filogeneze na osnovu rastojanja:
Konstruisati evolutivno stablo na osnovu aditivne matrice rastojanja.

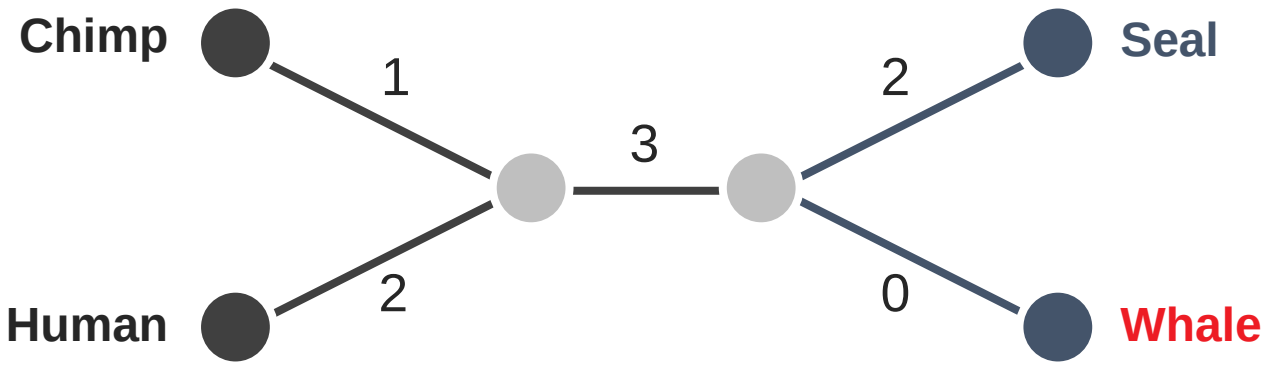
- **Ulaz:** Aditivna matrica rastojanja.
- **Izlaz:** Prosto stablo koje *odgovara* datoj matrici rastojanja.

Pregled

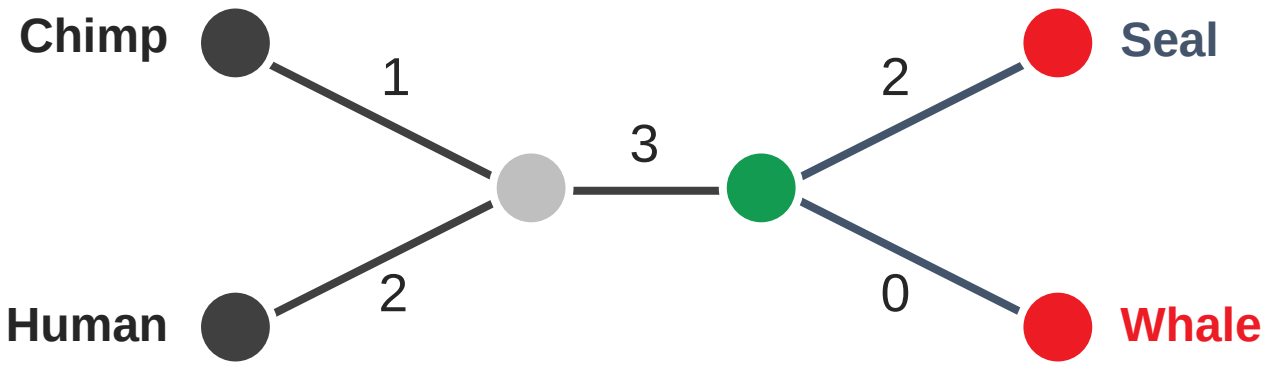
- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- **Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja**
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- Neighbour-Joining algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

- Primetimo da minimalna pozitivna vrednost matrice rastojanja odgovara listovima u stablu koje povezuje zajednički roditelj
- Takve listove zovemo **susednim** listovima

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	**2**
Whale	4	5	2	0

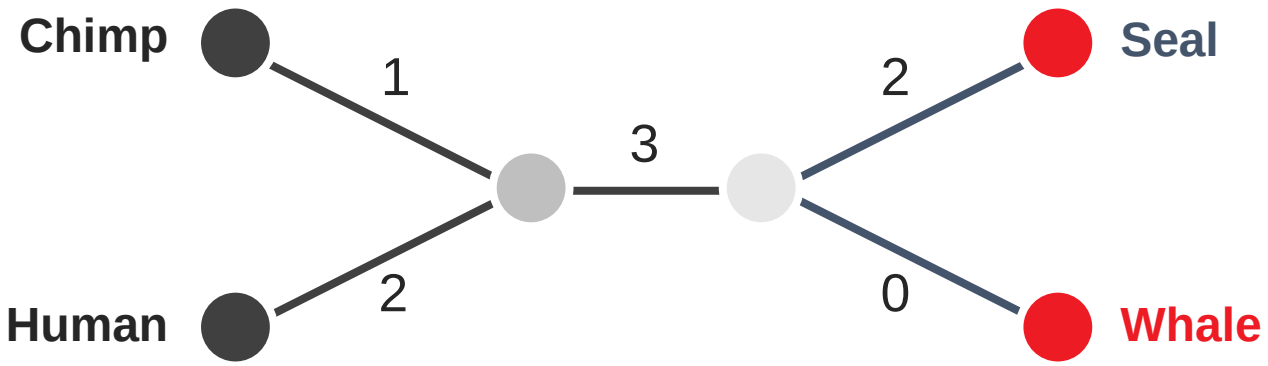


Foka i kit su **susedi** (dele isti roditeljski čvor).

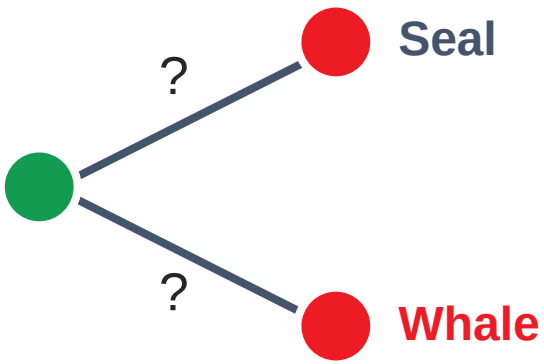


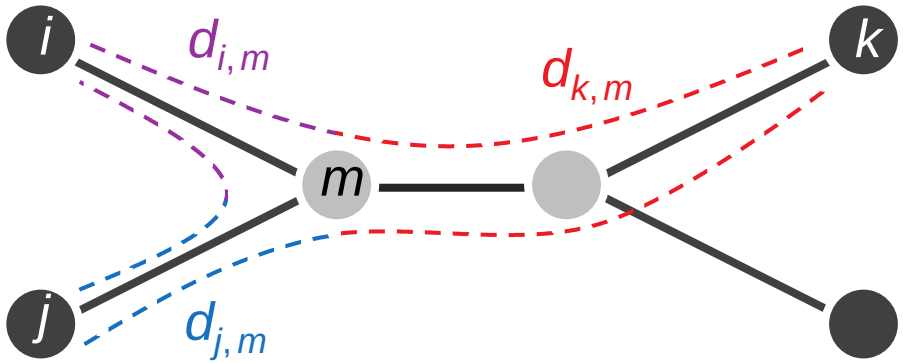
Foka i kit su **susedi** (dele isti roditeljski čvor).

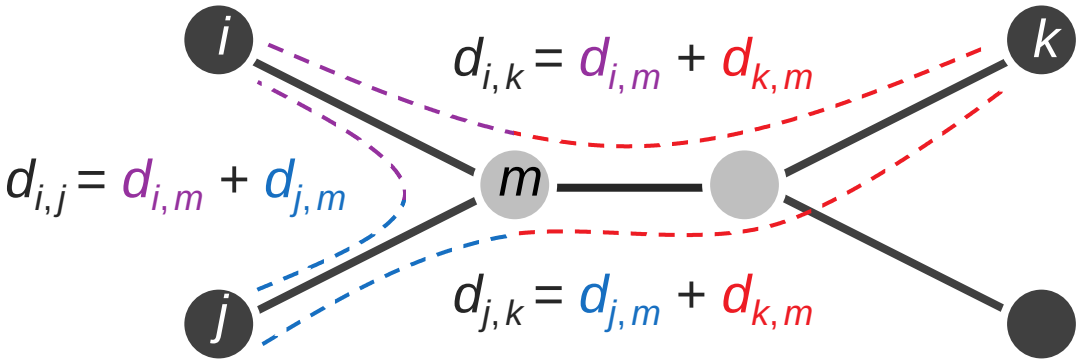
Teorema: Za svako prosto stablo sa bar četiri čvora postoji bar jedan par susednih listova.

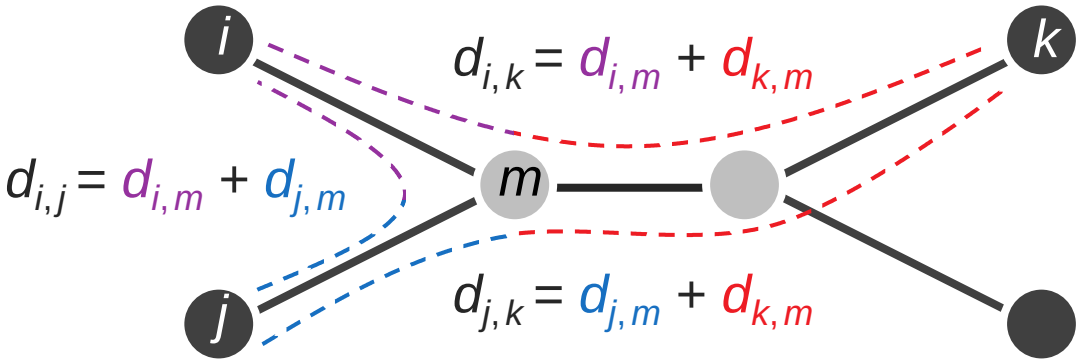


	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	**2**
Whale	4	5	2	0

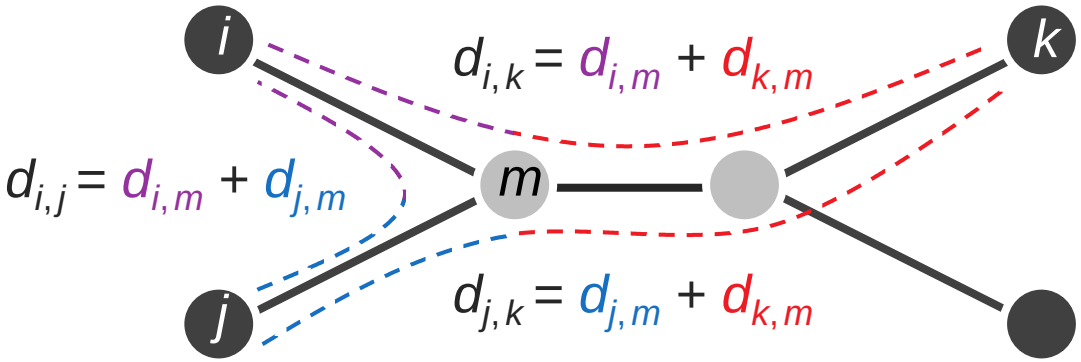



i
j
m
k
$d_{i,m}$
$d_{j,m}$
$d_{k,m}$


i
k
m
j
$d_{i,k} = d_{i,m} + d_{k,m}$
$d_{i,j} = d_{i,m} + d_{j,m}$
$d_{j,k} = d_{j,m} + d_{k,m}$



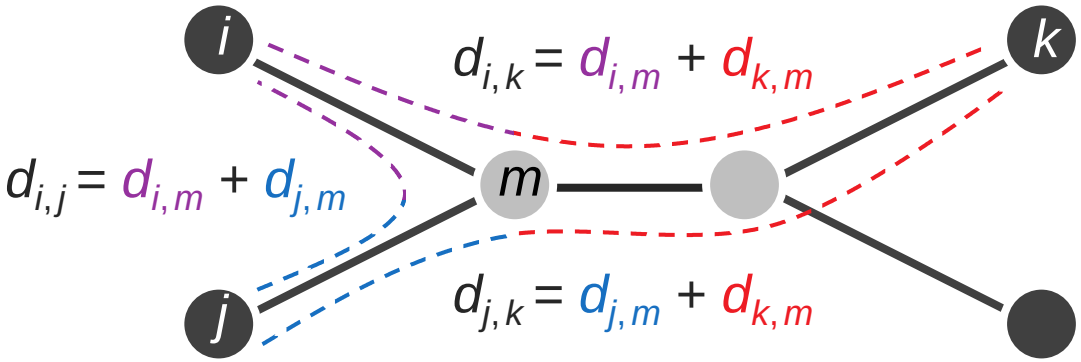
$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$



$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$\boldsymbol{d_{i,k}}$, $\boldsymbol{d_{j,k}}$,$\boldsymbol{d_{i,j}}$ su rastojanja između *listova* koja su data u matrici rastojanja, dok npr rastojanja $\boldsymbol{d_{i,m}}$ i $\boldsymbol{d_{j,m}}$, koja predstavljaju rastojanja između lista i unutrašnjeg čvora, nisu data

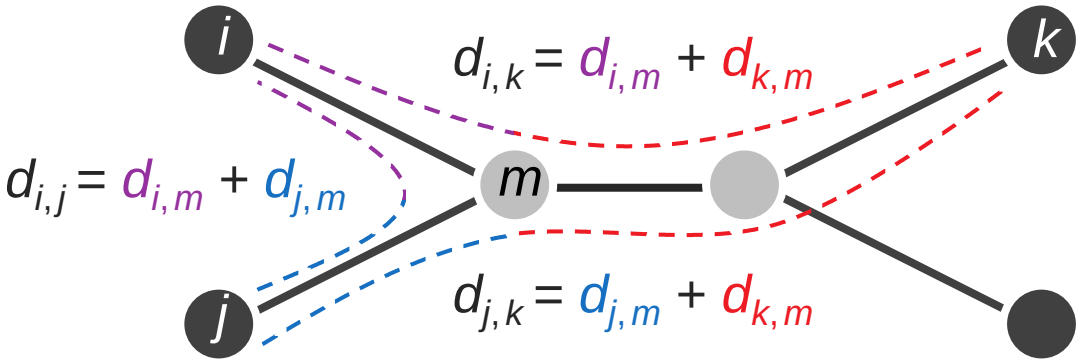


$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

sa velikim ***D*** (***D*****$_{i,k}$** , ***D*****$_{j,k}$** , ***D*****$_{i,j}$**) označavamo elemente matrice rastojanja (rastojanje između dva lista u evolutivnom stablu), dok sa malim ***d*** označavamo rastojanje između bilo koja dva čvora u evolutivnom stablu

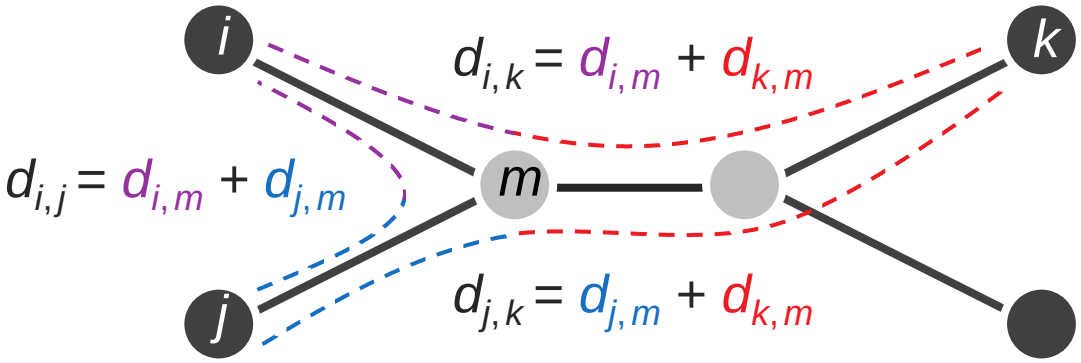


$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

$$\therefore \quad d_{i,m} = D_{i,k} - (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$



$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

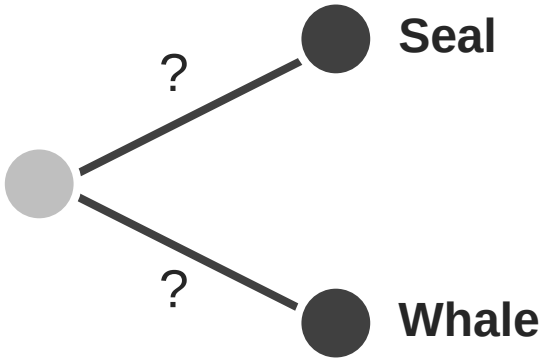
$$\therefore \quad d_{i,m} = D_{i,k} - (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2$$

Analogno za drugog suseda $\boldsymbol{d_{j,m}}$

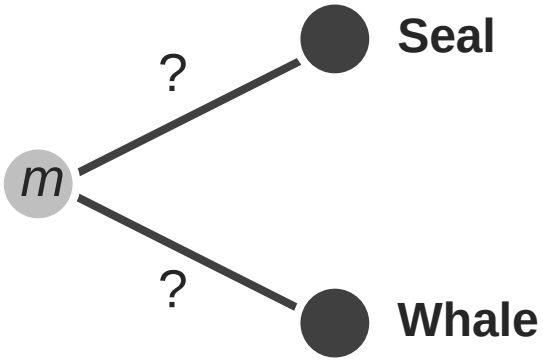
Obratimo pažnju da je čvor *k* proizvoljan – bilo koji list koji je različit od listova čija rastojanja do roditeljskog čvora tražimo

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



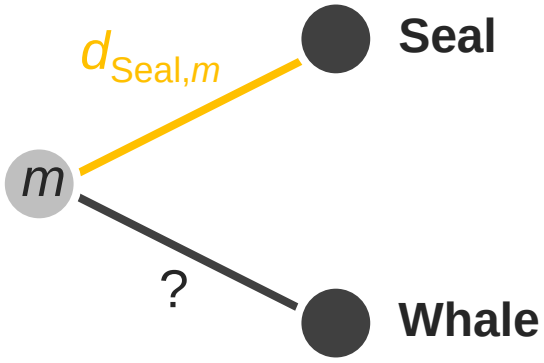
$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



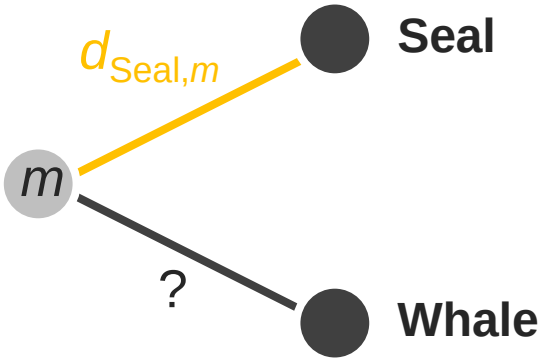
$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



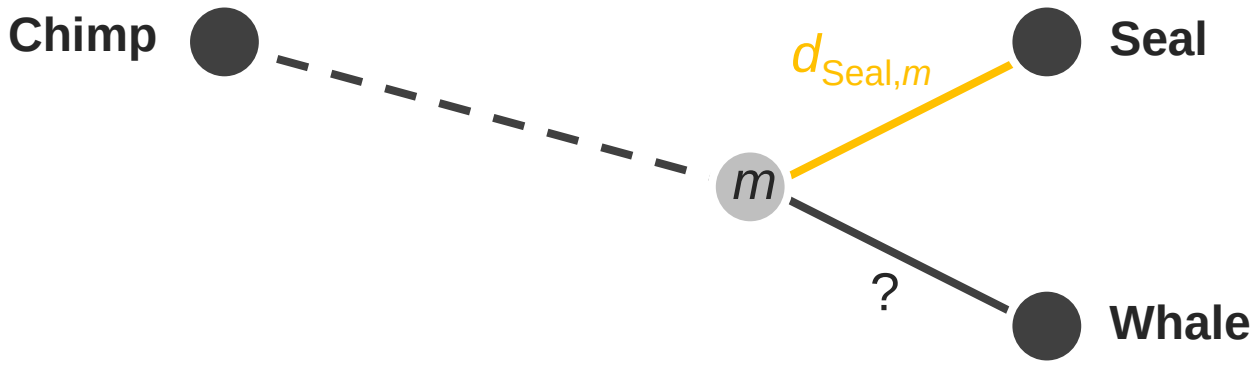
$$d_{Seal,m} = (D_{Seal,k} + D_{Seal,j} - D_{j,k}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



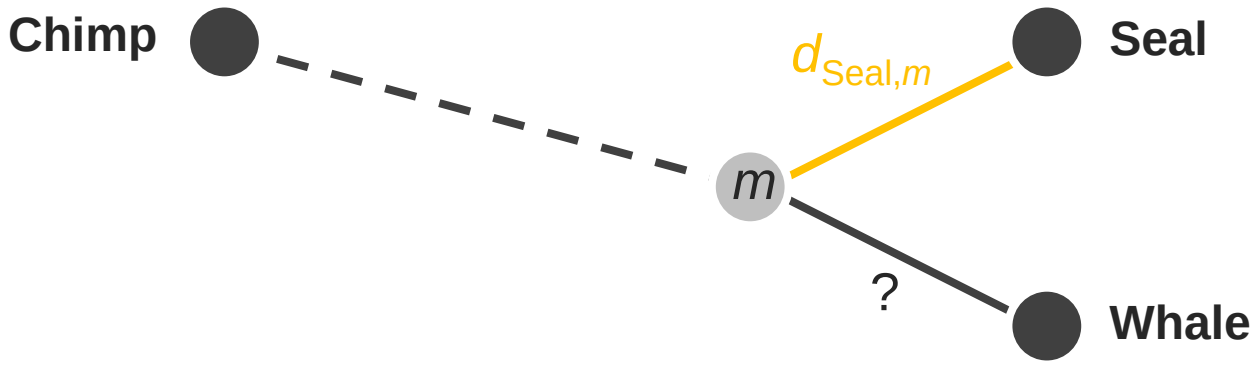
$$d_{Seal,m} = (D_{Seal,k} + D_{Seal,Whale} - D_{Whale,k}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



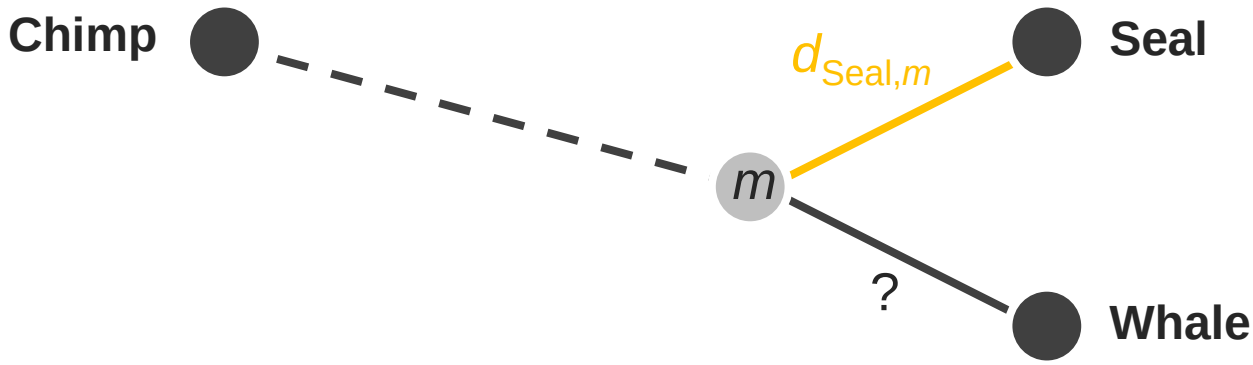
$$d_{Seal,m} = (D_{Seal,Chimp} + D_{Seal,Whale} - D_{Whale,Chimp}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



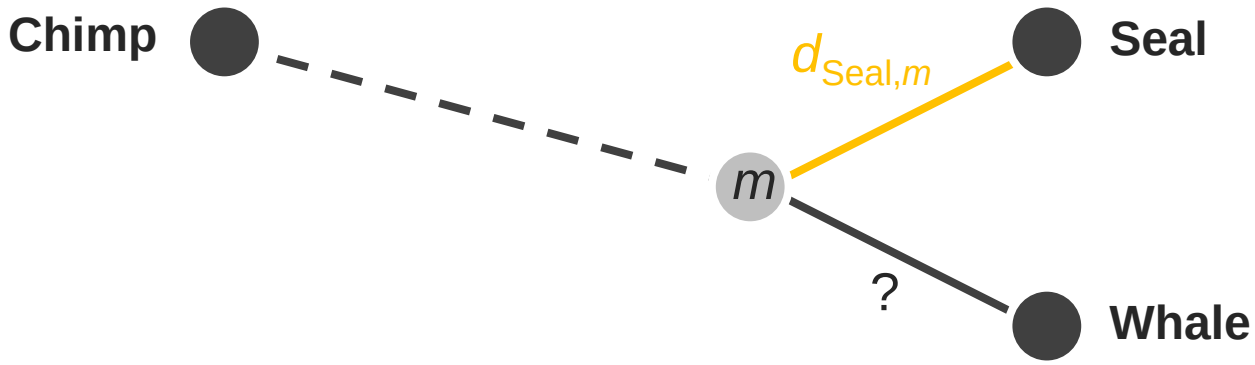
$$d_{Seal,m} = (\quad 6 \quad + D_{Seal,Whale} - D_{Whale,Chimp}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



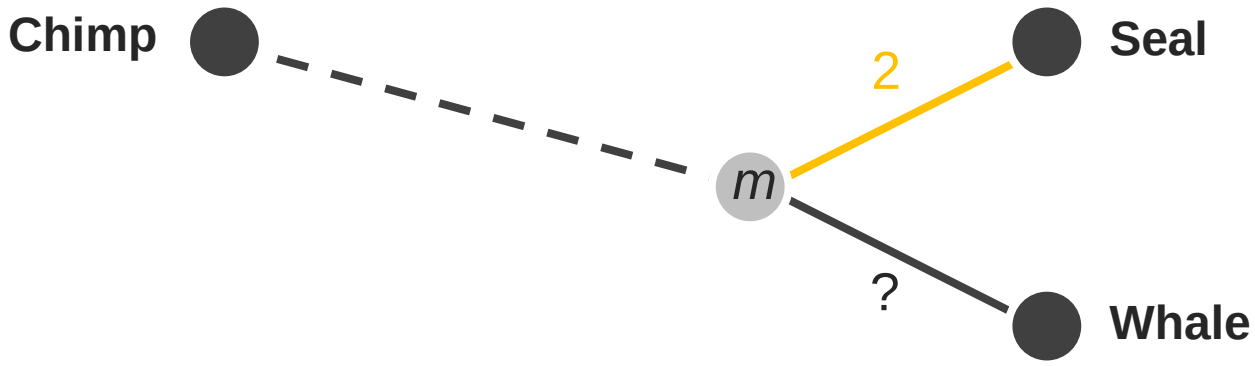
$$d_{Seal,m} = (\quad 6 \quad + \quad 2 \quad - D_{Whale,Chimp}) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



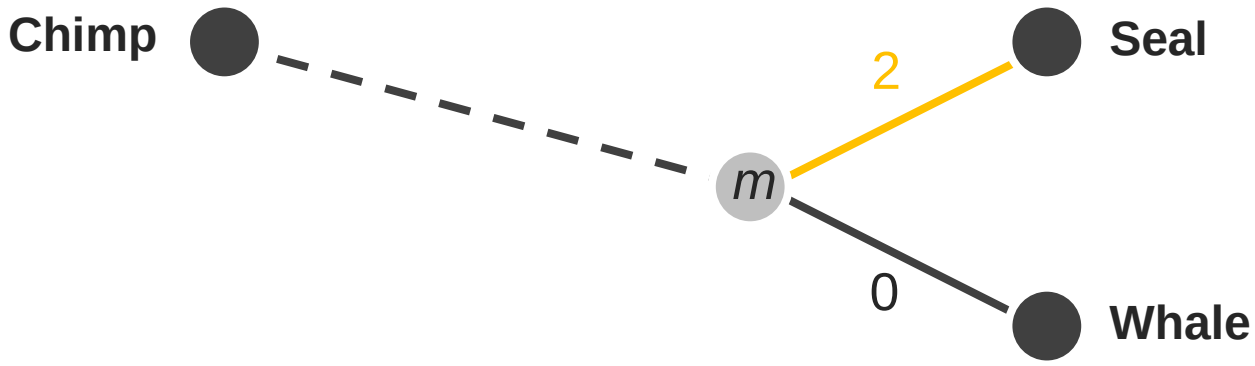
$$d_{Seal,m} = (\quad 6 \quad + \quad 2 \quad - \quad 4 \quad) / 2$$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



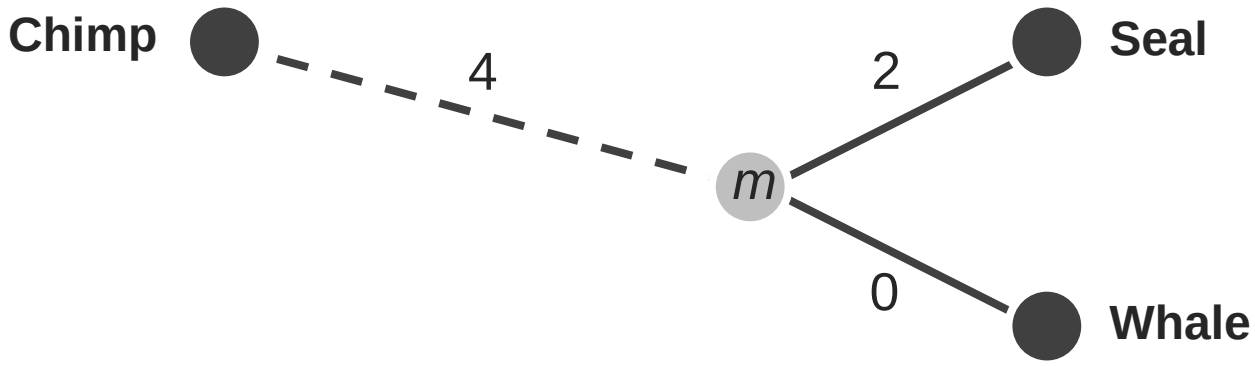
$d_{Seal,m} = 2$

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

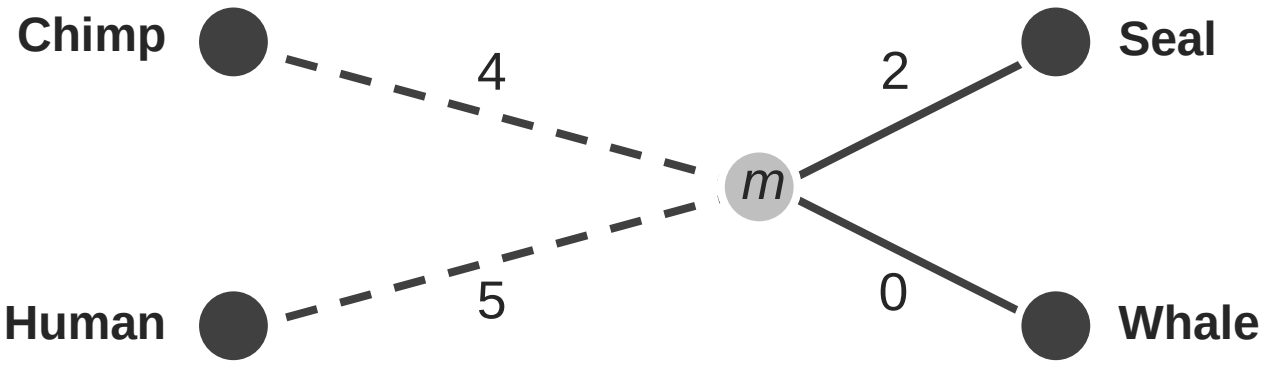


$d_{Seal,m} = 2$

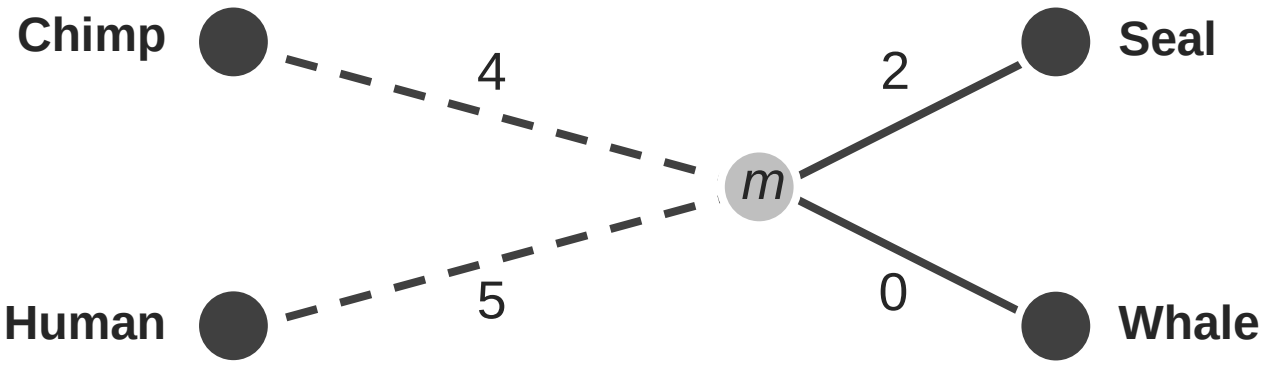
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



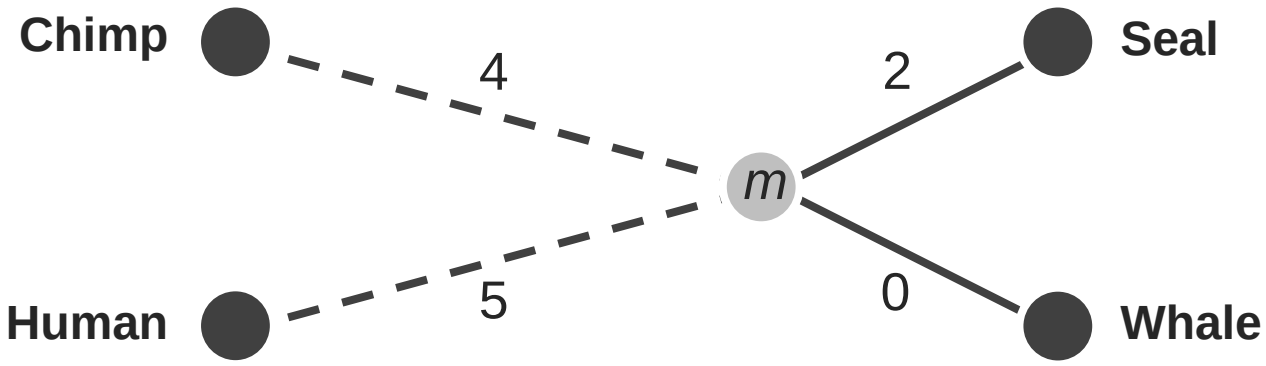
	Chimp	Human	Seal	Whale	*m*
Chimp	0	3	6	4	4
Human	3	0	7	5	5
Seal	6	7	0	2	2
Whale	4	5	2	0	0
m	4	5	2	0	0



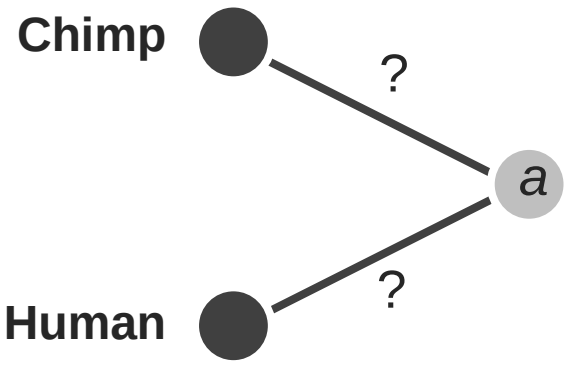
	Chimp	Human	Seal	Whale	*m*
Chimp	0	3	6	4	4
Human	3	0	7	5	5
Seal	6	7	0	2	2
Whale	4	5	2	0	0
m	4	5	2	0	0

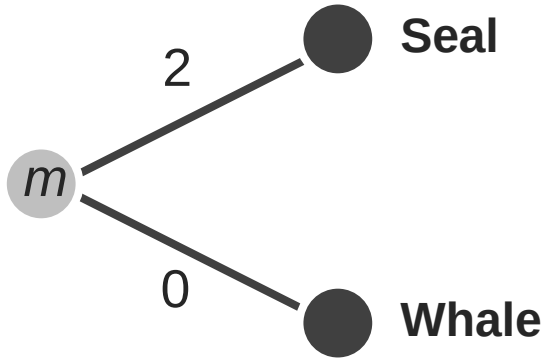


	Chimp	Human	*m*
Chimp	0	3	4
Human	3	0	5
m	4	5	0

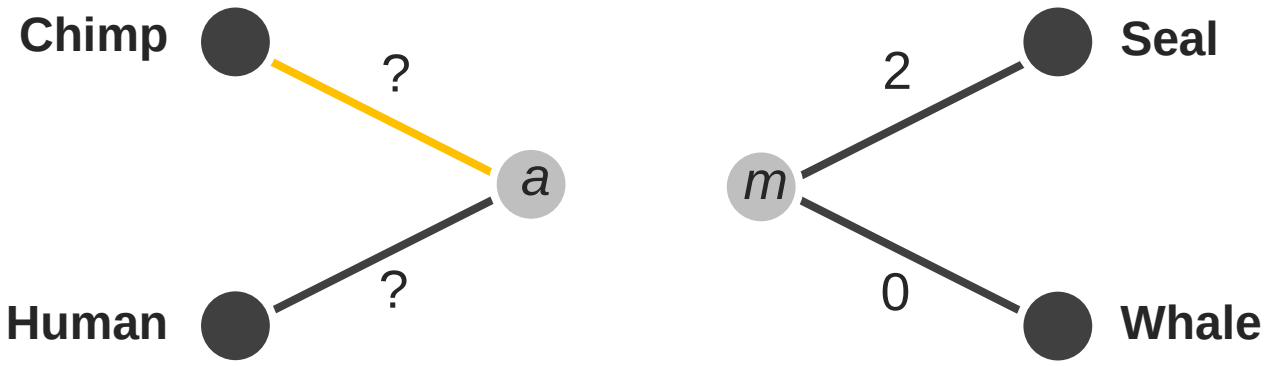


	Chimp	Human	*m*
Chimp	0	**3**	4
Human	**3**	0	5
m	4	5	0



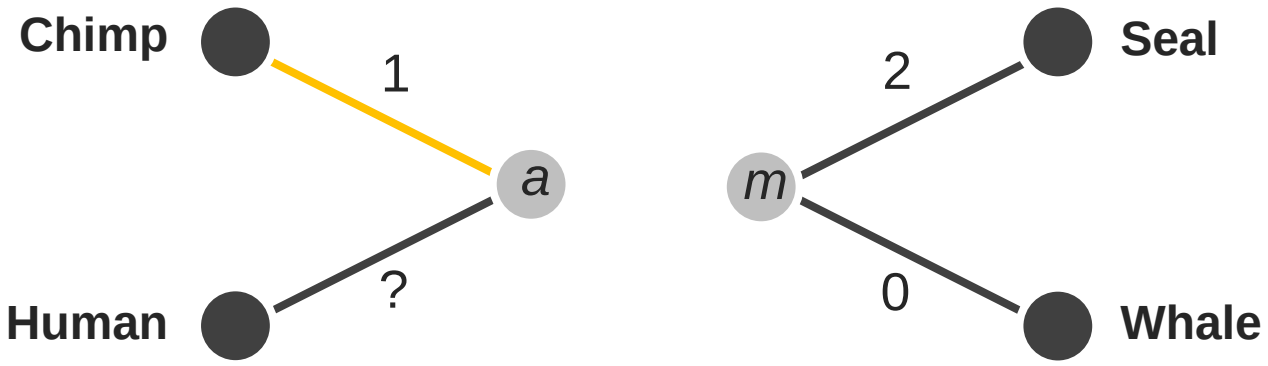


	Chimp	Human	*m*
Chimp	0	3	4
Human	3	0	5
m	4	5	0



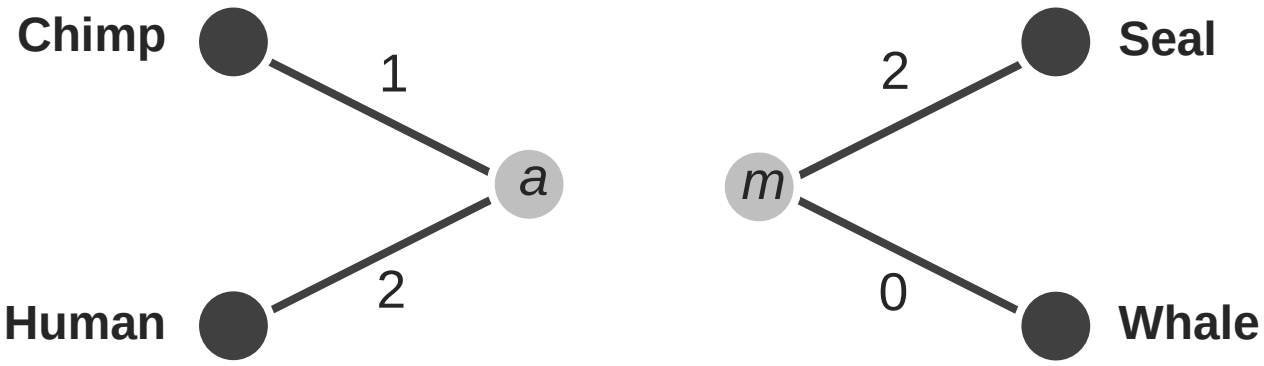
$$d_{\text{Chimp},a} = (D_{\text{Chimp},m} + D_{\text{Chimp,Human}} - D_{\text{Human},m}) / 2$$

	Chimp	Human	*m*
Chimp	0	3	4
Human	3	0	5
m	4	5	0

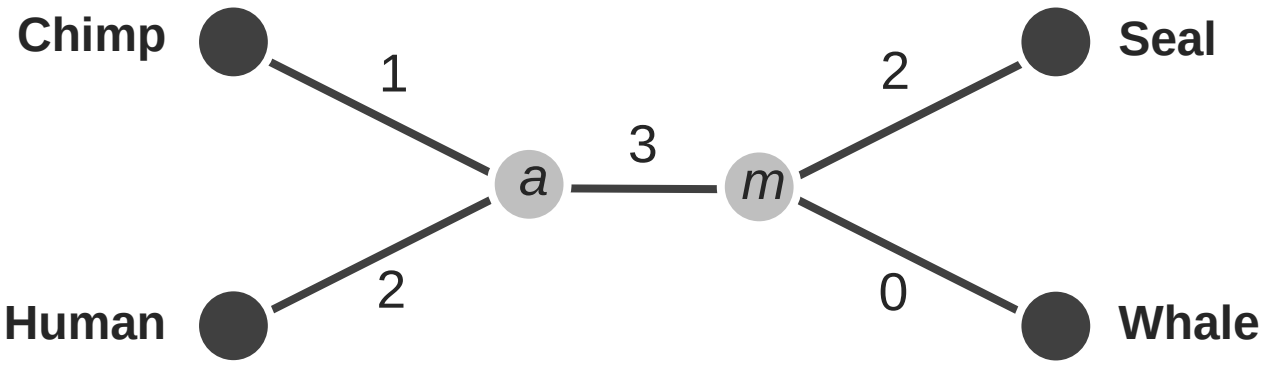


$d_{Chimp,a} = 1$

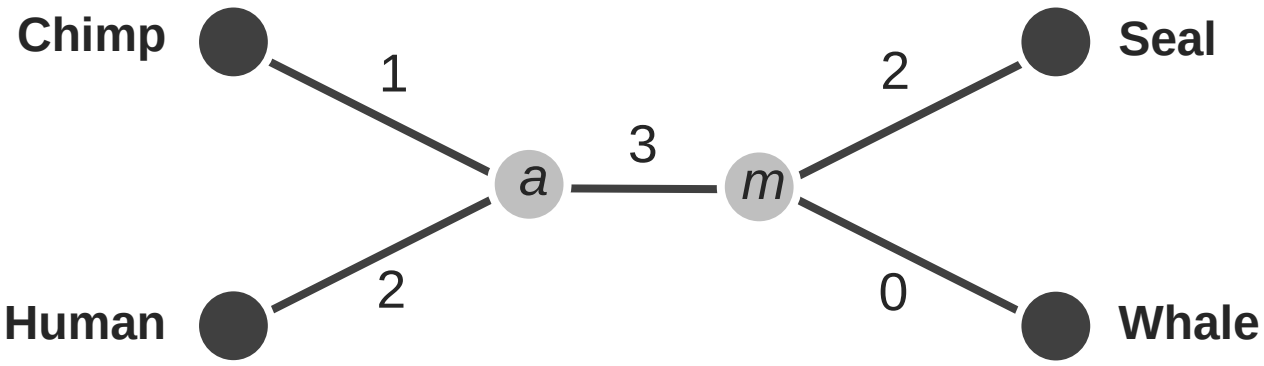
	Chimp	Human	*m*
Chimp	0	3	4
Human	3	0	5
m	4	5	0



	Chimp	Human	*m*
Chimp	0	3	4
Human	3	0	5
m	4	5	0



	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



Primenimo ovaj pristup na sledeću matricu:

	i	***j***	***k***	***l***
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

Pregled

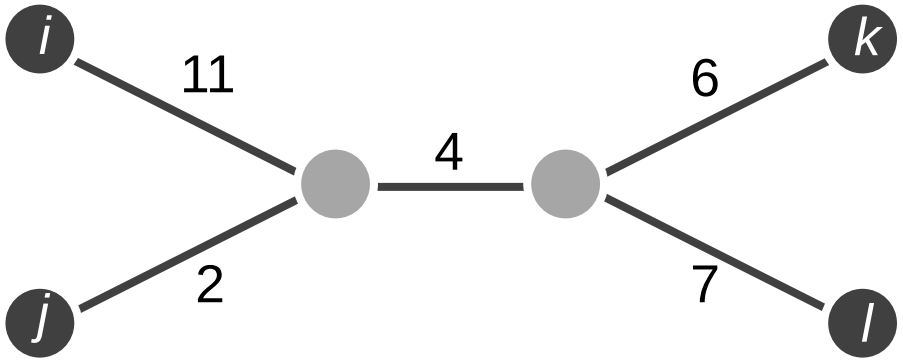
- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- ***AdditivePhylogeny* algoritam**
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- Neighbour-Joining algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

Zašto naš algoritam ne radi za ovu matricu?

	i	***j***	***k***	***l***
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

Zašto naš algoritam ne radi za ovu matricu?

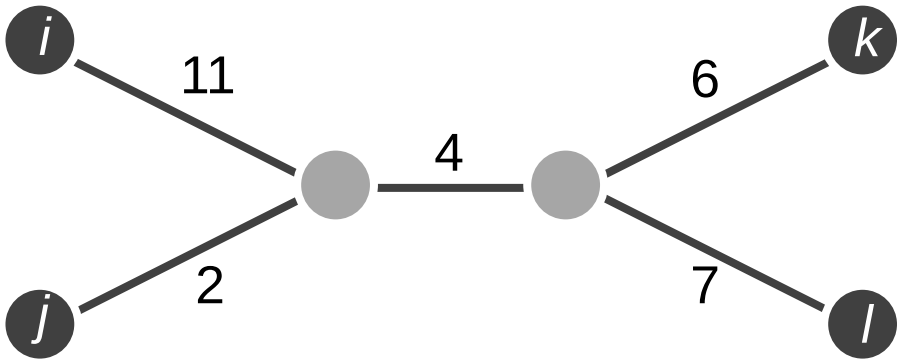
	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0



Zašto naš algoritam ne radi za ovu matricu?

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	**12**	13
k	21	12	0	13
l	22	13	13	0

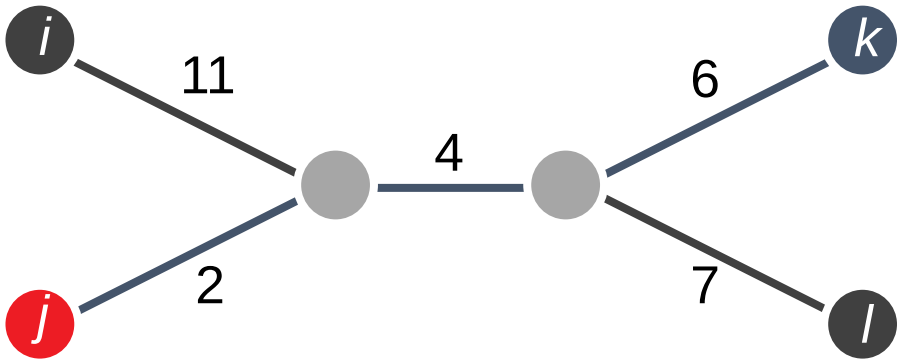
minimalni element je $D_{j,k}$



Zašto naš algoritam ne radi za ovu matricu?

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	**12**	13
k	21	12	0	13
l	22	13	13	0

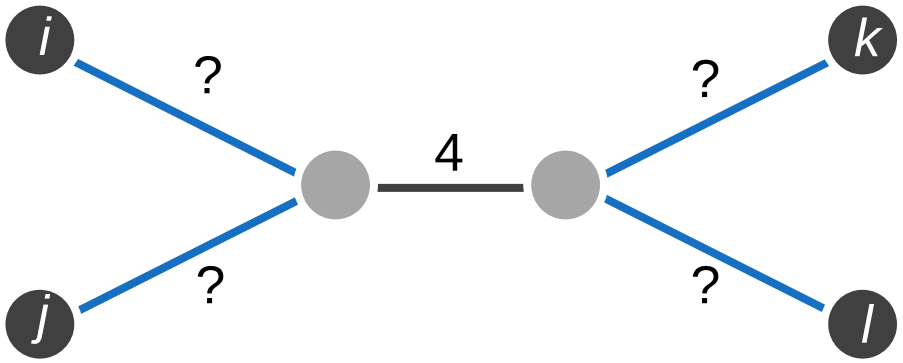
minimalni element je $D_{j,k}$



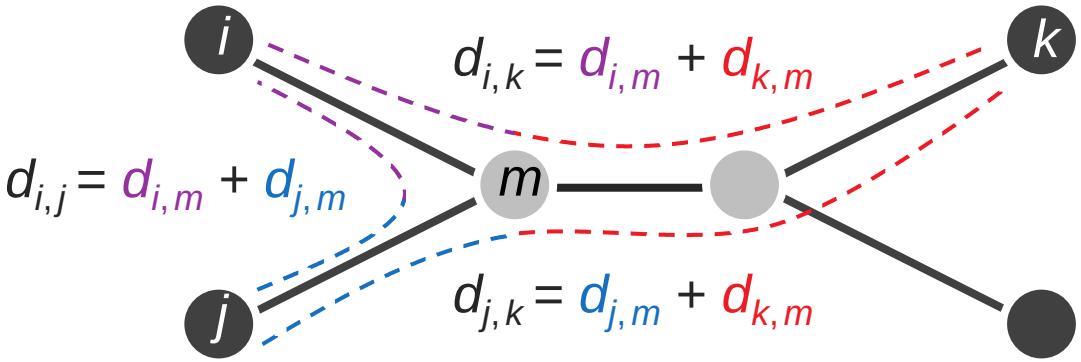
j i ***k*** **nisu** susedi!

Od suseda do spoljnih grana

Umesto traženja **suseda**, pokušajmo da izračunamo dužinu **spoljnih grana**, onih grana koje vode do listova.



Od suseda do spoljnih grana



$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

$$\therefore \quad d_{i,m} = D_{i,k} - (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2$$

Važi pod pretpostavkom da su i i j susedni listovi

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana:
LimbLength(i) je jednako minimalnoj vrednosti ($D_{i,k} + D_{i,j} - D_{j,k}$)/2 po svim listovima j i k.

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana:
LimbLength(*i*) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima *j* i *k*.

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana:
LimbLength(i) je jednako minimalnoj vrednosti ($D_{i,k} + D_{i,j} - D_{j,k}$)/2 po svim listovima j i k.

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

$$(D_{\text{chimp, human}} + D_{\text{chimp, seal}} - D_{\text{human, seal}}) \;/\; 2 = (3 + 6 - 7) \;/\; 2 = 1$$

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana: *LimbLength*(i) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima j i k.

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

$$(D_{\text{chimp, human}} + D_{\text{chimp, seal}} - D_{\text{human, seal}}) / 2 = (3 + 6 - 7) / 2 = 1$$

$$(D_{\text{chimp, human}} + D_{\text{chimp, whale}} - D_{\text{human, whale}}) / 2 = (3 + 4 - 5) / 2 = 1$$

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana:
LimbLength(i) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima j i k.

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

$(D_{chimp,\ human} + D_{chimp,\ seal} - D_{human,\ seal}) / 2 = (3 + 6 - 7) / 2 = 1$

$(D_{chimp,\ human} + D_{chimp,\ whale} - D_{human,\ whale}) / 2 = (3 + 4 - 5) / 2 = 1$

$(D_{chimp,\ whale} + D_{chimp,\ seal} - D_{whale,\ seal}) / 2 = (6 + 4 - 2) / 2 = 4$

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana: *LimbLength*(i) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima j i k.

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

$(D_{human,\ chimp} + D_{chimp,\ seal} - D_{human,\ seal}) / 2 = (3 + 6 - 7) / 2 = 1$

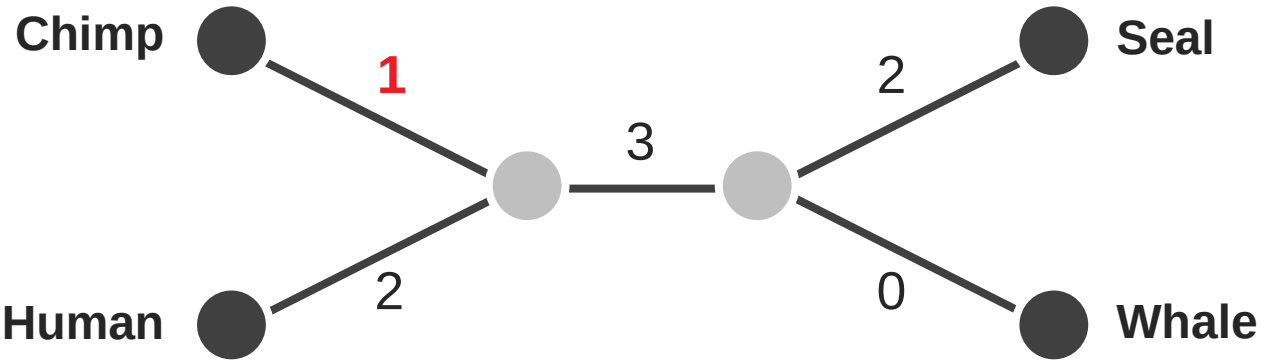
$(D_{human,\ chimp} + D_{chimp,\ whale} - D_{human,\ whale}) / 2 = (3 + 4 - 5) / 2 = 1$

$(D_{whale,\ chimp} + D_{chimp,\ seal} - D_{whale,\ seal}) / 2 = (6 + 4 - 2) / 2 = 4$

Računanje dužine spoljnih grana

Teorema o dužini spoljnih grana:
LimbLength(*i*) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima *j* i *k*.

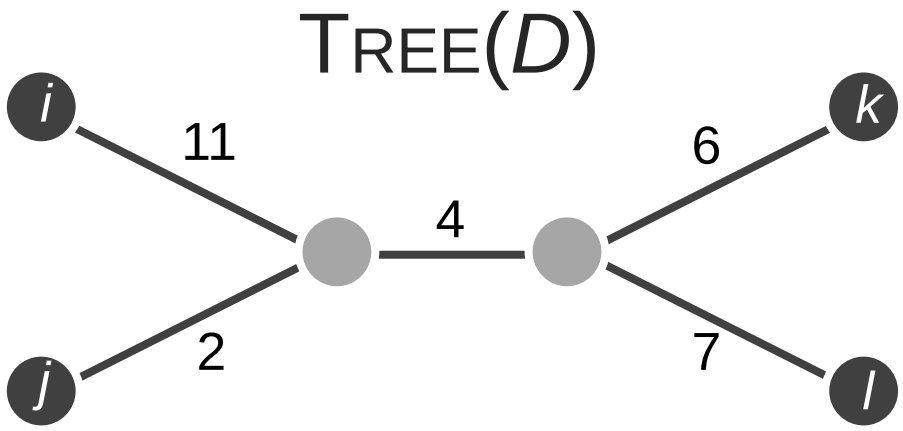
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



AdditivePhylogeny

D

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0



AdditivePhylogeny

D

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

1. Izaberemo proizvoljno list, npr. j.

AdditivePhylogeny

D

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

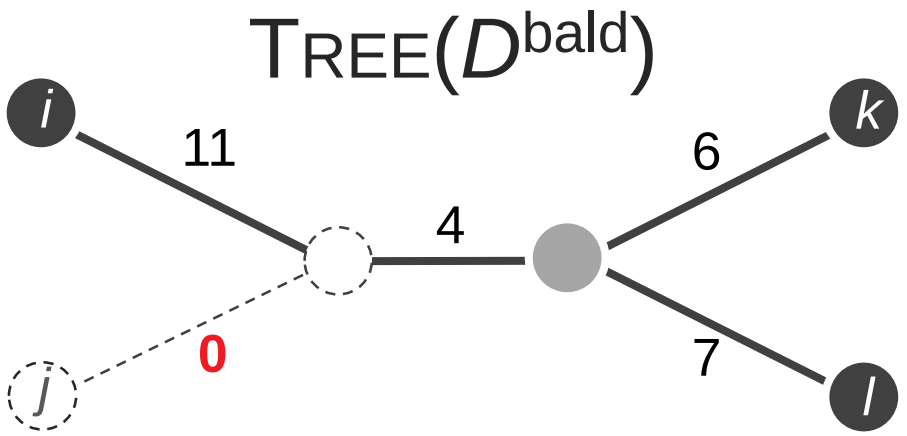
LimbLength(*j*) = 2

2. Izračunamo dužinu njegove krajnje grane, *LimbLength*(*j*).

AdditivePhylogeny

D^{bald}

	i	*j*	*k*	*l*
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0



3. Oduzmemo *LimbLength*(*j*) od svake grane i dobijemo matricu D^{bald} u kojoj do lista *j* vodi ogoljena (*bold*) grana (dužine 0).

AdditivePhylogeny

D^{trim}

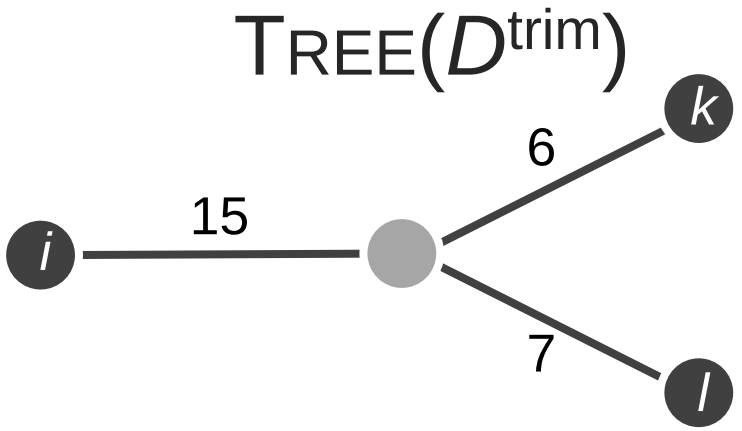
	i	*j*	***k***	***l***
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0

4. Uklonimo j-ti red i kolonu iz matrice i dobijemo $(n - 1) \times (n - 1)$ matricu D^{trim}.

AdditivePhylogeny

D^{trim}

	i	*j*	*k*	*l*
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0

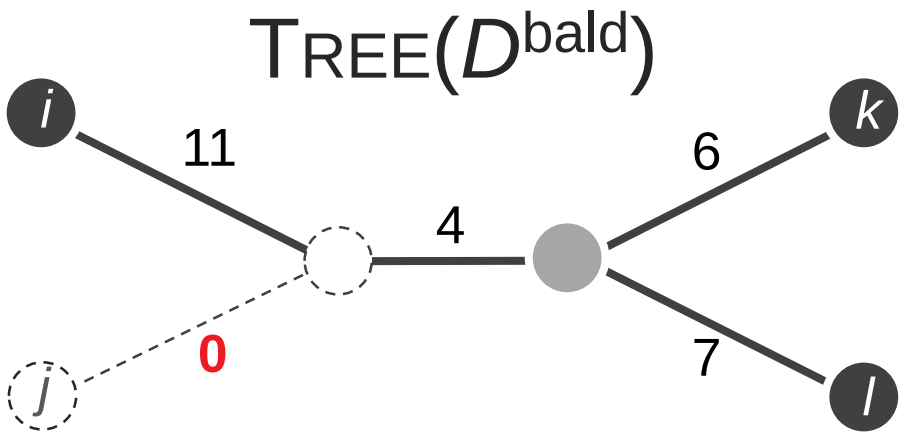


5. Konstruišemo *Tree*(D^{trim}).

AdditivePhylogeny

D^{bald}

	i	*j*	*k*	*l*
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0



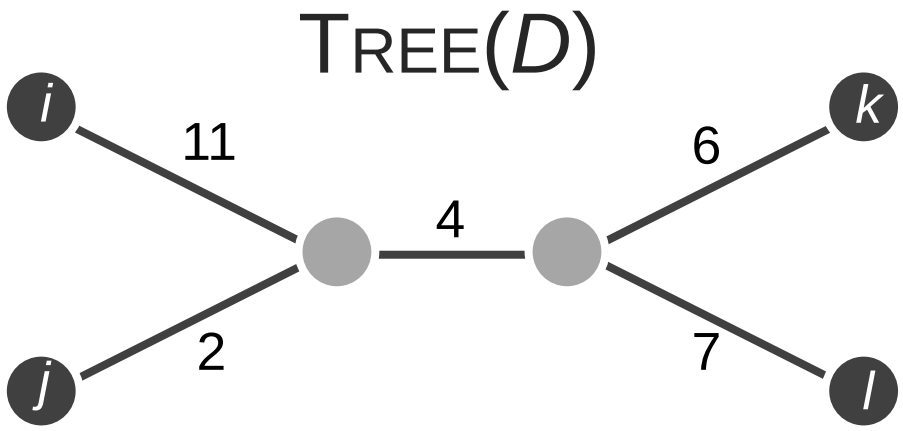
6. Identifikujemo tačku u *Tree*(D^{trim}) gde list *j* treba da se nalazi.

AdditivePhylogeny

D

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

LimbLength(*j*) = 2



7. Dodamo list *j* povezujući ga granom dužine *LimbLength*(*j*) kako bismo formirali *Tree*(*D*).

AdditivePhylogeny

***AdditivePhylogeny*(*D*):**

1. Izaberemo proizvoljno list, npr. *j*.
2. Izračunamo dužinu njegove krajnje grane, LimbLength(j).
3. Oduzmemo LimbLength(j) od svake grane i dobijemo matricu Dbald u kojoj do lista j vodi ogoljena (bold) grana (dužine 0).
4. Uklonimo j-ti red i kolonu iz matrice i dobijemo (n – 1) x (n – 1) matricu Dtrim.
5. Konstruišemo Tree(Dtrim).
6. Identifikujemo tačku u Tree(Dtrim) gde list j treba da se nalazi.
7. Dodamo list j povezujući ga granom dužine LimbLength(j) kako bismo formirali Tree(D).

AdditivePhylogeny

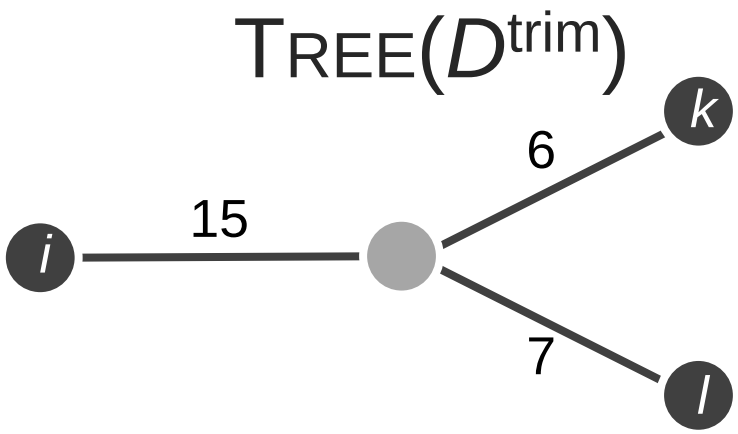
***AdditivePhylogeny*(*D*):**

1. Izaberemo proizvoljno list, npr. *j*.
2. Izračunamo dužinu njegove krajnje grane, LimbLength(j).
3. Oduzmemo LimbLength(j) od svake grane i dobijemo matricu Dbald u kojoj do lista j vodi ogoljena (bold) grana (dužine 0).
4. Uklonimo j-ti red i kolonu iz matrice i dobijemo (n – 1) x (n – 1) matricu Dtrim.
5. Konstruišemo Tree(Dtrim).
6. **Identifikujemo tačku u Tree(Dtrim) gde list j treba da se nalazi.**
7. Dodamo list j povezujući ga granom dužine LimbLength(j) kako bismo formirali Tree(D).

Povezivanje lista

D^{bald}

	i	*j*	*k*	*l*
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0

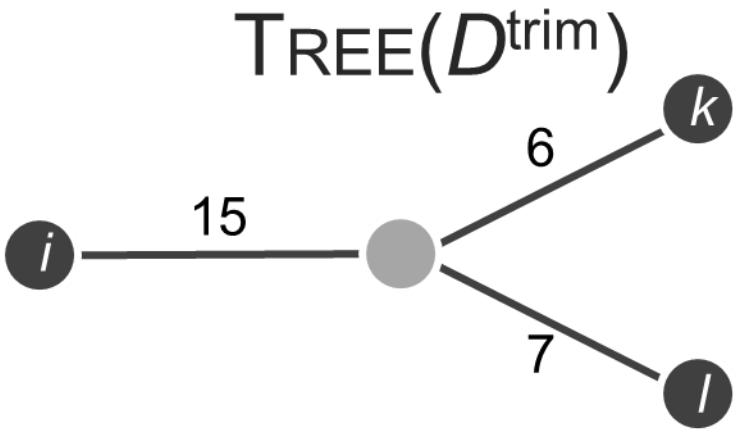


Teorema o dužini spoljnih grana:
LimbLength(*i*) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima *j* i *k*.

Povezivanje lista

D^{bald}

	i	*j*	*k*	*l*
i	0	**11**	**21**	22
j	11	0	**10**	11
k	21	10	0	13
l	22	11	13	0



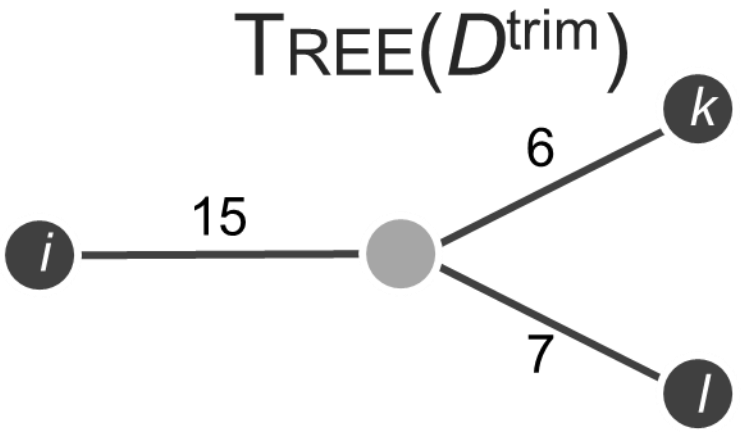
Teorema o dužini spoljnih grana:
LimbLength(*i*) je jednako minimalnoj vrednosti $(D_{i,k} + D_{i,j} - D_{j,k})/2$ po svim listovima *j* i *k*.

$$(D^{bald}_{i,j} + D^{bald}_{j,k} - D^{bald}_{i,k})/2 = \mathbf{0}$$

Povezivanje lista

D^{bald}

	i	*j*	*k*	*l*
i	0	**11**	**21**	22
j	11	0	**10**	11
k	21	10	0	13
l	22	11	13	0



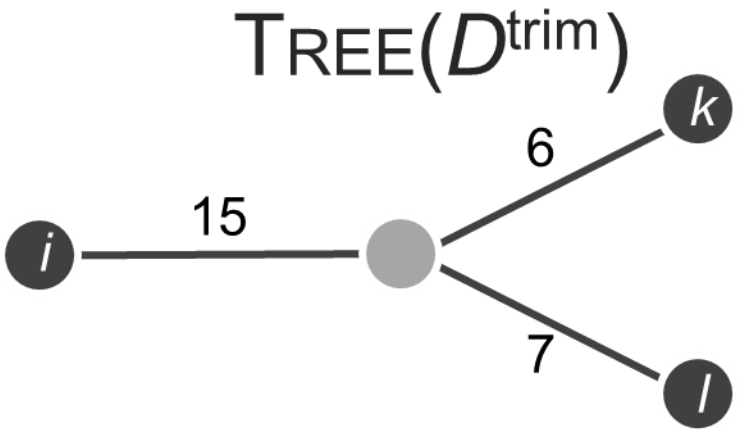
$$(D^{bald}_{i,j} + D^{bald}_{j,k} - D^{bald}_{i,k})/2 = \mathbf{0}$$

$$D^{bald}_{i,j} + D^{bald}_{j,k} = D^{bald}_{i,k}$$

Povezivanje lista

D^{bald}

	i	*j*	*k*	*l*
i	0	**11**	**21**	22
j	11	0	**10**	11
k	21	10	0	13
l	22	11	13	0



Tačka povezivanja za list *j* je na putanji između listova *i* i *k* na rastojanju $D^{bald}_{i,j}$ od *i*.

$$D^{bald}_{i,j} + D^{bald}_{j,k} = D^{bald}_{i,k}$$

Matrica rastojanja za *spike* protein

	Cow	Pig	Horse	Mouse	Dog	Cat	Turkey	Civet	Human
Cow	0	226	249	436	958	916	730	787	785
Pig	226	0	292	436	903	905	744	802	813
Horse	249	292	0	426	927	907	735	795	791
Mouse	436	436	426	0	917	946	725	767	782
Dog	958	903	927	917	0	706	730	844	846
Cat	916	905	907	946	706	0	736	840	836
Turkey	730	744	735	725	730	736	0	763	760
Civet	787	802	795	767	844	840	763	0	16
Human	785	813	791	782	846	836	760	16	0

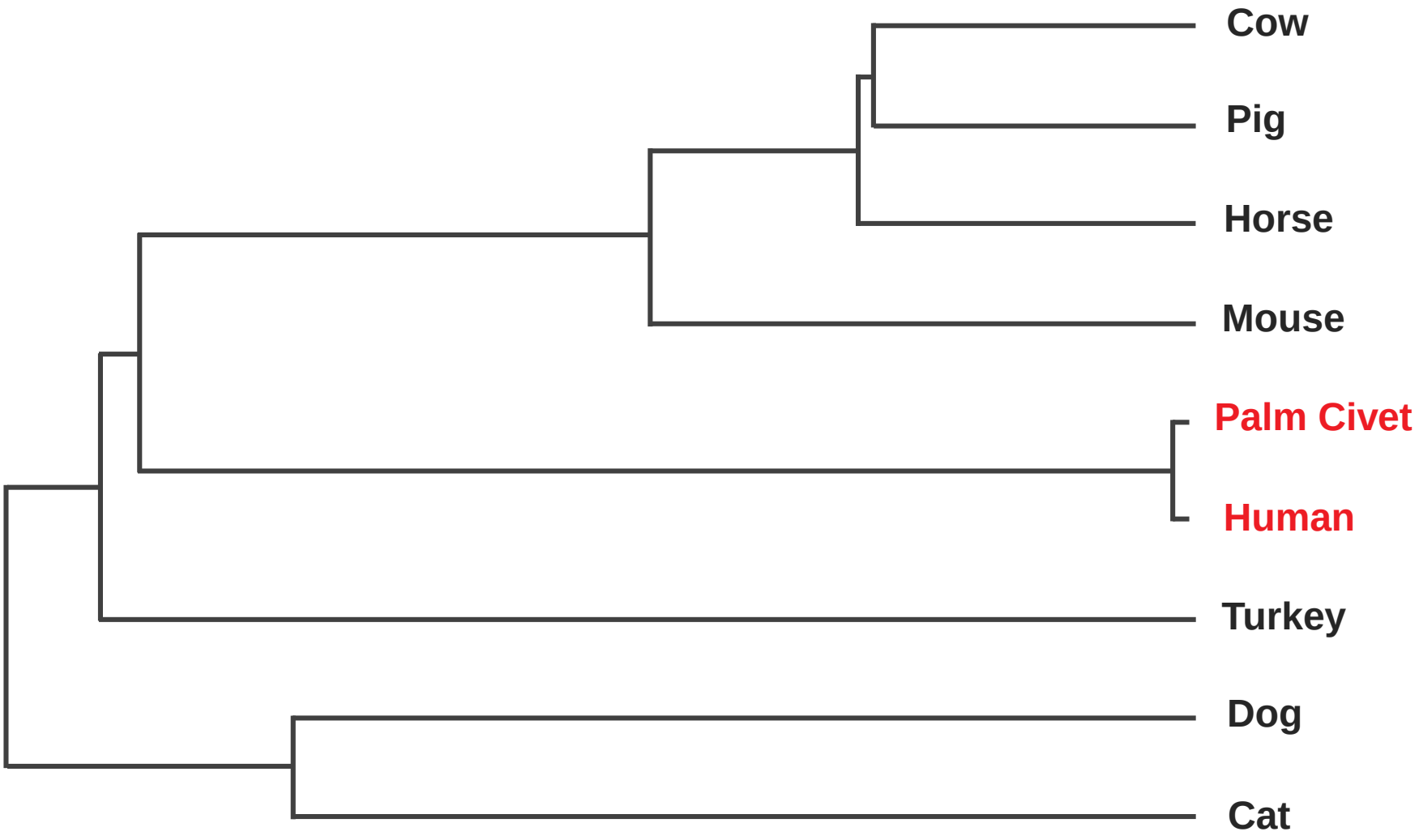
Nije aditivna! Malo je izmenimo...

Matrica rastojanja za *spike* protein

	Cow	Pig	Horse	Mouse	Dog	Cat	Turkey	Civet	Human
Cow	0	295	306	497	1081	1091	1003	956	954
Pig	295	0	309	500	1084	1094	1006	959	957
Horse	306	309	0	489	1073	1083	995	948	946
Mouse	497	500	489	0	1092	1102	1014	967	965
Dog	1081	1084	1073	1092	0	818	1056	1053	1051
Cat	1091	1094	1083	1102	818	0	1066	1063	1061
Turkey	1003	1006	995	1014	1056	1066	0	975	973
Civet	956	959	948	967	1053	1063	975	0	**16**
Human	954	957	946	965	1051	1061	973	16	0

Koja životinja nam je donela SARS?

Filogeneza coronavirus-a



Matrica rastojanja za *spike* protein

	Cow	Pig	Horse	Mouse	Dog	Cat	Turkey	Civet	Human
Cow	0	226	249	436	958	916	730	787	785
Pig	226	0	292	436	903	905	744	802	813
Horse	249	292	0	426	927	907	735	795	791
Mouse	436	436	426	0	917	946	725	767	782
Dog	958	903	927	917	0	706	730	844	846
Cat	916	905	907	946	706	0	736	840	836
Turkey	730	744	735	725	730	736	0	763	760
Civet	787	802	795	767	844	840	763	0	16
Human	785	813	791	782	846	836	760	16	0

Šta ako matrica nije aditivna?

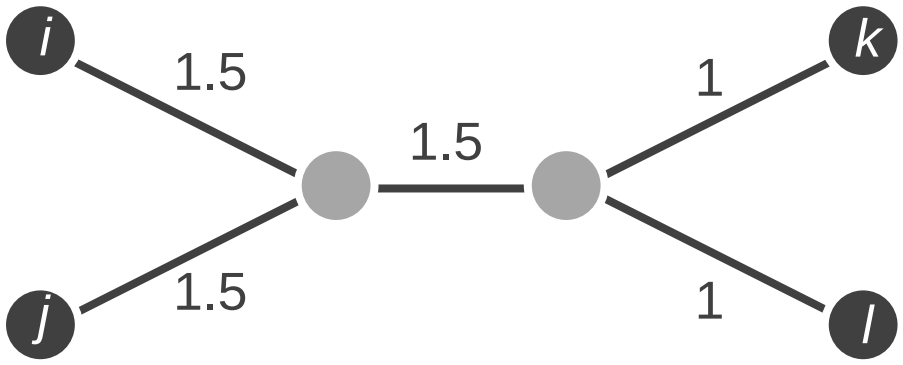
Pregled

- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- AdditivePhylogeny algoritam
- **Metod najmanjih kvadrata**
- Ultrametrična evolutivna stabla
- Neighbour-Joining algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

Metod najmanjih kvadrata

- Šta ako imamo matricu koja nije aditivna?

T

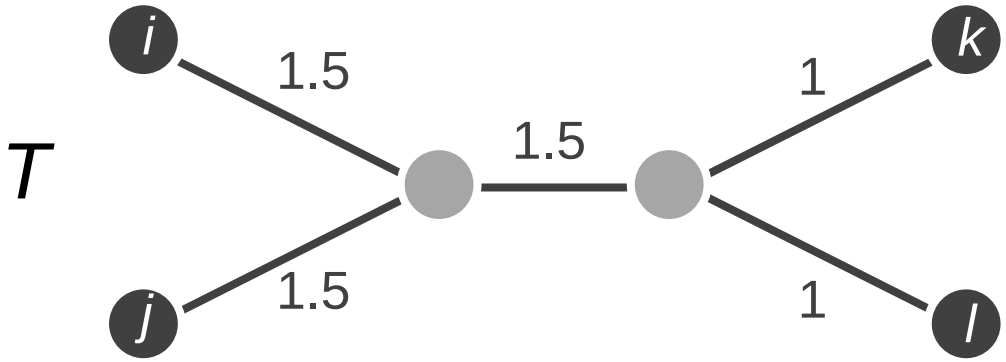


D

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

Metod najmanjih kvadrata

- Šta ako imamo matricu koja nije aditivna?
- Uzmemo u obzir njenu aproksimaciju nekom aditivnom matricom



D

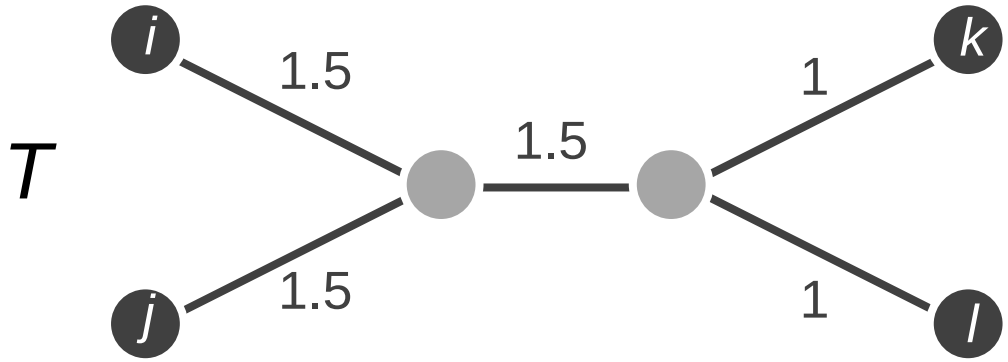
	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

d

	i	*j*	*k*	*l*
i	0	3	4	4
j	3	0	4	4
k	4	4	0	2
l	4	4	2	0

Metod najmanjih kvadrata

- Šta ako imamo matricu koja nije aditivna?
- Uzmemo u obzir njenu aproksimaciju nekom aditivnom matricom
- Koja je najbolja aproksimacija?



D

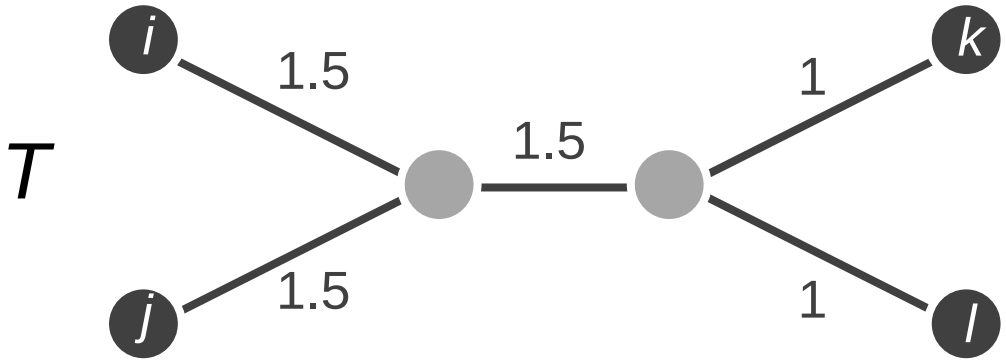
	i	*j*	*k*	*l*
i	0	3	4	**3**
j	3	0	4	**5**
k	4	4	0	2
l	3	5	2	0

d

	i	*j*	*k*	*l*
i	0	3	4	**4**
j	3	0	4	**4**
k	4	4	0	2
l	4	4	2	0

Metod najmanjih kvadrata

$$Discrepancy(T, D) = \Sigma_{1 \le i < j \le n} (d_{i,j}(T) - D_{i,j})^2$$



D

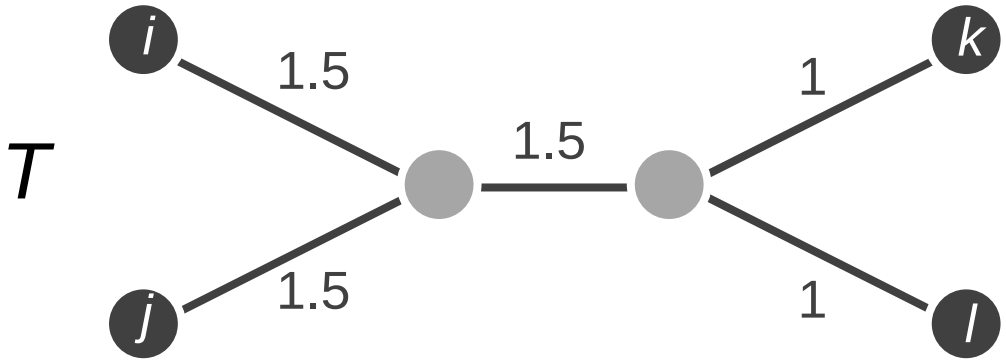
	i	j	k	l
i	0	3	4	**3**
j	3	0	4	**5**
k	4	4	0	2
l	3	5	2	0

d

	i	j	k	l
i	0	3	4	**4**
j	3	0	4	**4**
k	4	4	0	2
l	4	4	2	0

Metod najmanjih kvadrata

$$Discrepancy(T, D) = \Sigma_{1 \le i < j \le n} (d_{i,j}(T) - D_{i,j})^2$$
$$= 1^2 + 1^2 = 2$$



D

	i	j	k	l
i	0	3	4	**3**
j	3	0	4	**5**
k	4	4	0	2
l	3	5	2	0

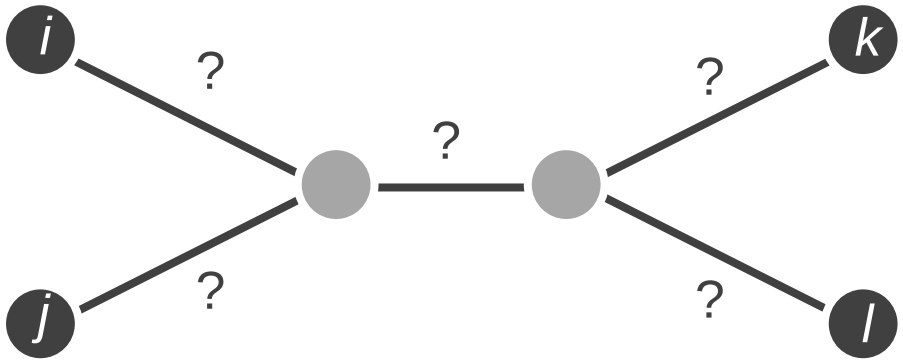
d

	i	j	k	l
i	0	3	4	**4**
j	3	0	4	**4**
k	4	4	0	2
l	4	4	2	0

Metod najmanjih kvadrata

Šta ako aditivna matrica nije poznata: Dodeliti dužine granama u stablu *T* tako da veličina *Discrepancy*(*T, D*) bude minimalna.

T



D

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

d

	i	*j*	*k*	*l*
i	0	?	?	?
j	?	0	?	?
k	?	?	0	?
l	?	?	?	0

Metod najmanjih kvadrata

- U opštem slučaju, za stablo date topologije postoji algoritam polinomijalne složenosti koji će dodeliti dužine granama stabla tako da diskrepanca bude minimalna
- Međutim, u praktičnim primenama neće biti poznata topologija stabla pa stoga moramo računati minimum po svim mogućim stablima
- Sa dodavanjem svakog lista u stablo, broj različitih topologija stabala raste eksponencijalno
- Problem minimizacije diskrepance po svim mogućim stablima je NP kompletan
- U nastavku, razmotrićemo dve heuristike za konstrukciju stabla na osnovu neaditivnih matrica

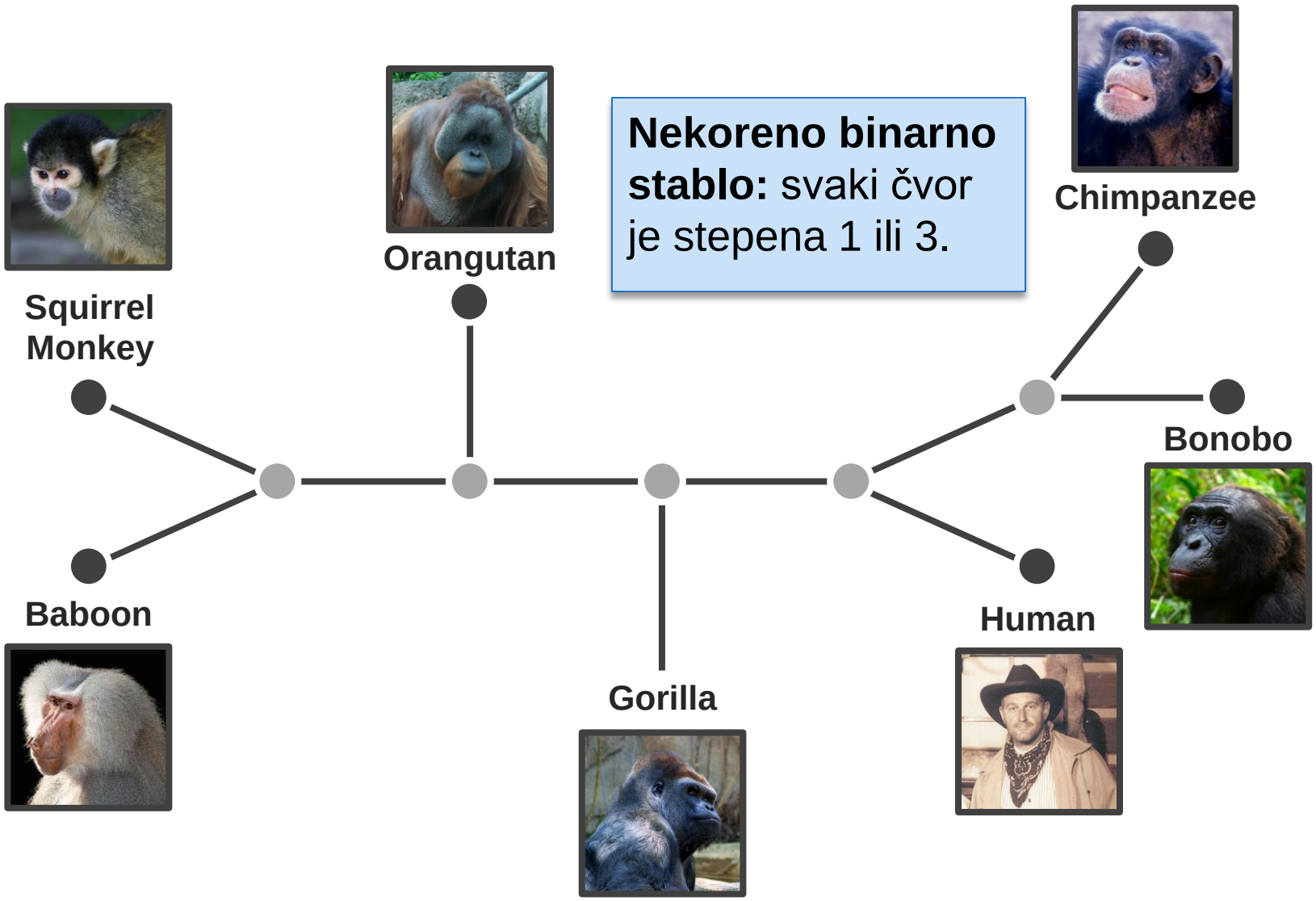
Pregled

- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- **Ultrametrična evolutivna stabla**
- Neighbour-Joining algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

Modelovanje specijacije

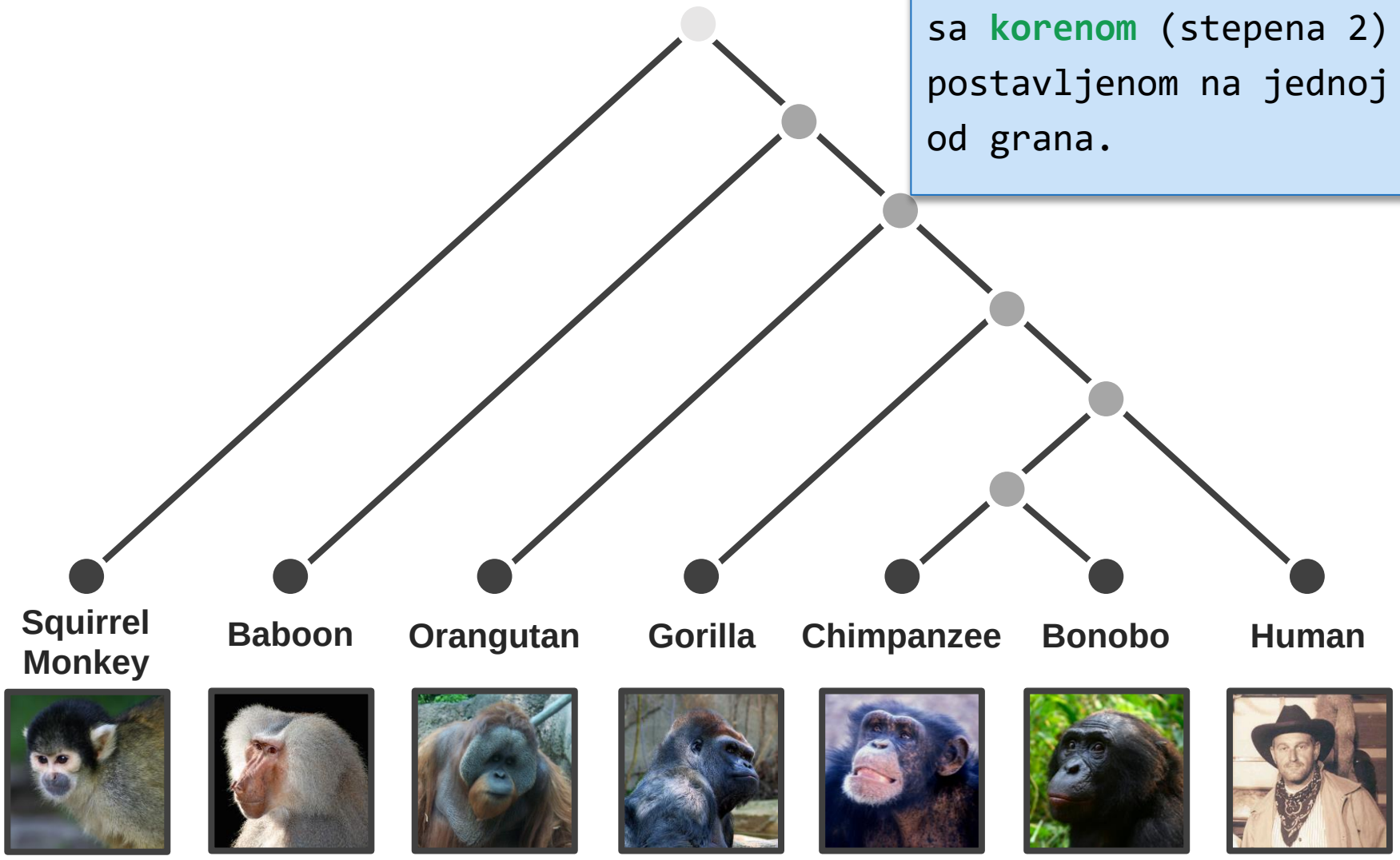
U praktičnim primenama, istraživači često pretpostavljaju da svaki unutrašnji čvor odgovara *specijaciji* kada se jedna vrsta deli na dve.

Modelovanje specijacije



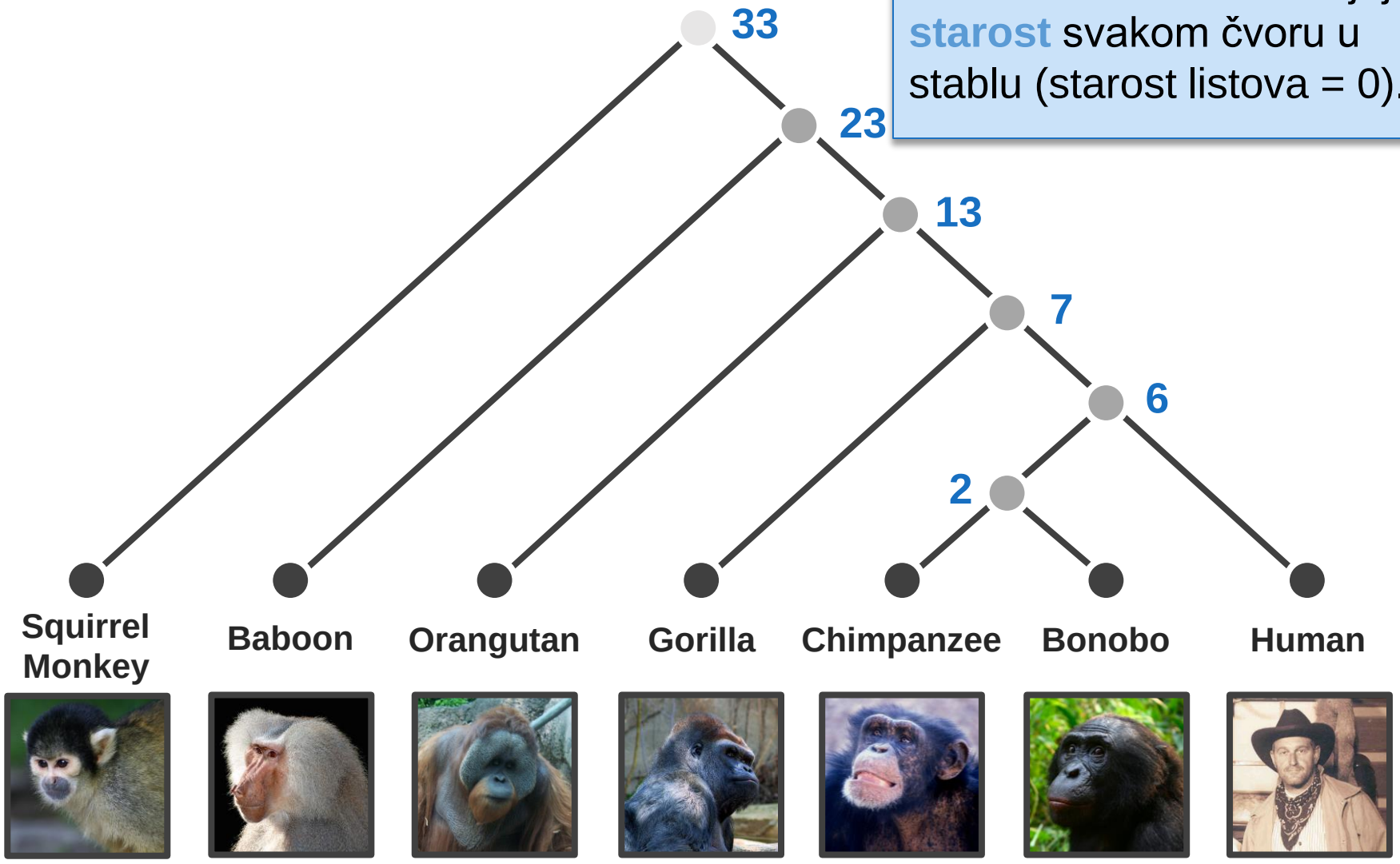
Modelovanje specijacije

Koreno binarno stablo: nekoreno binarno stablo sa **korenom** (stepena 2) postavljenom na jednoj od grana.



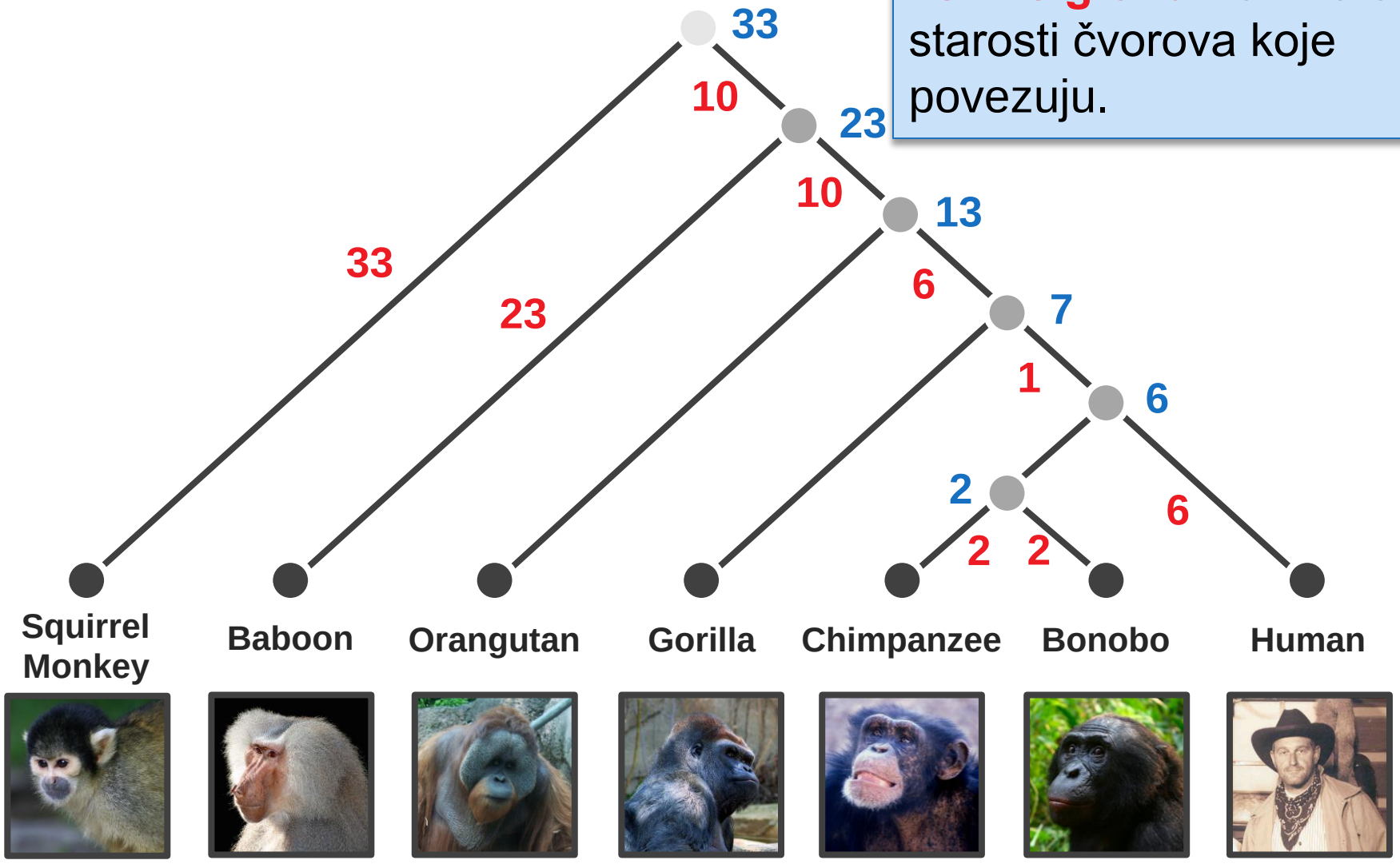
Ultrametrična stabla

Molekularni sat: dodeljuje **starost** svakom čvoru u stablu (starost listova = 0).



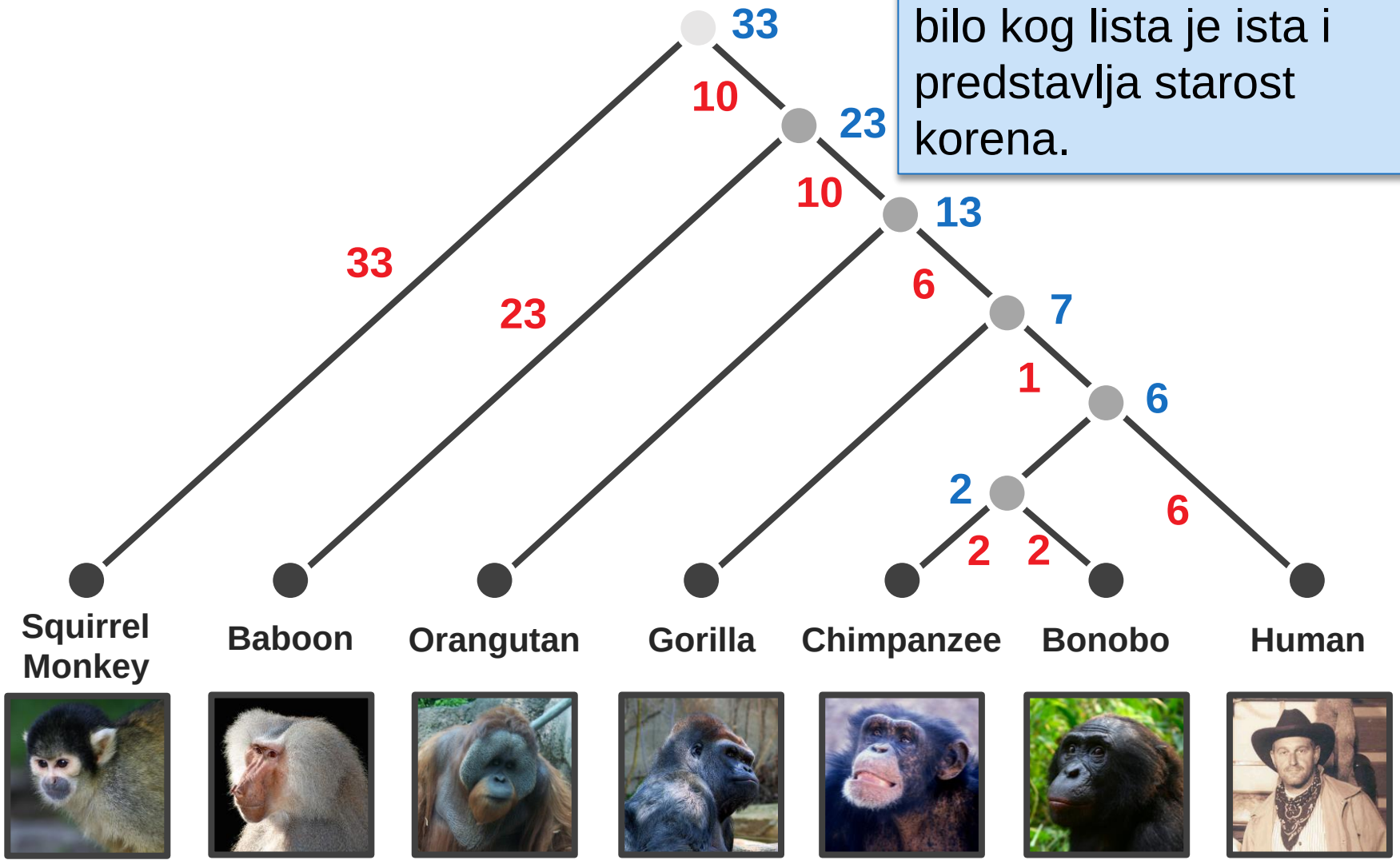
Ultrametrična stabla

Težine grana: razlika u starosti čvorova koje povezuju.



Ultrametrična stabla

Ultrametrično stablo: udaljenost od korena do bilo kog lista je ista i predstavlja starost korena.



UPGMA: heurističko klasterovanje

1. Formirati klaster za svaku današnju vrstu. Svaki klaster sadrži jedan list

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

UPGMA: heurističko klasterovanje

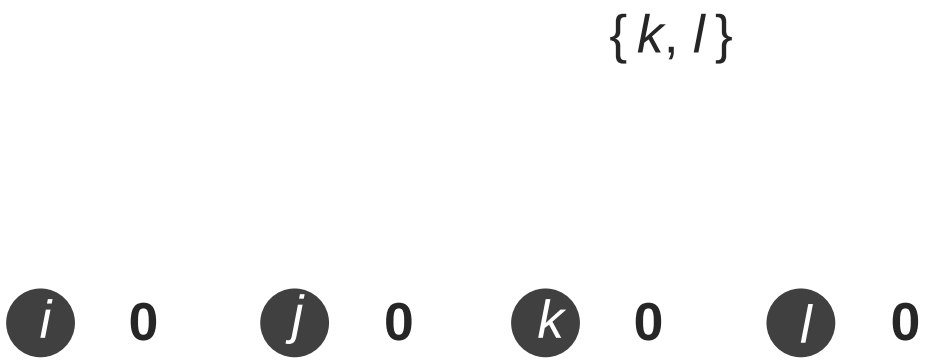
2. Naći dva najbliža klastera C_1 i C_2 na osnovu prosečnog rastojanja između njihovih članova
$D_{avg}(C_1, C_2) = \Sigma_{i \text{ in } C1,\ j \text{ in } C2}\ D_{i,j} \ /\ |C_1| \bullet |C_2|$
gde $|C|$ označava broj elemenata u klasteru C.

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	**2**
l	3	5	**2**	0

UPGMA: heurističko klasterovanje

3. Spojiti C_1 i C_2 u jedinstveni klaster C.

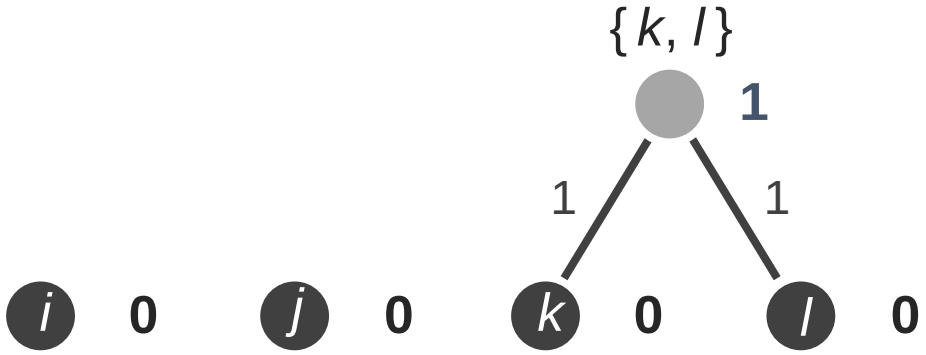
	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	**2**
l	3	5	**2**	0



UPGMA: heurističko klasterovanje

4. Formirati novi čvor za klaster i granama povezati ga za čvorovima. Postaviti starost čvora C na $D_{avg}(C_1, C_2)/2$.

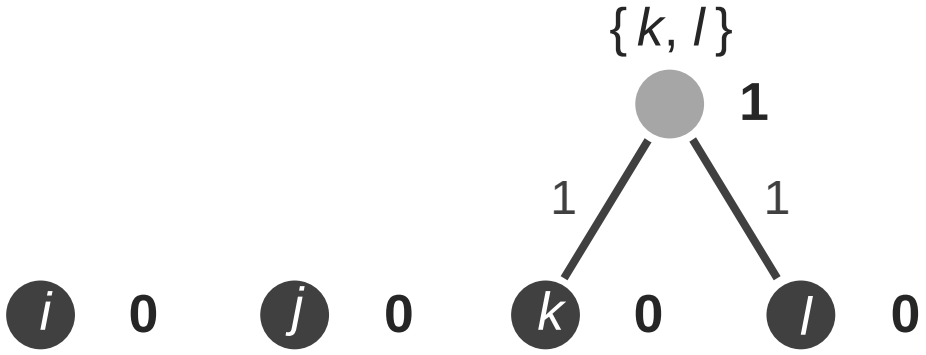
	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	**2**
l	3	5	**2**	0



UPGMA: heurističko klasterovanje

5. Ažurirati matricu rastojanja tako što ubacimo novi čvor, izbacimo čvorove koje on sadrži i izračunamo rastojanja kao prosečna rastojanja između svaka dva para klastera.

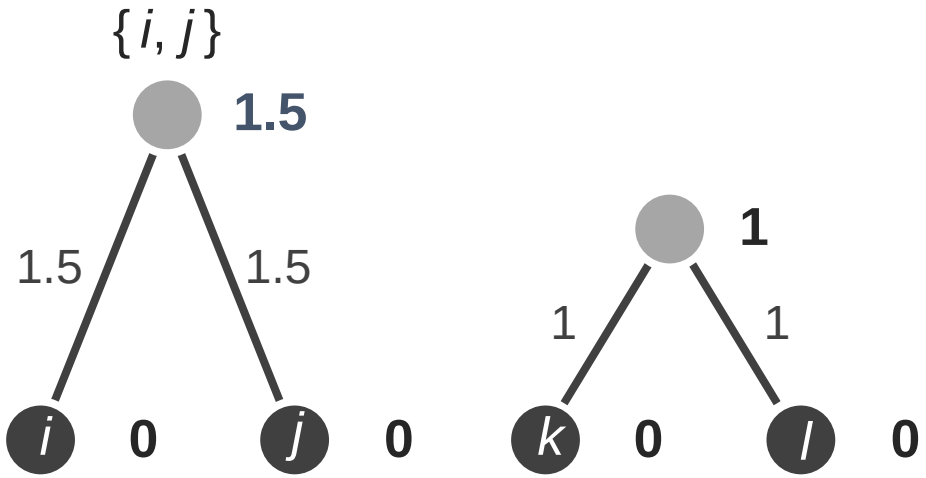
	i	j	$\{k, l\}$
i	0	3	3.5
j	3	0	4.5
$\{k, l\}$	3.5	4.5	0



UPGMA: heurističko klasterovanje

6. Iteriramo dok ne dođemo do jednog klastera koji sadrži sve vrste.

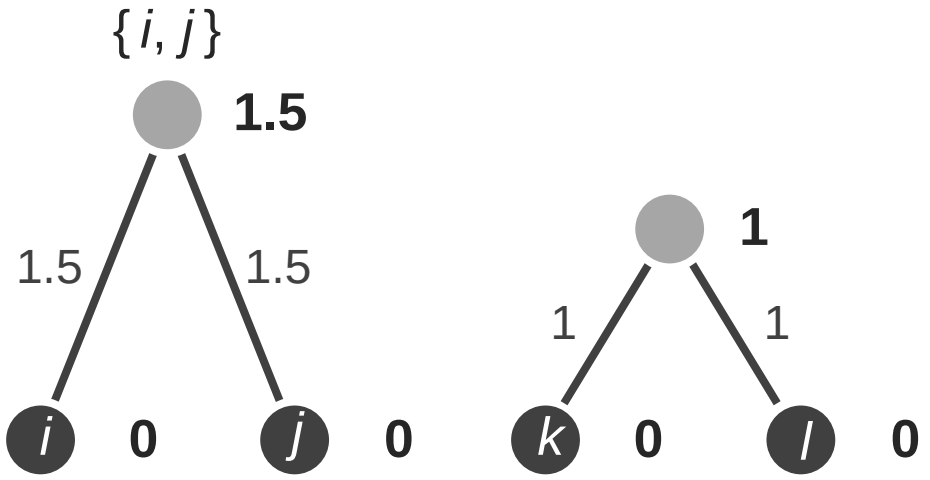
	i	*j*	{*k*, *l*}
i	0	**3**	3.5
j	**3**	0	4.5
{*k*, *l*}	3.5	4.5	0



UPGMA: heurističko klasterovanje

6. Iteriramo dok ne dođemo do jednog klastera koji sadrži sve vrste.

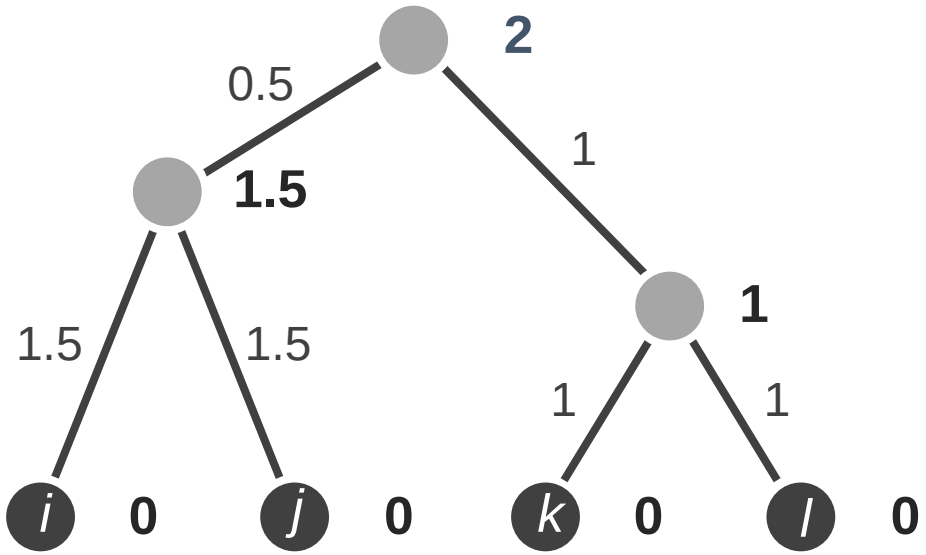
	$\{i, j\}$	$\{k, l\}$
$\{i, j\}$	0	4
$\{k, l\}$	4	0



UPGMA: heurističko klasterovanje

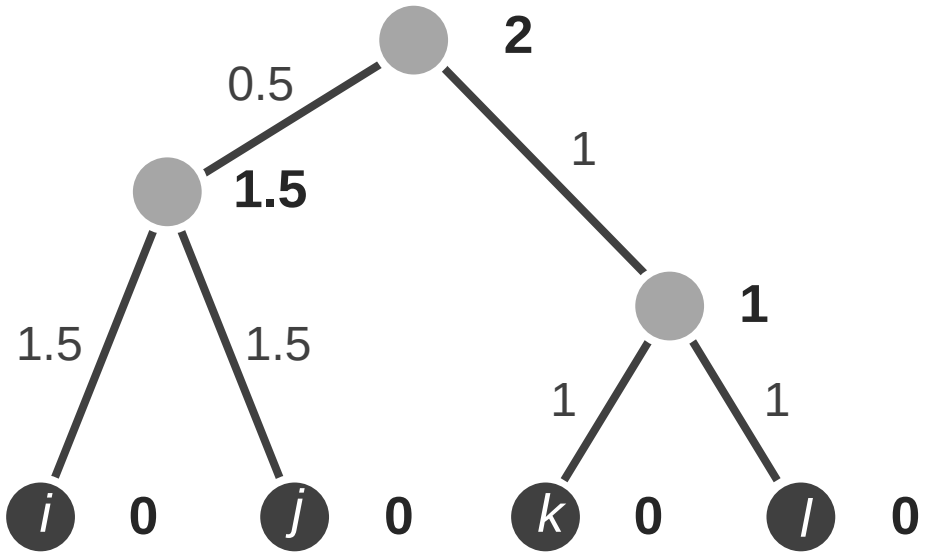
6. Iteriramo dok ne dođemo do jednog klastera koji sadrži sve vrste.

	{*i*, *j*}	{*k*, *l*}
{*i*, *j*}	0	4
{*k*, *l*}	4	0

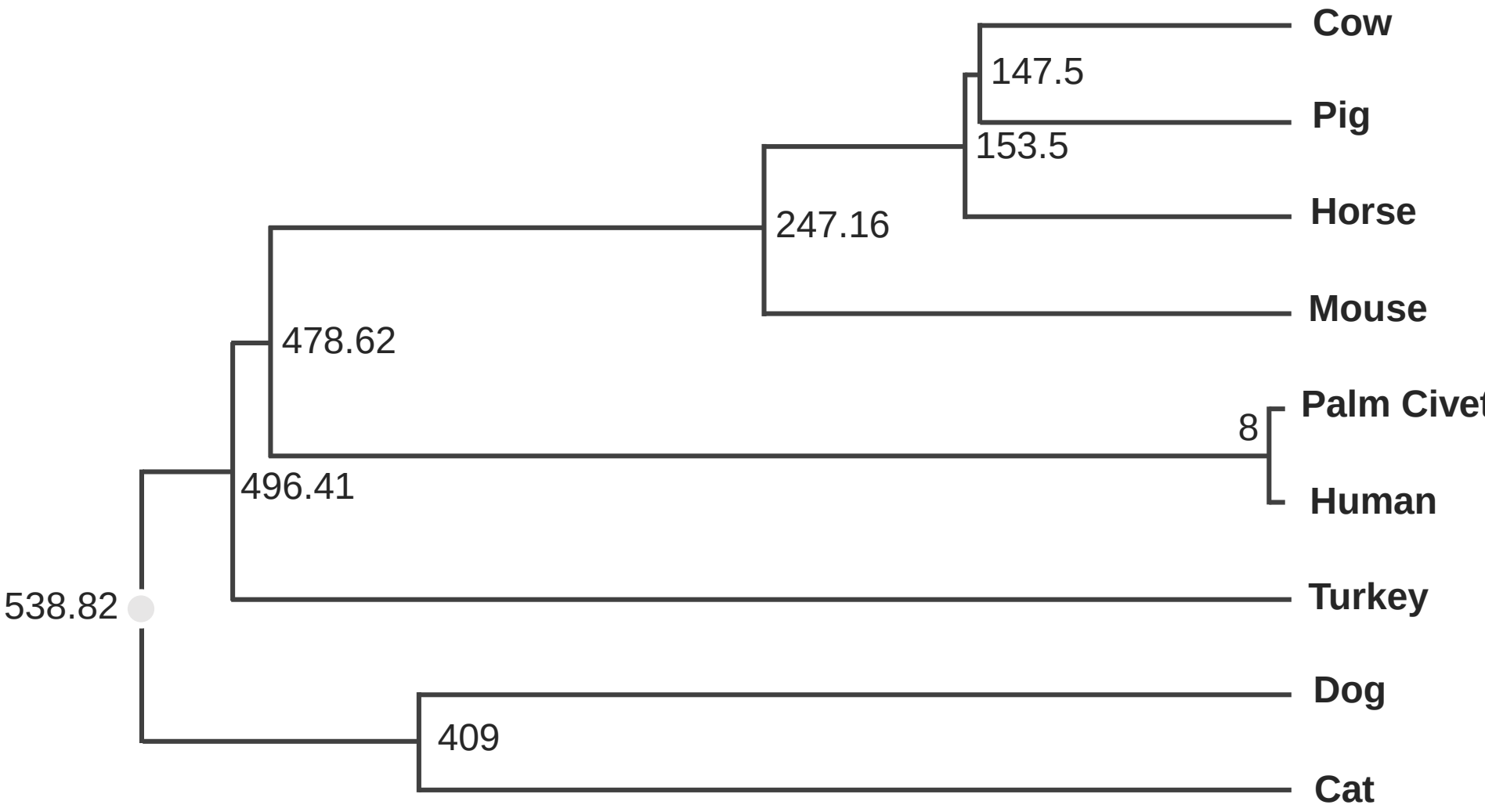


UPGMA: heurističko klasterovanje

6. Iteriramo dok ne dođemo do jednog klastera koji sadrži sve vrste.

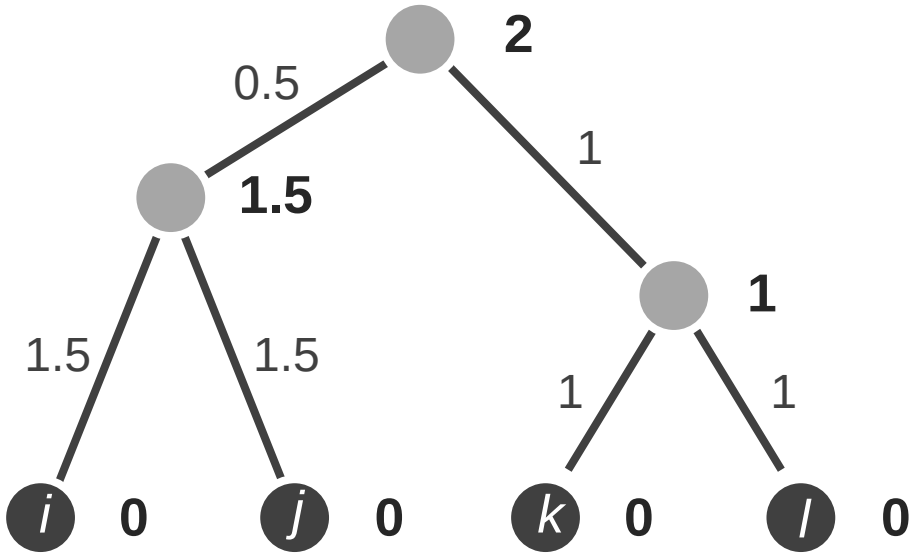


Primena UPGMA na *spike* proteine



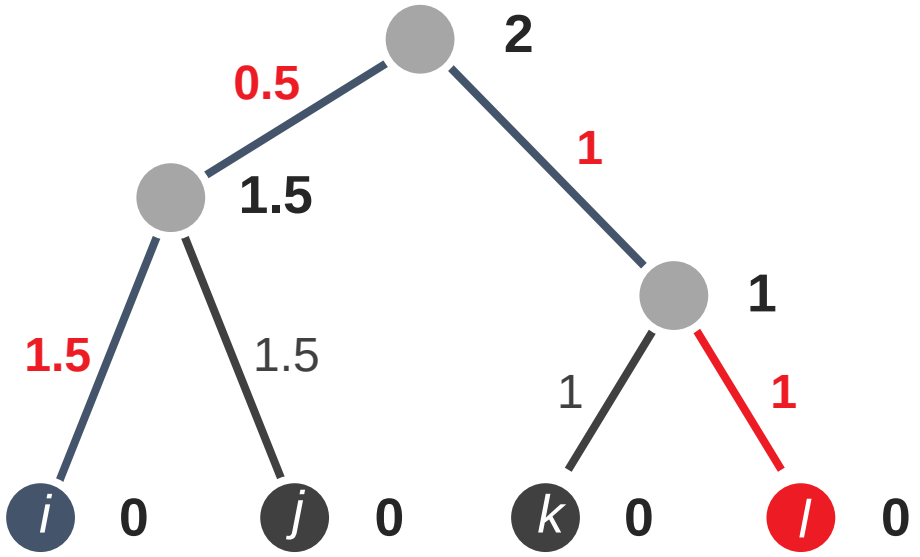
UPGMA ne proizvodi stablo koje odgovara matrici rastojanja

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0



UPGMA ne proizvodi stablo koje odgovara matrici rastojanja

	i	*j*	*k*	*l*
i	0	3	4	**3**
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0



Da rezimiramo...

- ***AdditivePhylogeny***:
  - dobra strana: kreira stablo koje odgovara aditivnoj matrici
  - loša strana: ne radi za neaditivne matrice
- **UPGMA:**
  - dobra strana: kreira stablo za svaku matricu
  - loša strana: stablo ne mora da odgovara aditivnoj matrici
- **?????:**
  - dobra strana: kreira stablo koje odgovara aditivnoj matrici
  - dobra strana: heuristika za neaditivne matrice

Pregled

- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- ***Neighbour-Joining* algoritam**
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- Problem velike parsimonije

Neighbour-Joining teorema

Za datu matricu rastojanja D dimenzije n x n, njena ***neighbour-joining*** **matrica** u oznaci D^* defiše se kao

$$D^*_{i,j}=(n-2)\cdot D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

gde je $TotalDistance_D(i)$ suma rastojanja od i do svih ostalih listova.

D

	i	*j*	*k*	*l*
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

TotalDistance$_D$
56
38
46
48

D^*

	i	*j*	*k*	*l*
i	0	**-68**	-60	-60
j	**-68**	0	-60	-60
k	-60	-60	0	**-68**
l	-60	-60	**-68**	0

Neighbour-Joining teorema

Neighbour-joining teorema: ako je matrica *D* aditivna, onda minimalni element matrice *D** odgovara susednim listovima u stablu *Tree(D)*.

D

	i	***j***	***k***	***l***	***TotalDistance*$_D$**
i	0	13	21	22	56
j	13	0	12	13	38
k	21	12	0	13	46
l	22	13	13	0	48

*D**

	i	***j***	***k***	***l***
i	0	**-68**	-60	-60
j	**-68**	0	-60	-60
k	-60	-60	0	**-68**
l	-60	-60	**-68**	0

Neighbour-Joining algoritam

		i	*j*	*k*	*l*	$TotalDistance_D$
	i	0	-68	-60	-60	56
*D**	*j*	-68	0	-60	-60	38
	k	-60	-60	0	-68	46
	l	-60	-60	-68	0	48

1. Konstruišemo *neighbour-joining* matricu *D** na osnovu matrice *D*.

Neighbour-Joining algoritam

		i	*j*	*k*	*l*	$TotalDistance_D$
	i	0	**-68**	-60	-60	56
D^*	*j*	**-68**	0	-60	-60	38
	k	-60	-60	0	**-68**	46
	l	-60	-60	**-68**	0	48

2. Nađemo minimalni element $D^*_{i,j}$ matrice D^*.

Neighbour-Joining algoritam

D^*

	i	*j*	*k*	*l*	$TotalDistance_D$
i	0	**-68**	-60	-60	**56**
j	**-68**	0	-60	-60	**38**
k	-60	-60	0	-68	46
l	-60	-60	-68	0	48

$\Delta_{i,j} = (\mathbf{56} - \mathbf{38}) / (4 - 2)$
$= 9$

3. Izračunamo $\Delta_{i,j} = (TotalDistance_D(i) - TotalDistance_D(j)) / (n - 2)$.

Neighbour-Joining algoritam

D

	i	*j*	*k*	*l*	$TotalDistance_D$
i	0	**13**	21	22	56
j	**13**	0	12	13	38
k	21	12	0	13	46
l	22	13	13	0	48

$\Delta_{i,j} = (56 - 38) / (4 - 2) = 9$

LimbLength(*i*) = ½(13 + 9) = 11

LimbLength(*i*) = ½(13 – 9) = 2

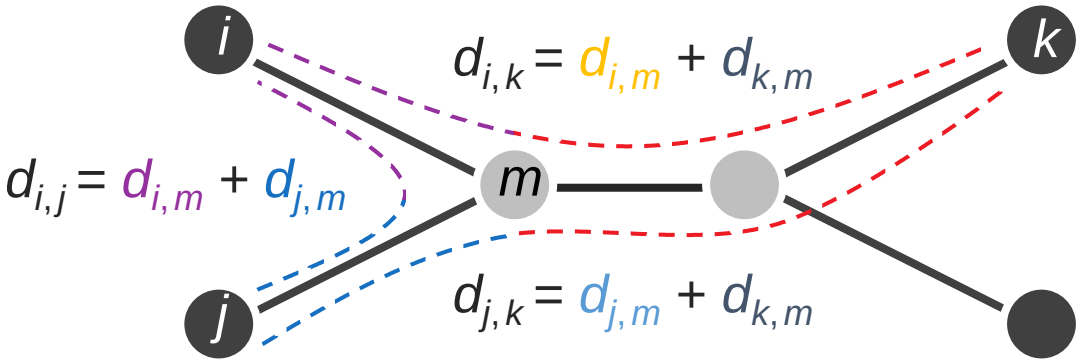
4. Postavimo *LimbLength*(*i*) na ½($D_{i,j}$ + $\Delta_{i,j}$) i *LimbLength*(*j*) na ½($D_{i,j}$ – $\Delta_{i,j}$).

Neighbour-Joining algoritam

		m	*k*	*l*	$TotalDistance_D$
	m	0	10	11	21
D'	***k***	10	0	13	23
	l	11	13	0	24

5. Formiramo matricu *D'* tako što uklonimo *i*-ti i *j*-ti red/kolonu iz *D* i dodamo *m*-ti red/kolonu tako da za svako k važi $D_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$.

Podsećanje: računanje of $d_{k,m}$



$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

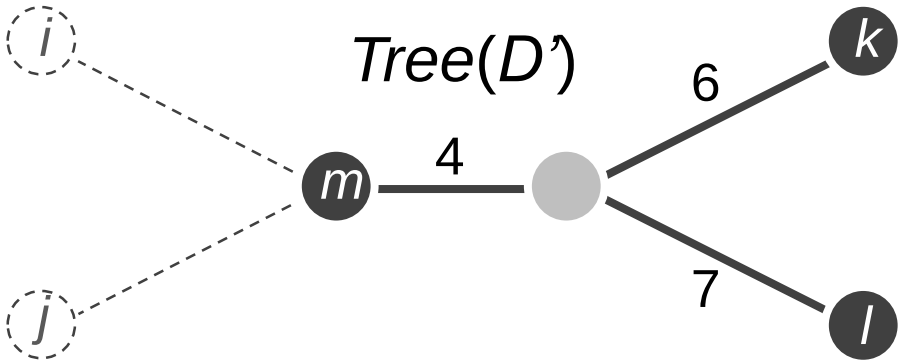
$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

Neighbour-Joining algoritam

D’

	m	***k***	***l***
m	0	10	11
k	10	0	13
l	11	13	0

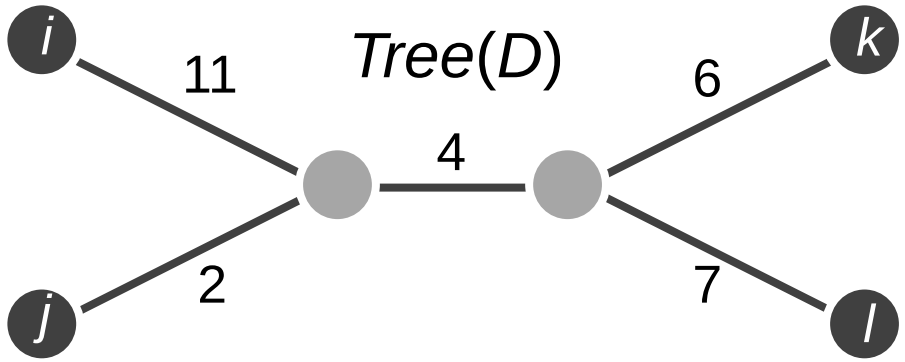


6. Primenimo ***NeighborJoining*** rekurzivno na *D’* da dobijemo *Tree*(*D’*).

Neighbour-Joining algoritam

D'

	m	*k*	*l*
m	0	10	11
k	10	0	13
l	11	13	0



LimbLength(*i*) = ½(13 + 9) = 11

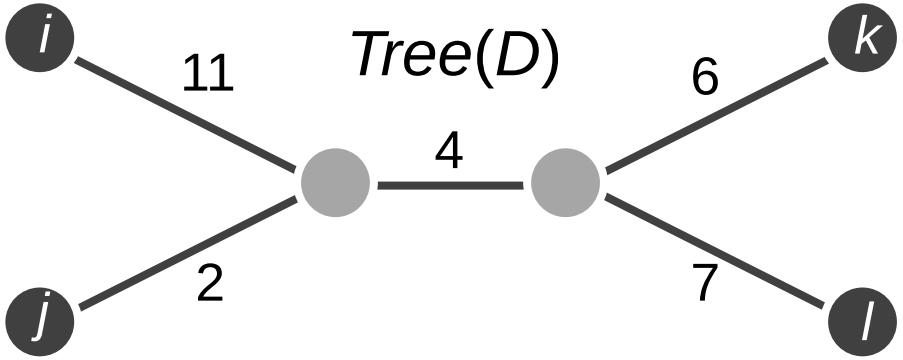
LimbLength(*i*) = ½(13 – 9) = 2

7. Vratimo krajnje grane do čvorova *i* i *j* i dobijemo *Tree*(*D*).

Neighbour-Joining algoritam

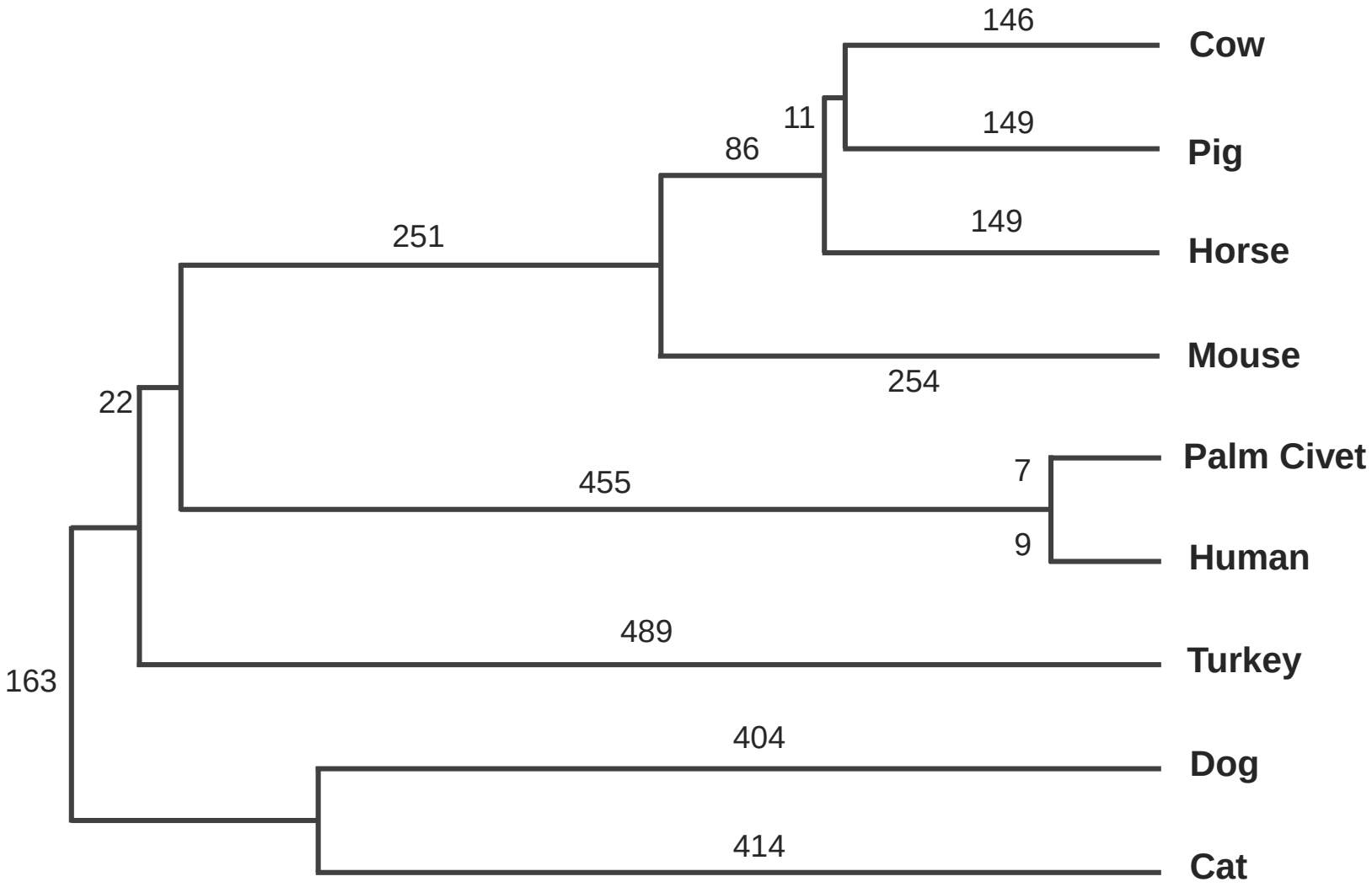
D'

	m	***k***	***l***
m	0	10	11
k	10	0	13
l	11	13	0



7. Vratimo krajnje grane do čvorova *i* i *j* i dobijemo *Tree*(*D*).

Neighbour-Joining za coronavirus



Slabosti metoda zasnovanih na rastojanju

Kada višestruko poravnanje zamenimo matricom rastojanja, gubimo informacije o sekvencama iz poravnanja

Zbog toga ne možemo da zaključimo kakva je sekvenca odgovarala vrstama iz unutrašnjih čvorova

Species	Alignment	Distance Matrix			
		Chimp	**Human**	**Seal**	**Whale**
Chimp	ACGTAGGCCT	0	3	6	4
Human	ATGTAAGACT	3	0	7	5
Seal	TCGAGAGCAC	6	7	0	2
Whale	TCGAAAGCAT	4	5	2	0

Pregled

- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- *Neighbour-Joining* algoritam
- **Rekonstrukcija stabla na osnovu karakteristika**
- Problem male parsimonije
- Problem velike parsimonije

Tabele karakteristika

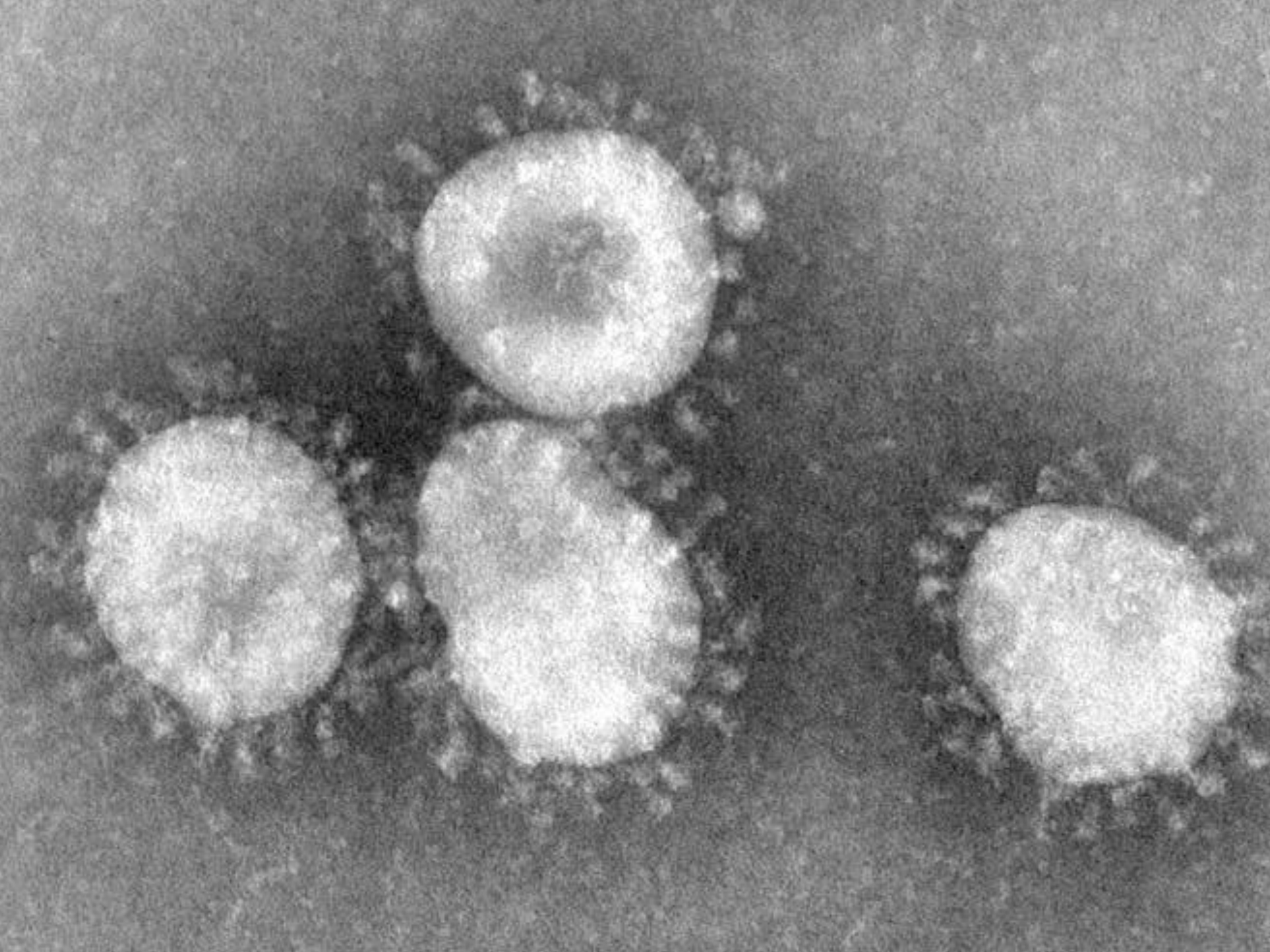
Pre oko pedeset godina, istraživači su konstruisali filogenetska stabla na osnovu anatomsko-fizioloških osobina organizama koje su nazvane **karakteristikama.**

	wings	legs
winged stick insect	Yes	6
wingless stick insect	No	6
giant centipede	No	42

Filogeneza na osnovu karakteristika

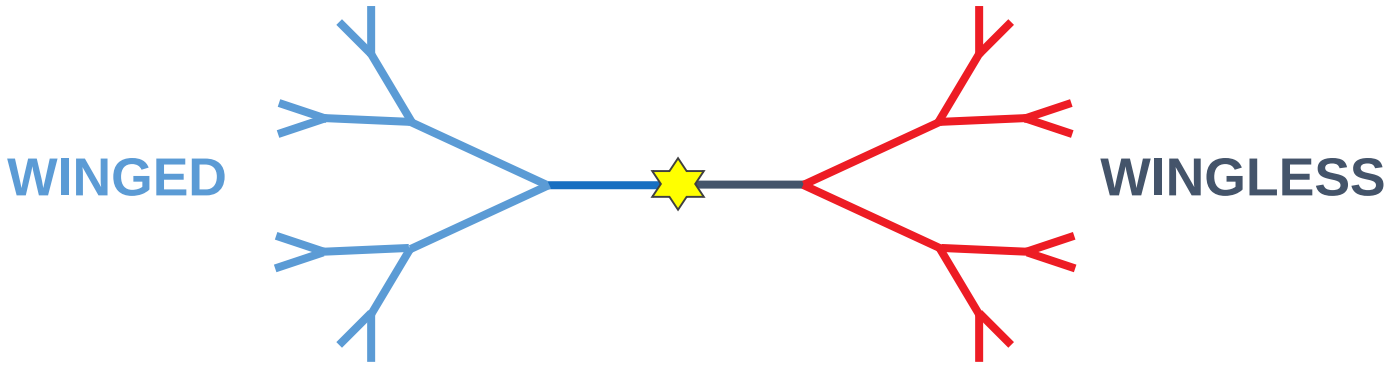
Problem filogeneze zasnovane na karakteristikama: *Rekonstruisati evolutivno stablo na osnovu karakteristika.*

- **Ulaz:** Tabela karakteristika *n* x *m* za *n* vrsta i *m* karakteristika.
- **Izlaz:** Stablo kod kog su vrste sa sličnim karakteristikama blizu jedna drugoj.



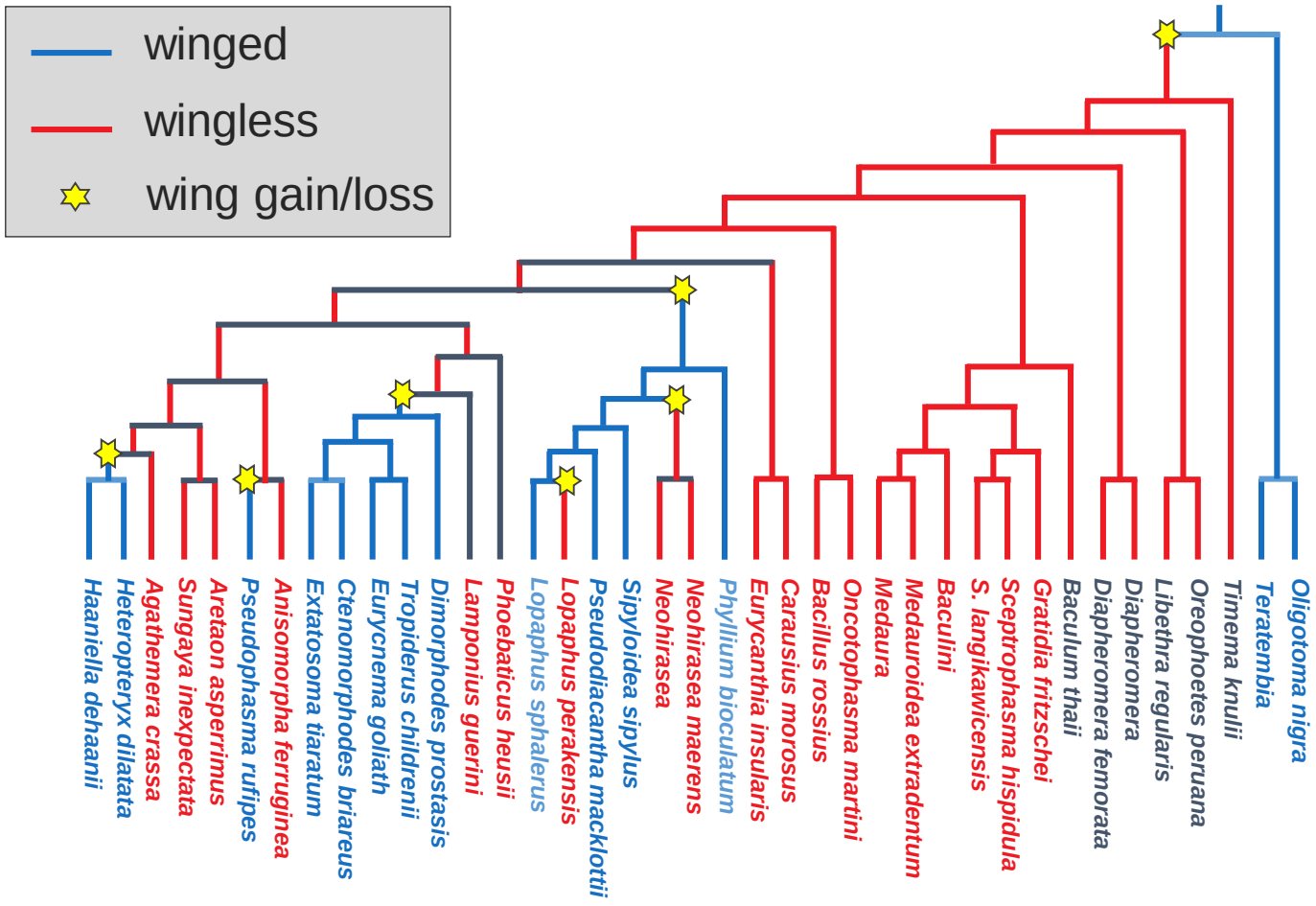
Od karakteristika do filogeneze

Kako bismo konstruisali evolutivno stablo na osnovu karakteristika?



Dolov zakon o nepovratnosti evolucionih procesa (1893): evolucija ne izmišlja dva puta isti organ (npr. krila kod insekata).

Izuzeci Dolovog zakona

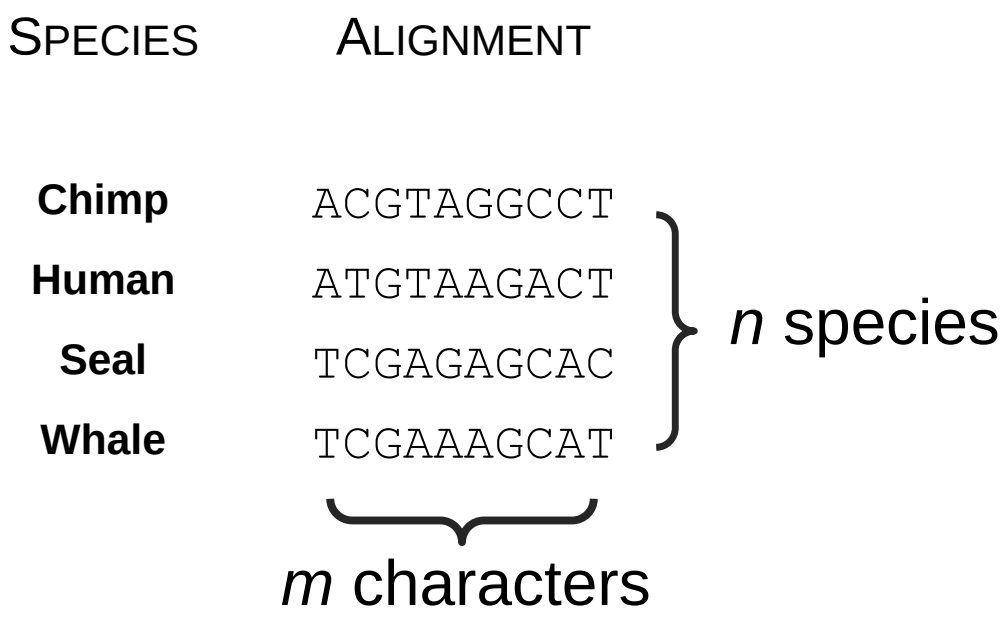


Zašto je došlo do ovoga?

Poravnanje kao tabela karakteristika

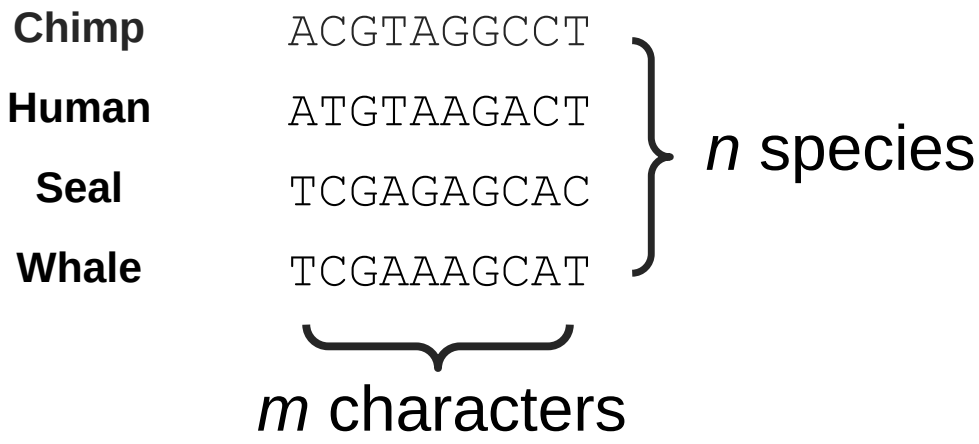
SPECIES	ALIGNMENT
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

Poravnanje kao tabela karakteristika

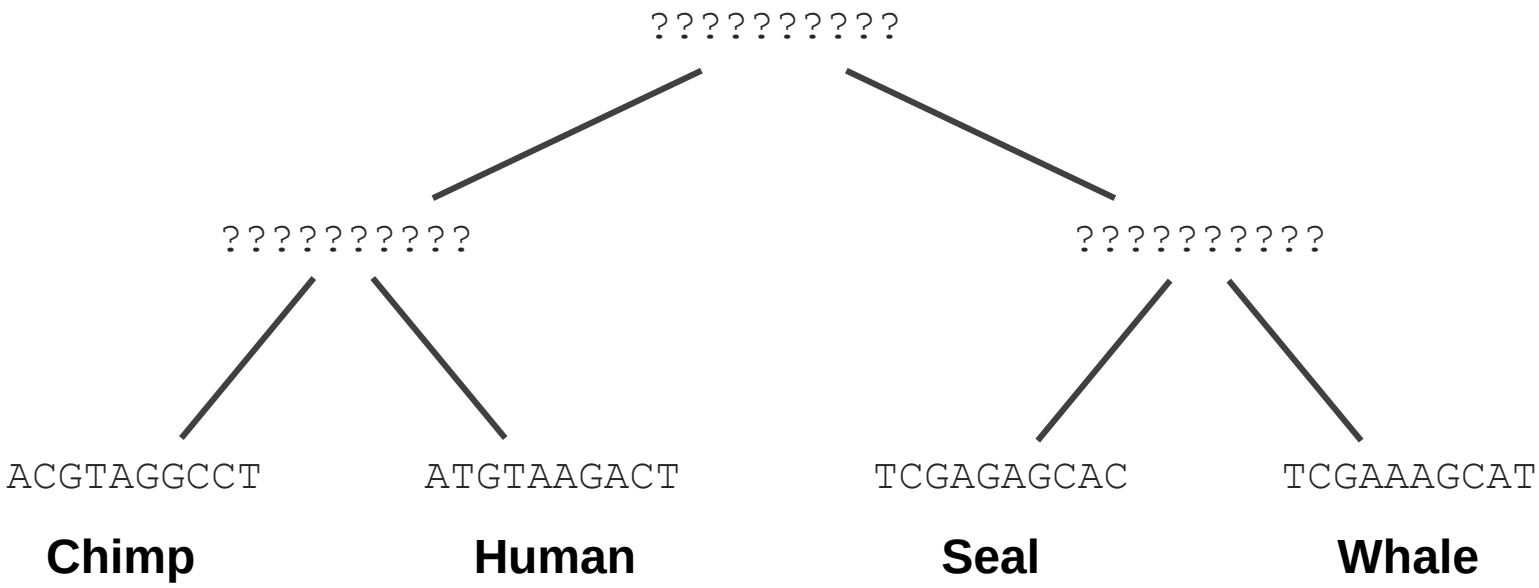


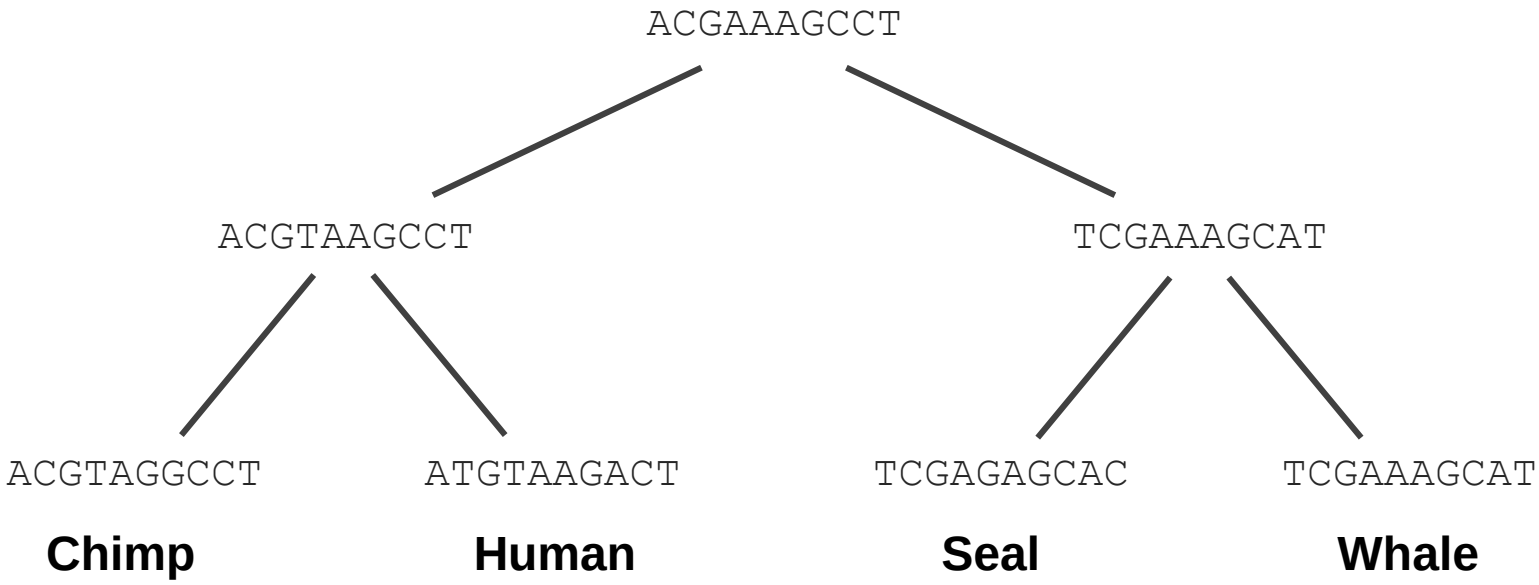
Pregled

- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- *Neighbour-Joining* algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- **Problem male parsimonije**
- Problem velike parsimonije


Chimp
Human
Seal
Whale
ACGTAGGCCT
ATGTAAGACT
TCGAGAGCAC
TCGAAAGCAT
n species
m characters

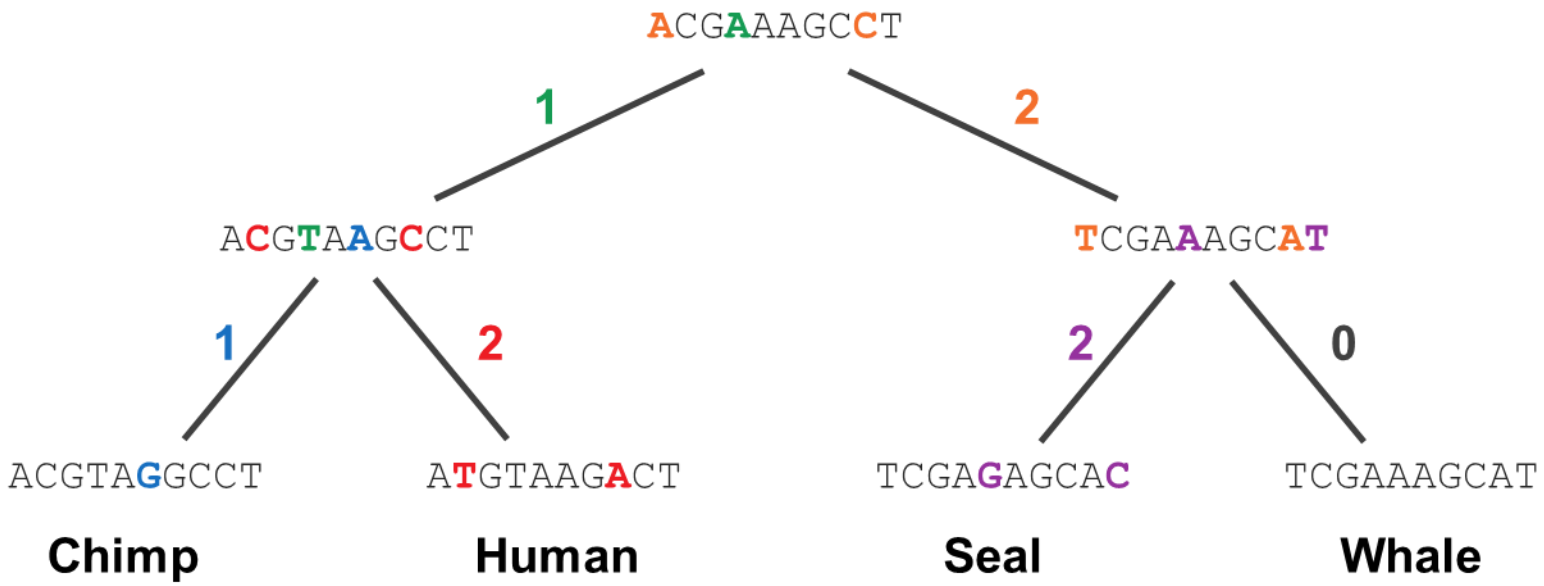
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT




ACGAAAGCCT
ACGTAAGCCT
TCGAAAGCAT
ACGTAGGCCT
ATGTAAGACT
TCGAGAGCAC
TCGAAAGCAT
Chimp
Human
Seal
Whale

Skor parsimonije: suma Hamingovih rastojanja duž svake grane.

Skor parsimonije: 8



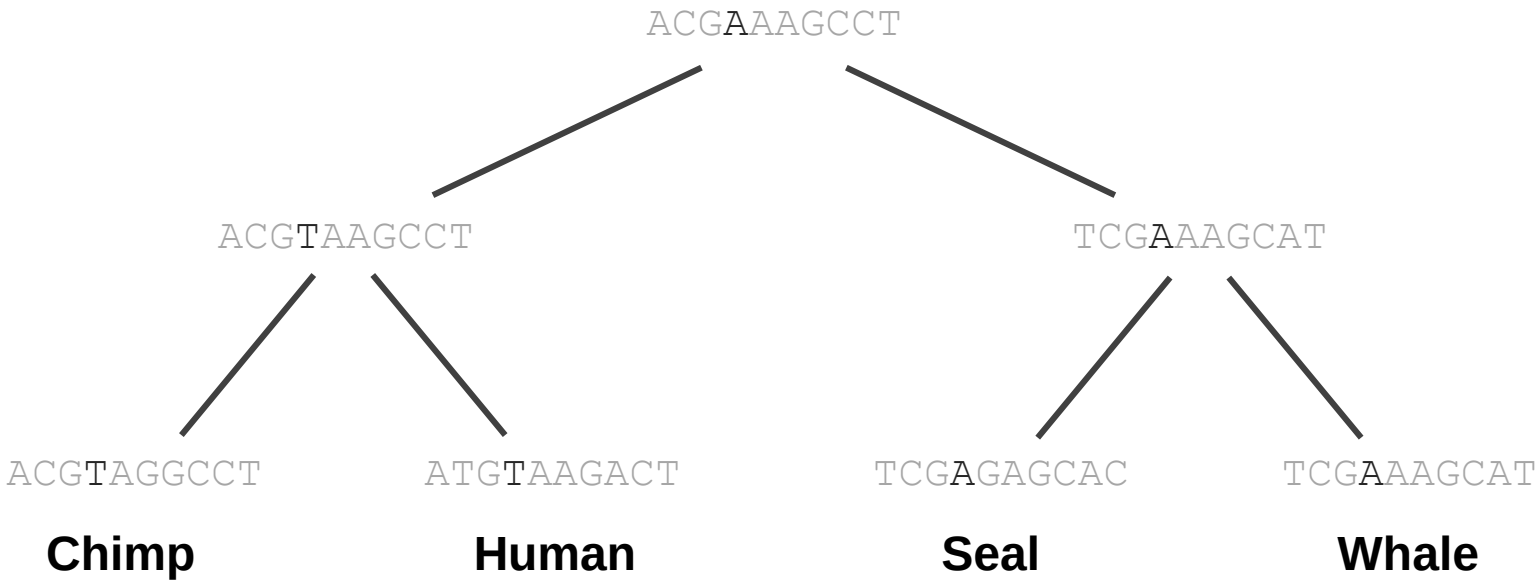
Problem male parsimonije: *Odrediti oznake za unutrašnje čvorove korenog stabla.*

- **Ulaz:** Koreno binarno stablo gde je svaki list označen niskom karaktera dužine m.
- **Izlaz:** Oznake za sve ostale čvorove stabla takve da minimizuju skor parsimonije stabla.

Da li ovu formulaciju možemo pojednostaviti?

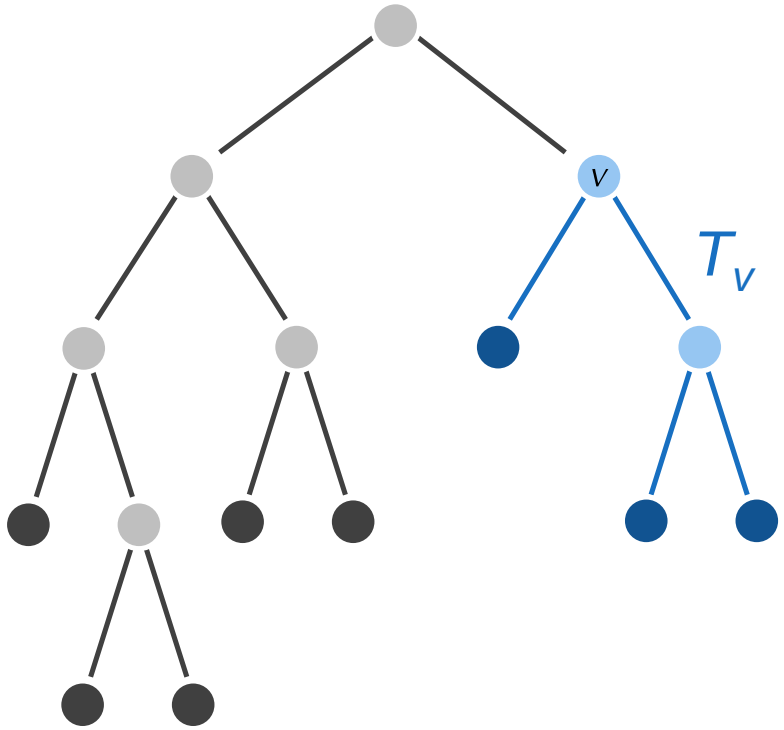
Problem male parsimonije: *Odrediti oznake za unutrašnje čvorove korenog stabla.*

- **Ulaz:** Koreno binarno stablo gde je svaki list označen jednim simbolom.
- **Izlaz:** Oznake za sve ostale čvorove stabla takve da minimizuju skor parsimonije stabla.


ACGAAAGCCT
ACGTAAGCCT
TCGAAAGCAT
ACGTAGGCCT
ATGTAAGACT
TCGAGAGCAC
TCGAAAGCAT
Chimp
Human
Seal
Whale

Algoritam dinamičkog programiranja

Neka je T_v podstablo stabla T sa korenom u čvoru v.



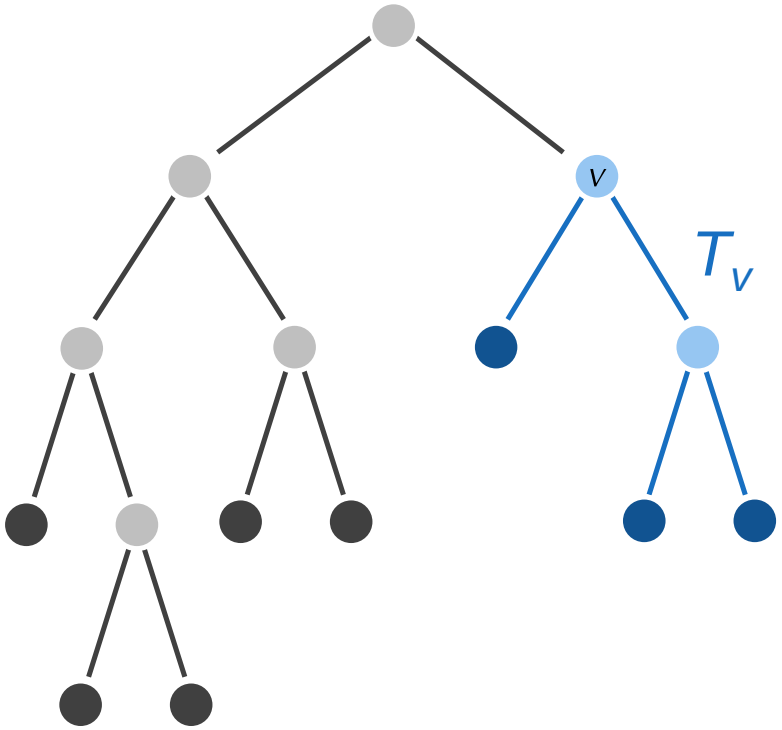
Neka je $s_k(v)$ minimalni skor parsimonije stabla T_v za sva moguća obeležavanja, pod pretpostavkom da je čvor v obeležen simbolom k.

Minimalni skor parsimonije stabla jednak je minimalnoj vrednosti $s_k(root)$ po svim simbolima k.

Algoritam dinamičkog programiranja

Neka je $\delta_{i,j}$ Kronekerov delta simbol:

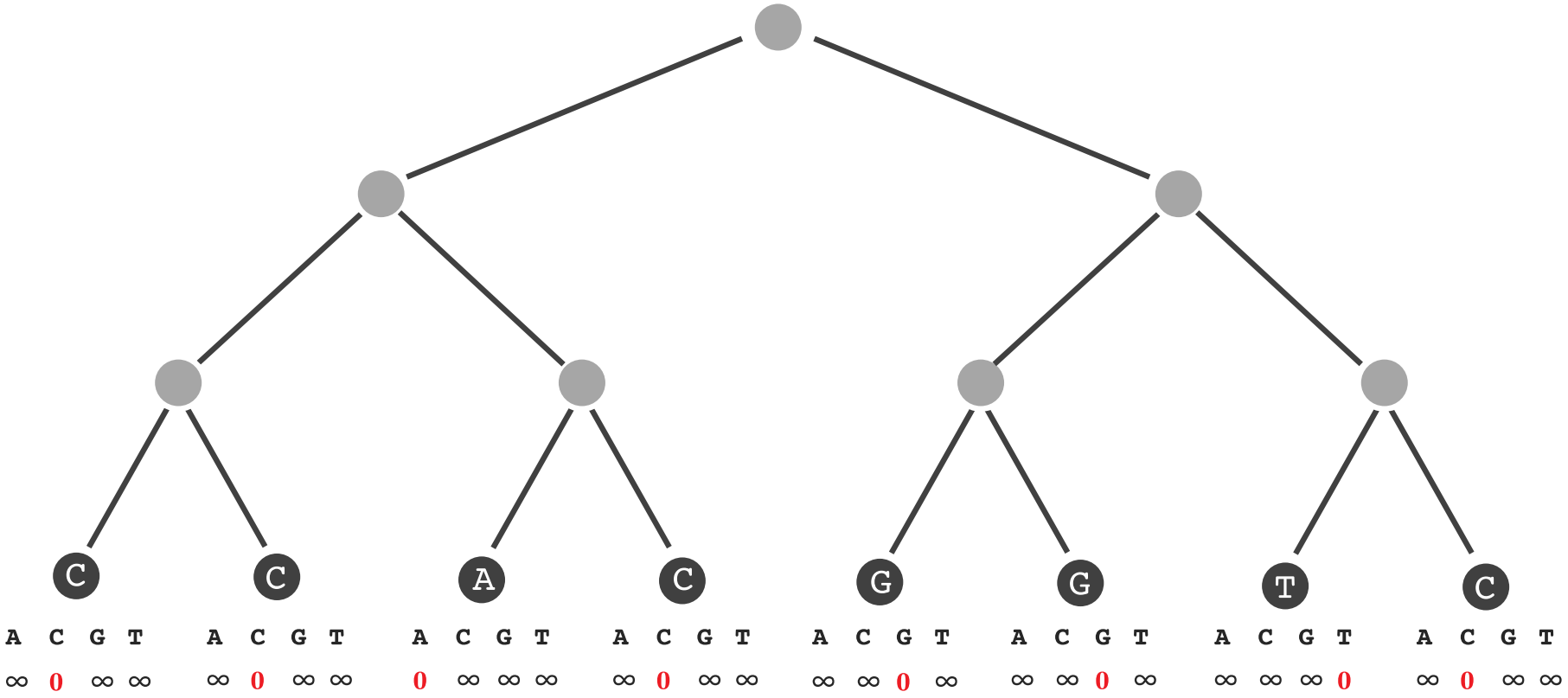
- $\delta_{i,j} = 0$ ako $i = j$
- $\delta_{i,j} = 1$ inače



Važi sledeća rekurentna relacija:

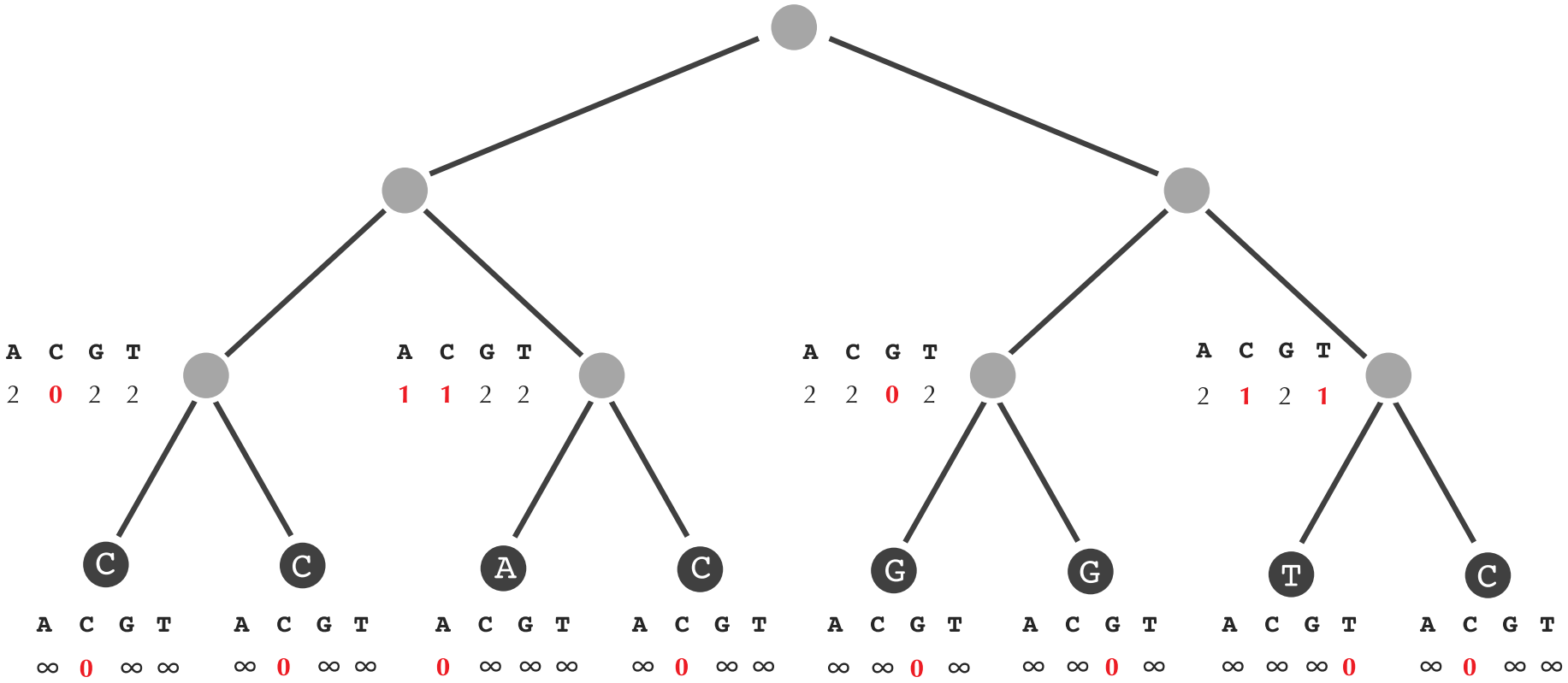
$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \delta_{i,k}\} + \min_{\text{all symbols } j} \{s_j(Son(v)) + \delta_{j,k}\}$$

Algoritam dinamičkog programiranja



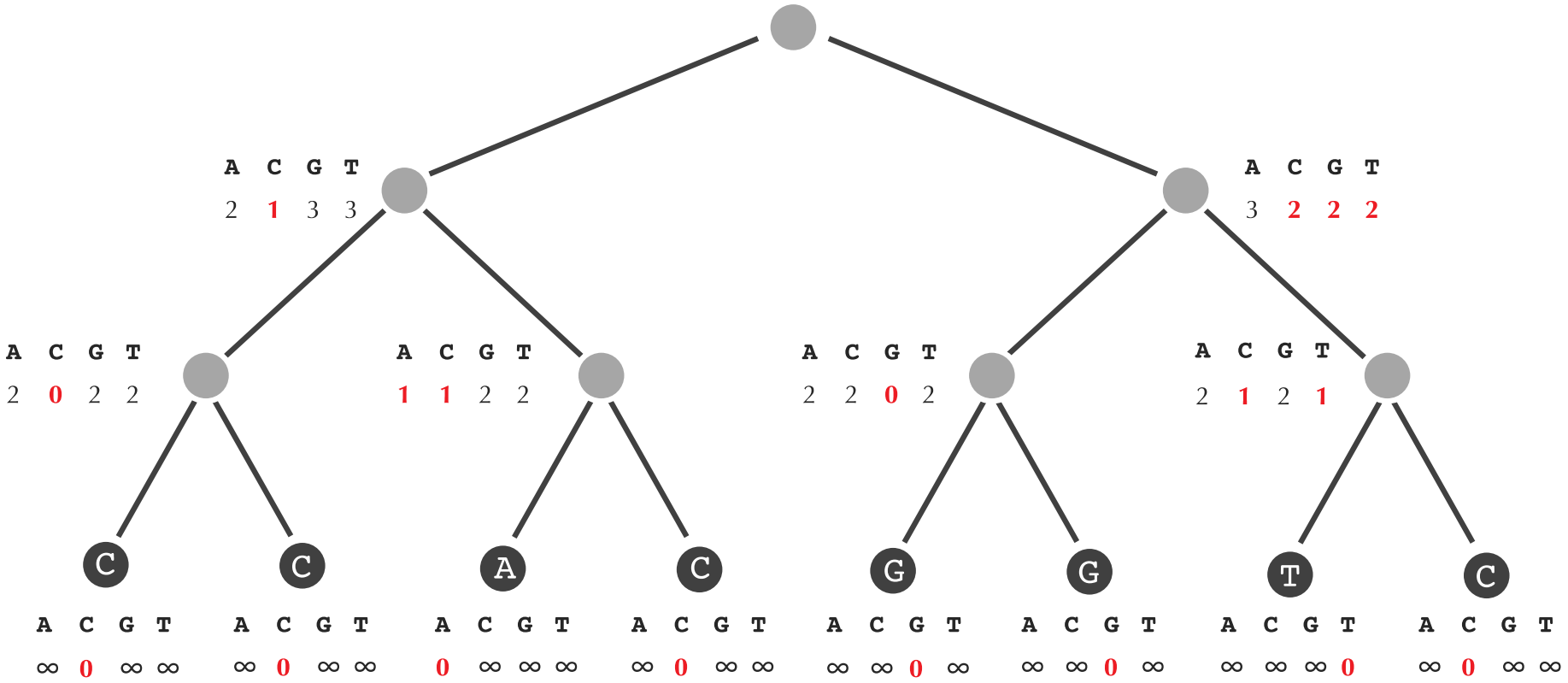
$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \delta_{i,k}\} + \min_{\text{all symbols } j} \{s_i(Son(v)) + \delta_{j,k}\}$$

Algoritam dinamičkog programiranja



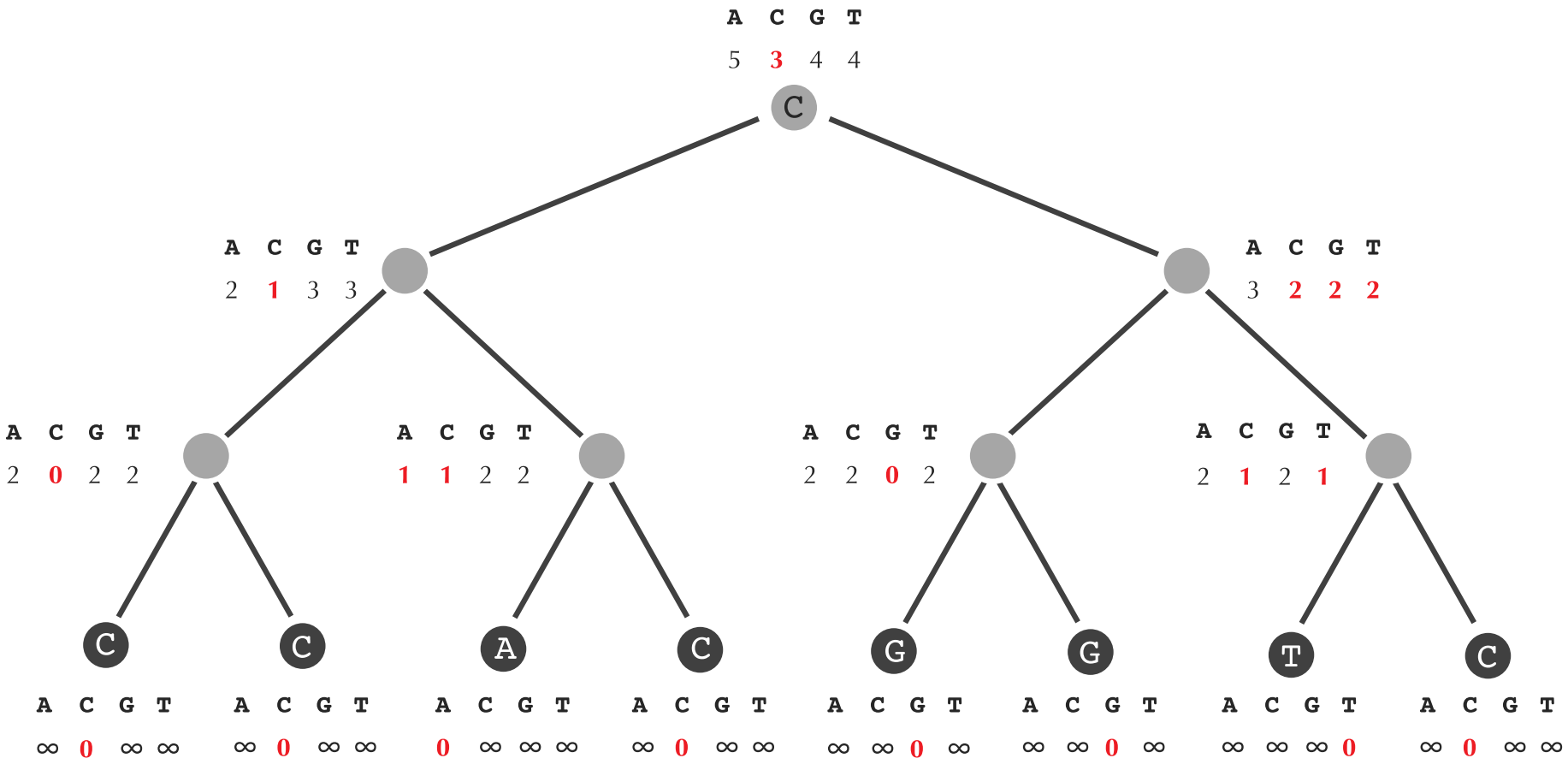
$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \delta_{i,k}\} + \min_{\text{all symbols } j} \{s_i(Son(v)) + \delta_{j,k}\}$$

Algoritam dinamičkog programiranja



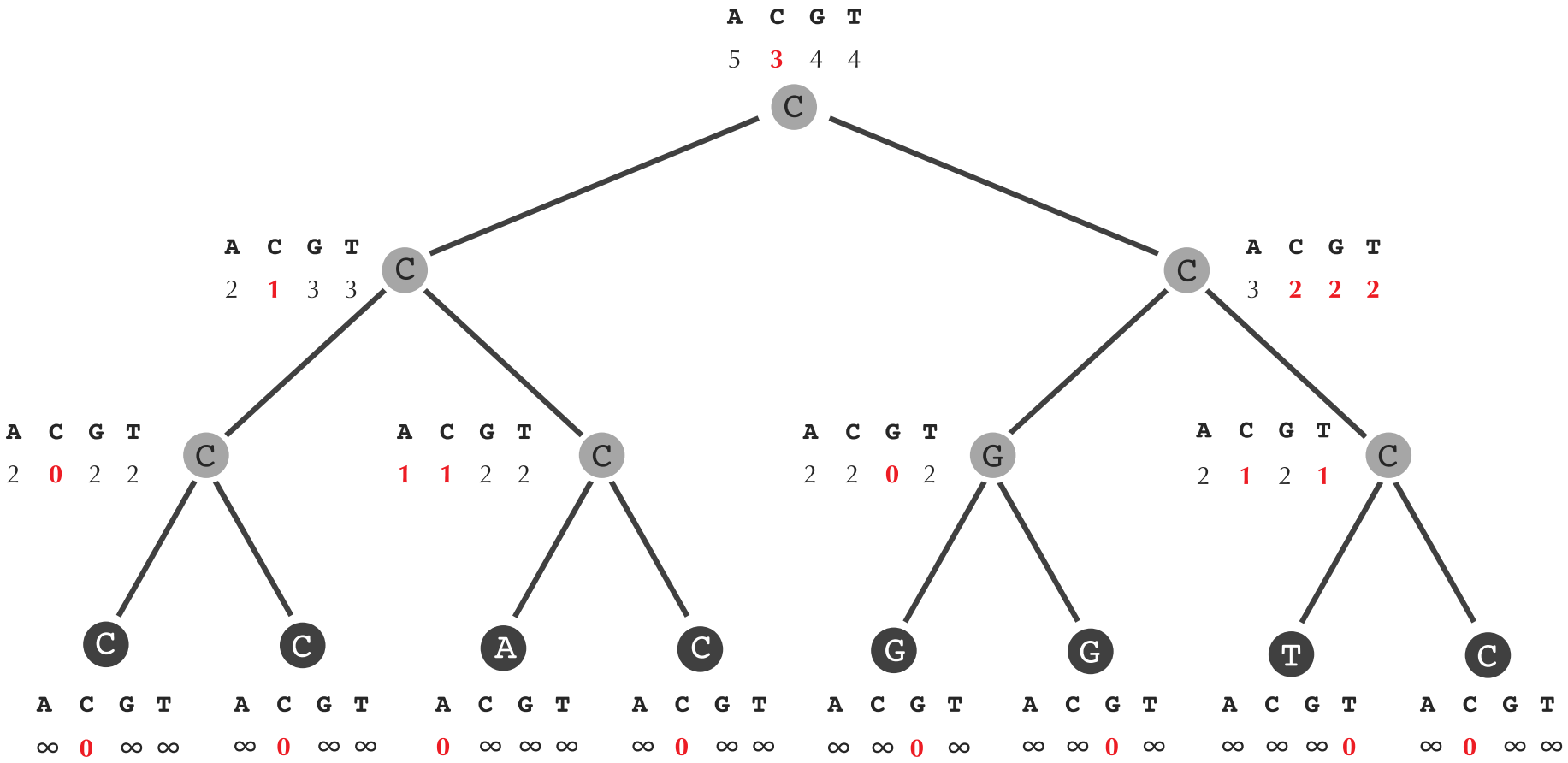
$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \delta_{i,k}\} + \min_{\text{all symbols } j} \{s_i(Son(v)) + \delta_{j,k}\}$$

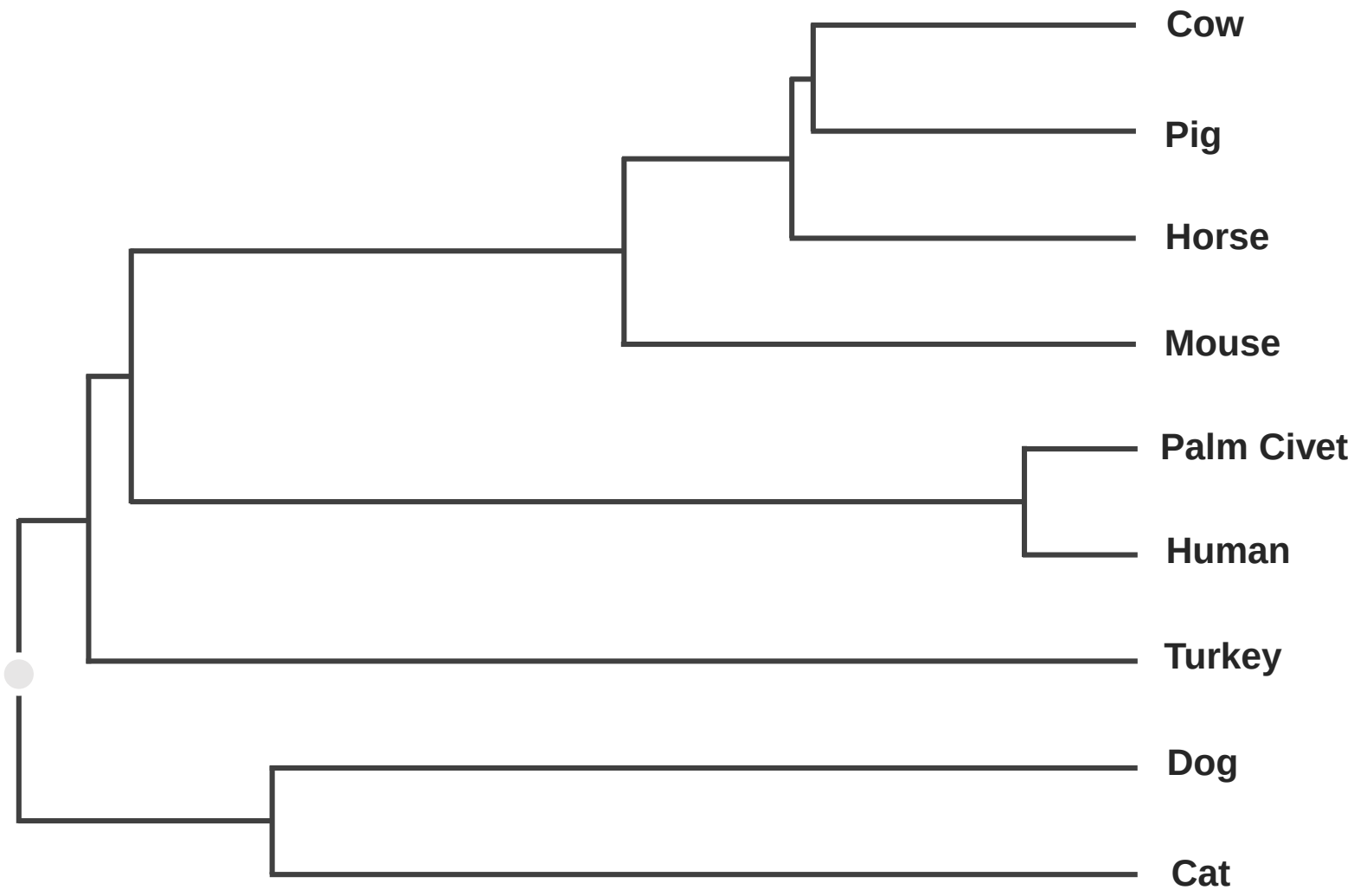
Algoritam dinamičkog programiranja



$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \delta_{i,k}\} + \min_{\text{all symbols } i} \{s_i(Son(v)) + \delta_{j,k}\}$$

Algoritam dinamičkog programiranja

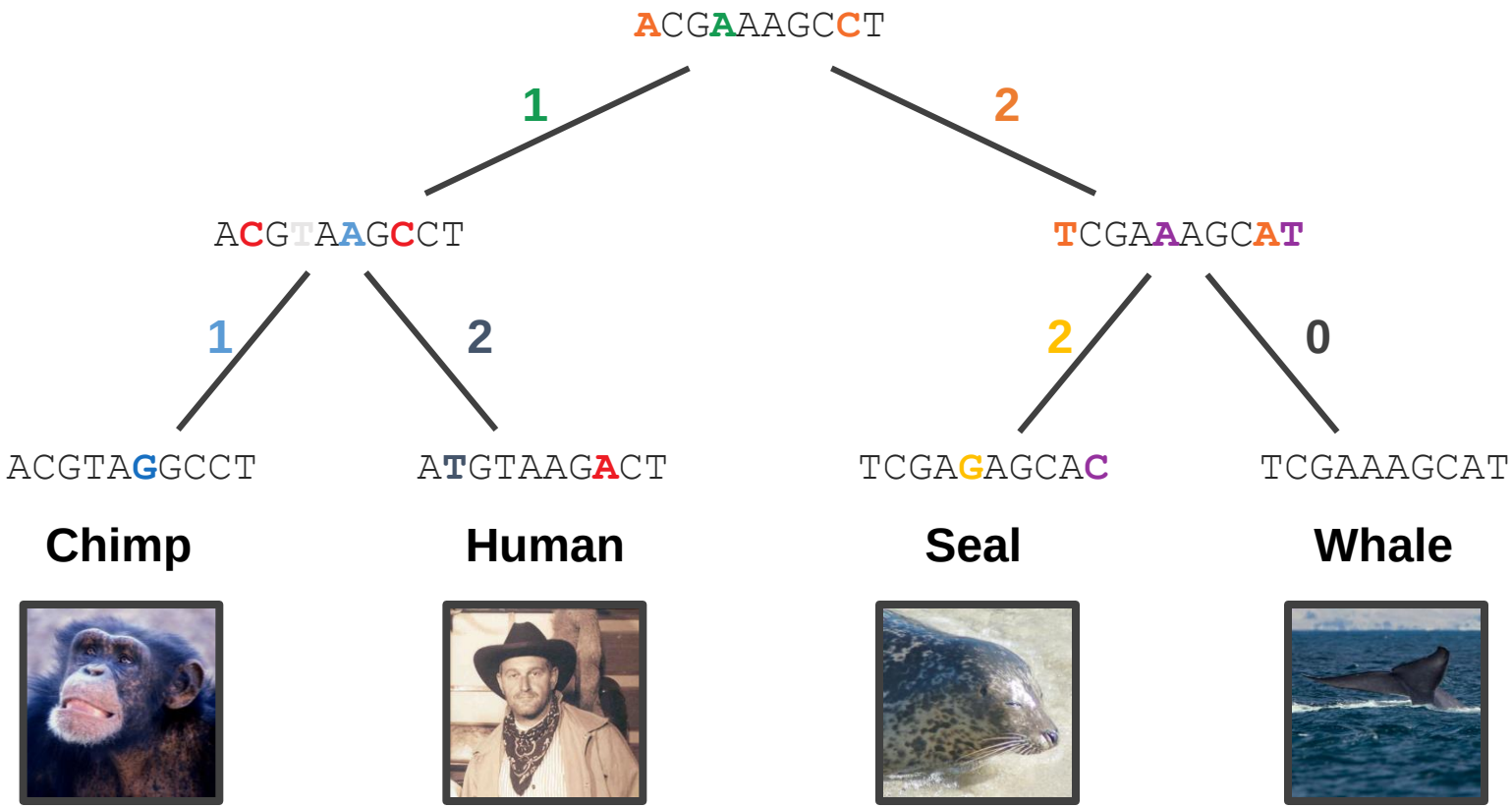



Cow
Pig
Horse
Mouse
Palm Civet
Human
Turkey
Dog
Cat

Pregled

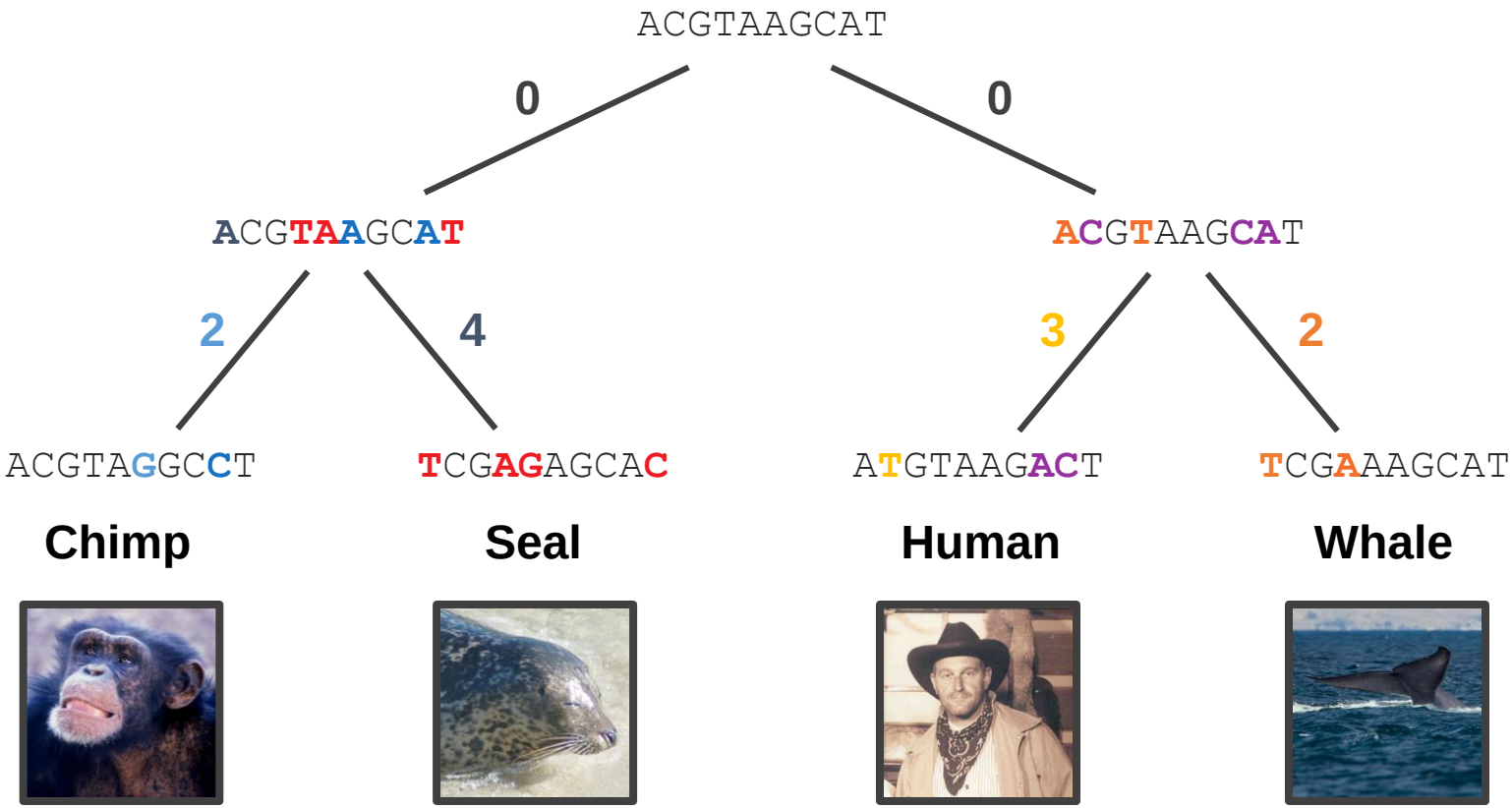
- Izbijanje epidemije
- Transformacija matrice rastojanja u evolutivno stablo
- Prema algoritmu za rekonstrukciju filogenetskog stabla na osnovu rastojanja
- *AdditivePhylogeny* algoritam
- Metod najmanjih kvadrata
- Ultrametrična evolutivna stabla
- *Neighbour-Joining* algoritam
- Rekonstrukcija stabla na osnovu karakteristika
- Problem male parsimonije
- **Problem velike parsimonije**

Traženje odgovarajućeg stabla



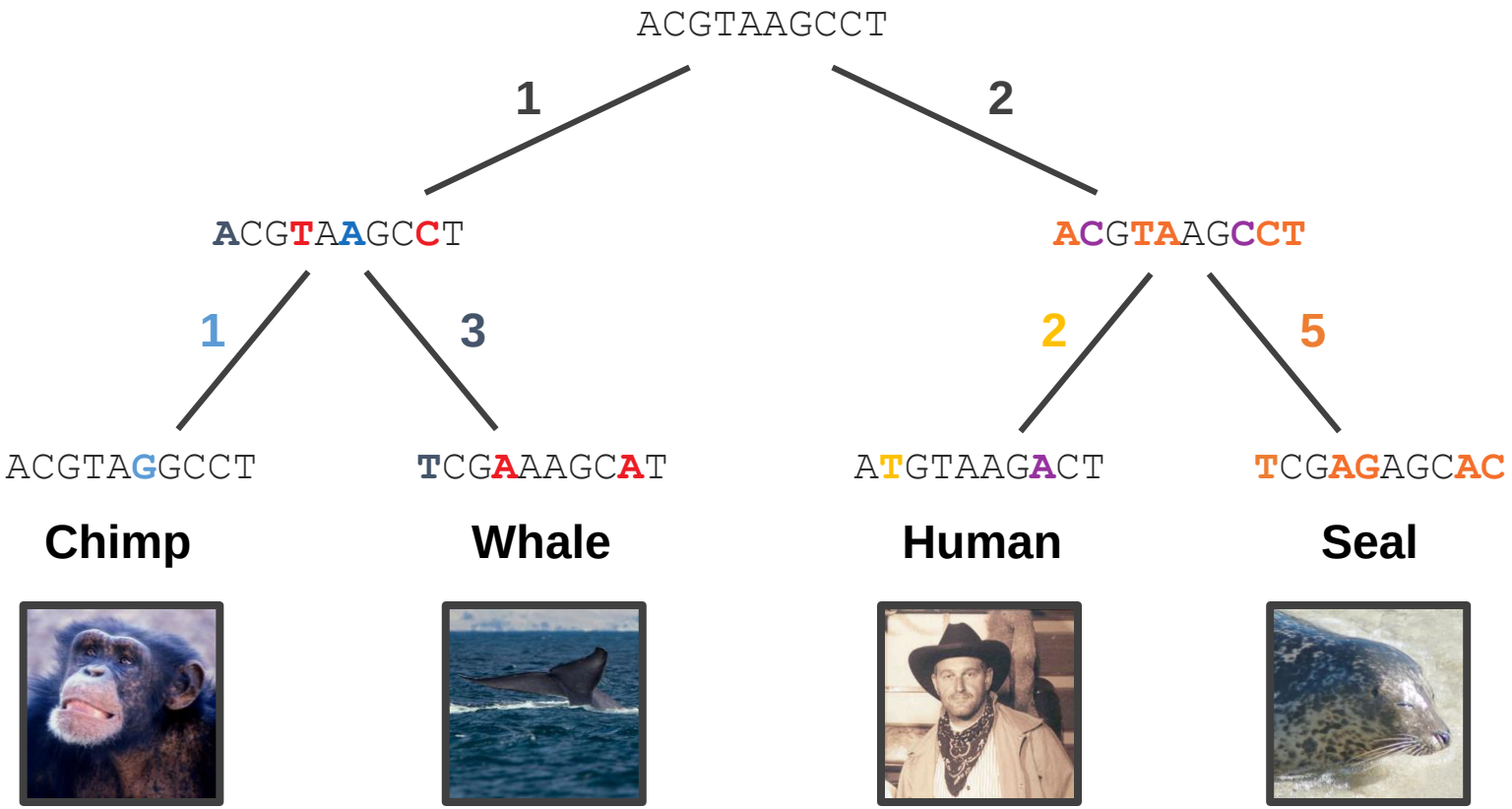
Skor parsimonije: 8

Traženje odgovarajućeg stabla



Skor parsimonije: 11

Traženje odgovarajućeg stabla



Skor parsimonije: 14

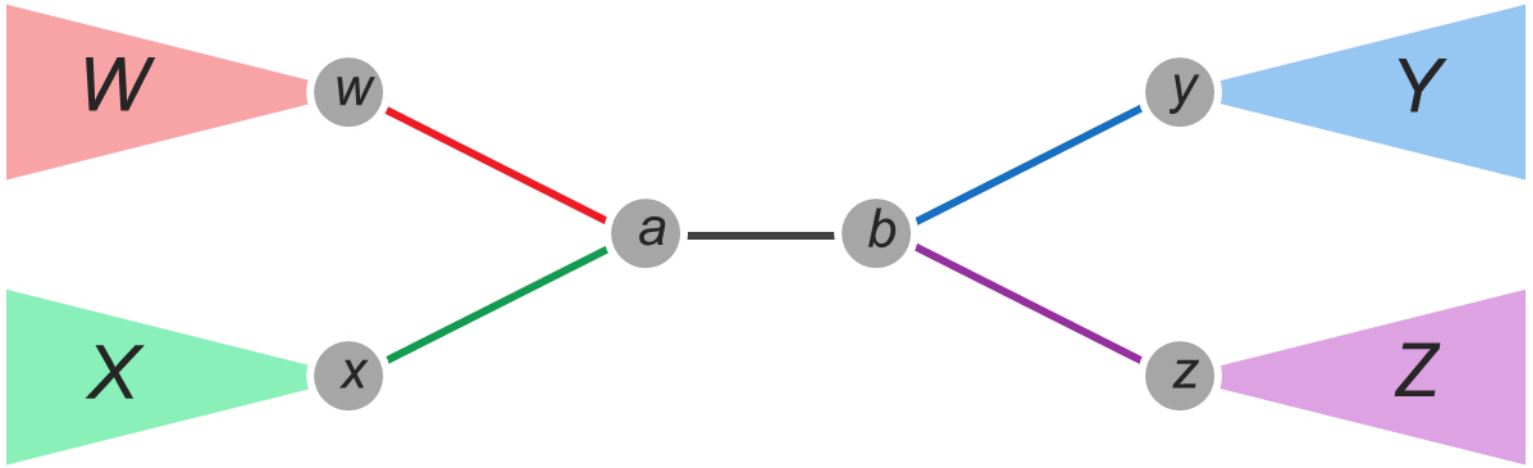
Problem velike parsimonije: *Za dati skup niski, naći stablo čiji su listovi označeni ovim niskama koje ima najmanji skor parsimonije.*

- **Ulaz:** Kolekcija niski jednake dužine.
- **Izlaz:** Koreno binarno stablo T koje minimizuje skor parsimonije po svim mogućim korenim binarnim stablima čiji su listovi označeni datim niskama.

Ovaj problem je NP-kompletan

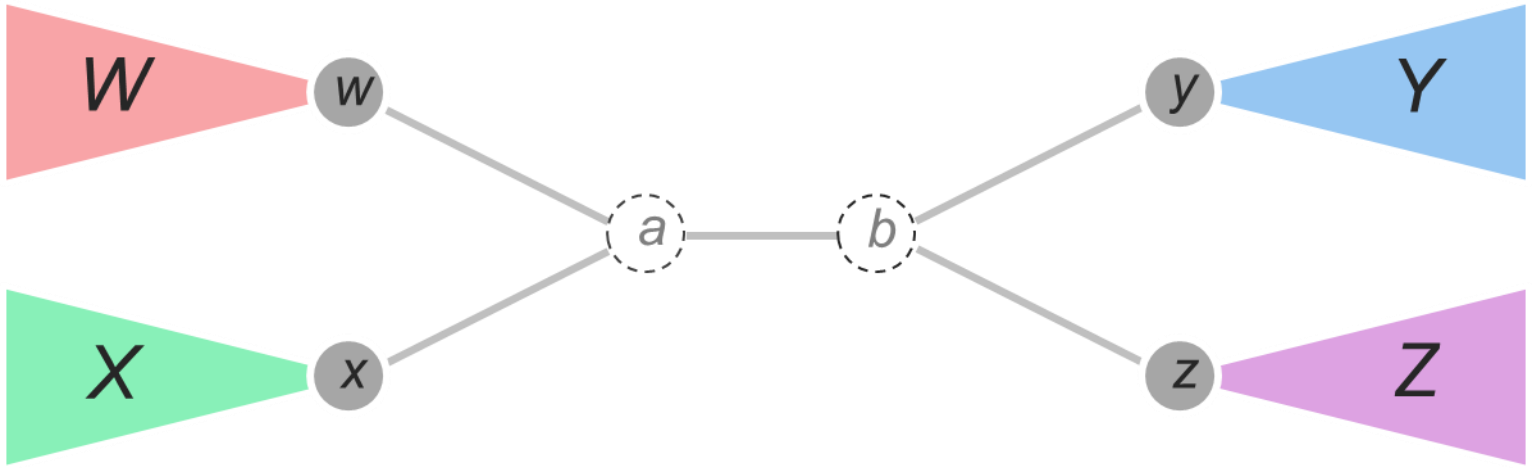
Pohlepna heuristika za veliku parsimoniju

Primetimo da uklanjanje jedne **unutrašnje grane**, grane koja povezuje dva unutrašnja čvora (zajedno sa čvorovima), dovodi do stvaranja četiri podstabla (*W*, *X*, *Y*, *Z*).



Pohlepna heuristika za veliku parsimoniju

Primetimo da uklanjanje jedne **unutrašnje grane**, grane koja povezuje dva unutrašnja čvora (zajedno sa čvorovima), dovodi do stvaranja četiri podstabla (*W*, *X*, *Y*, *Z*).

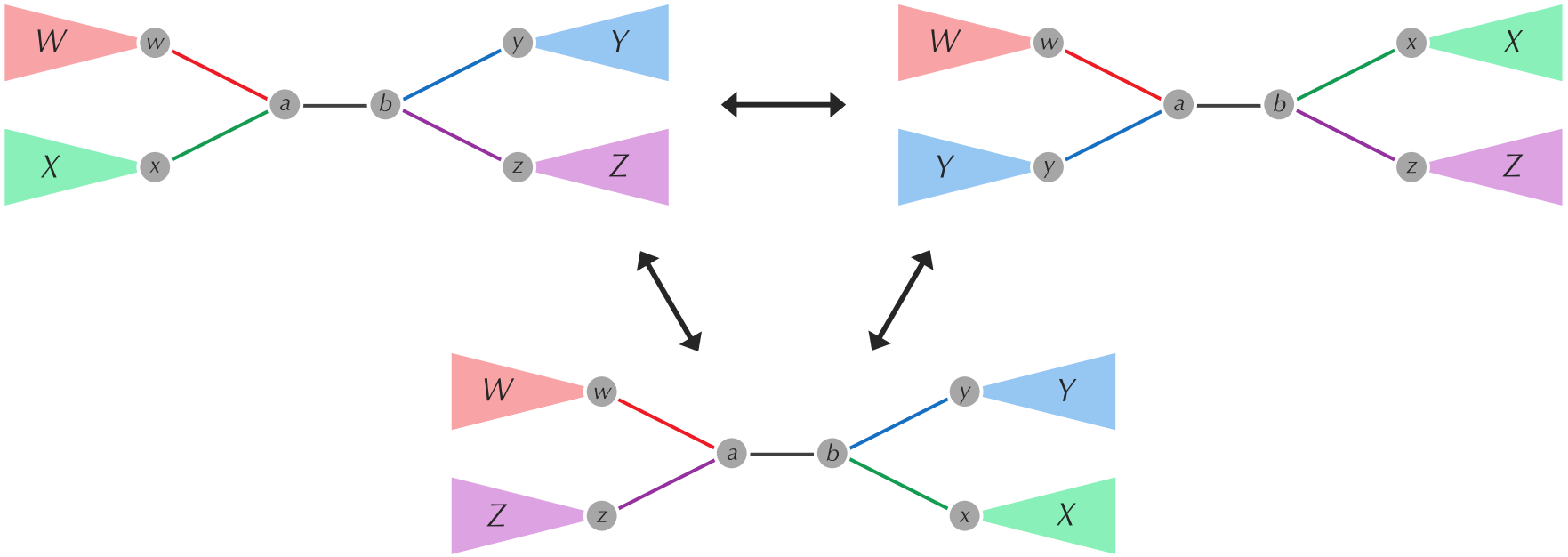


Pohlepna heuristika za veliku parsimoniju

Primetimo da uklanjanje jedne **unutrašnje grane**, grane koja povezuje dva unutrašnja čvora (zajedno sa čvorovima), dovodi do stvaranja četiri podstabla (*W*, *X*, *Y*, *Z*).

Pohlepna heuristika za veliku parsimoniju

Preuređenje rasporeda ovih podstabala se naziva **razmena najbližih suseda.**



Pohlepna heuristika za veliku parsimoniju

Problem najbližih suseda u stablu: *Za datu granu u binarnom stablu, generisati dva suseda ovog stabla.*

- **Ulaz:** Unutrašnja grana binarnog stabla.
- **Izlaz:** Dva najbliža suseda ovog stabla za datu unutrašnju granu.

Pohlepna heuristika za veliku parsimoniju

Heuristika za razmenu najbližih suseda:

1. Postaviti trenutno stablo na koreno binarno stablo proizvoljne strukture

Pohlepna heuristika za veliku parsimoniju

Heuristika za razmenu najbližih suseda:

1. Postaviti trenutno stablo na koreno binarno stablo proizvoljne strukture
2. Proći kroz sve unutrašnje grane i izvršiti sve moguće razmene najbližih suseda

Pohlepna heuristika za veliku parsimoniju

Heuristika za razmenu najbližih suseda:

1. Postaviti trenutno stablo na koreno binarno stablo proizvoljne strukture
2. Proći kroz sve unutrašnje grane i izvršiti sve moguće razmene najbližih suseda
3. Rešiti problem male parsimonije za svako takvo stablo

Pohlepna heuristika za veliku parsimoniju

Heuristika za razmenu najbližih suseda:

1. Postaviti trenutno stablo na koreno binarno stablo proizvoljne strukture
2. Proći kroz sve unutrašnje grane i izvršiti sve moguće razmene najbližih suseda
3. Rešiti problem male parsimonije za svako takvo stablo
4. Ako stablo ima skor parsimonije bolje od optimalnog stabla, postaviti da to bude trenutno stablo; inače, vratiti trenutno stablo

- Slajdovi pokrivaju poglavlje 7 knjige *Bioinformatics Algorithms: an Active Learning Approach*
- Sadržaj slajdova je preuzet sa zvaničnih prezentacija autora i dodatno prilagođen