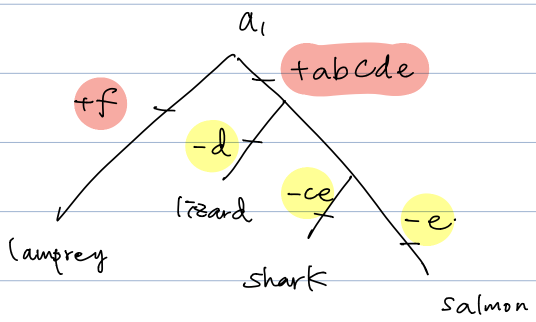
prob 2:

ⓐ



ⓑ

	a	b	c	d	e	f
lamprey	0	0	0	0	0	1
shark	1	1	0	1	0	0
salmon	1	1	1	1	0	0
lizard	1	1	1	0	1	0

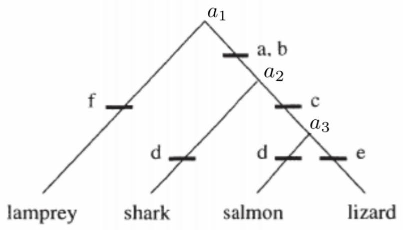


Figure 1: The data from Exercise 2

edges: there is no edge between two species, terminal nodes can only be connected via Steiner nodes. And each edge represents either developing or disappearing one or more genes.

weights: the weight of each edge is the length of longest common substring between two DNAs. the distance function c is the weight.

nodes: each node represents a kind of DNA combination,

① terminal nodes: each kind of species.

② steiner nodes: intersection nodes that connects two species, the DNA of steiner node is the longest common substring between two species it connects, and the rest gene will leave it 0.

① the longest common substring among each pair is 'salmon' and 'lizard'

② Connect salmon and lizard with a steiner nodes and the gene for that steiner nodes is 111000

③ the next longest common substring among remaining species is the previous steiner node with shark

④ Create a new steiner node to connect the previous steiner node and shark.

⑤ and the new steiner point has the gene 110000

⑥ Connect the lamprey with the newest steiner point with a steiner point that has gene 000000

⑦ return the Steiner Tree, and this tree is the most parsimonious tree

ⓒ We can treat the M as m DNAs (species) that has n characters in each DNA.

1, To construct a parisomonious tree, we first calculate the longest common substring for each pair (m DNAs).

2, And starting to construct the tree with the most common two genes.

3. Connect the two genes with a steiner node that has the gene the same as the longest common substring of the two genes.

4. and then choose the DNA genes that has the longest common substring with the previous steiner node

5. Connect the next gene and the steiner node with a new steiner node

6, the new steiner node has gene that is the longest common substring of the previous steiner node and the selected species.

7, Iteratively construct the tree until all kind of species are included.

prob 4 :

① construct a DP table with len(S) columns and len(T) rows

```
② if S[i] = T[j] :
       return 1 + Table [i-1, j-1]
   elif S[i] ≠ T[j]:
       return 0
```

③ ex. S = "abcxe" , T = "abce"

	a	b	c	x	e
a	1	0	0	0	0
b	0	2	0	0	0
c	0	0	3	0	0
e	0	0	0	0	0

④ find the largest element in the table, the largest number in the table is the length of longest common substring. And, if we follow the diagonal values starting from the largest number we can get the longest common substring.

⑤ the above example will return 3 as the length of longest common substring and the longest common substring is "abc".