9. Übung

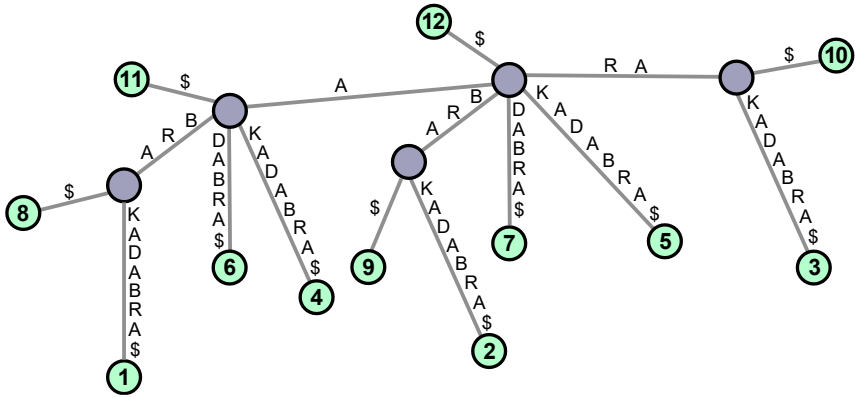
Einführung in die Bioinformatik I, 1. Teil
Wintersemester 2019/2020

Aufgabe 1 (5 Punkte):

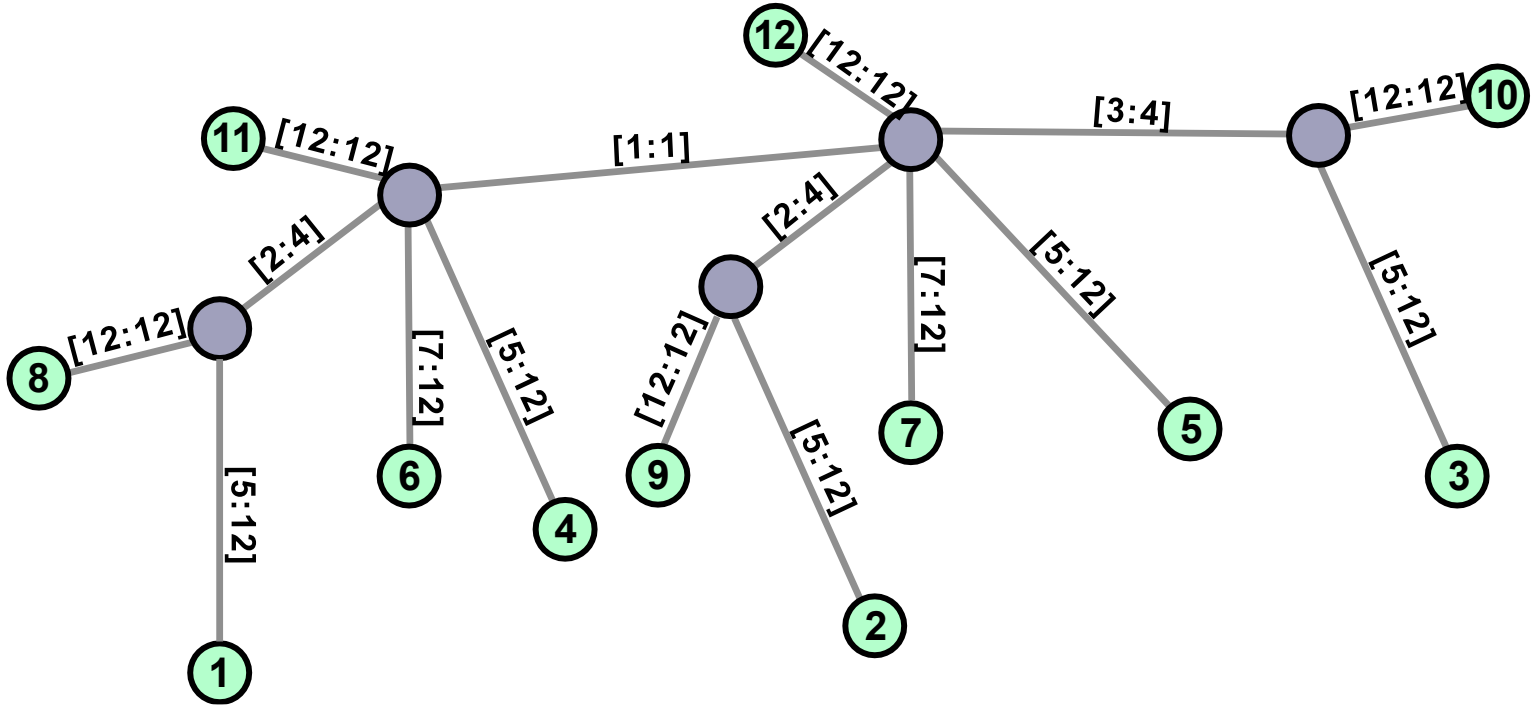
1. Bestimmen Sie für den String `abrakadabra` und `abaababab` den jeweiligen komprimierten Suffixbaum. Ist der komprimierte Suffixbaum nicht eindeutig, geben Sie bitte alle an.
   Hinweis: Es ist ausreichend, eine alternative Beschriftung in anderer Farbe anzubringen oder separat alternative Beschriftungen zu kennzeichnen.

Aufgabe 1 (5 Punkte):

1. Bestimmen Sie für den String `abrakadabra` und `abaababab` den jeweiligen komprimierten Suffixbaum. Ist der komprimierte Suffixbaum nicht eindeutig, geben Sie bitte alle an.
   Hinweis: Es ist ausreichend, eine alternative Beschriftung in anderer Farbe anzubringen oder separat alternative Beschriftungen zu kennzeichnen.



1	2	3	4	5	6	7	8	9	10	11	12
A	B	R	A	K	A	D	A	B	R	A	$



[1:1]=[4:4]=[6:6]=[8:8]=[11:11]

[3:4]=[10:11]

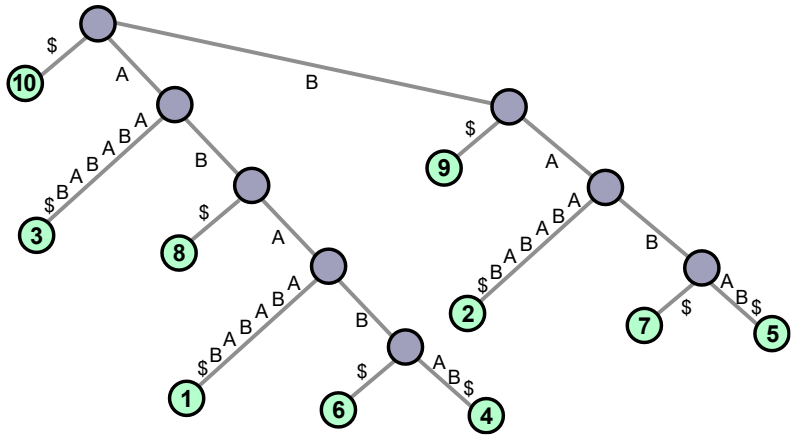
[2:4]=[9:11]

Aufgabe 1 (5 Punkte):

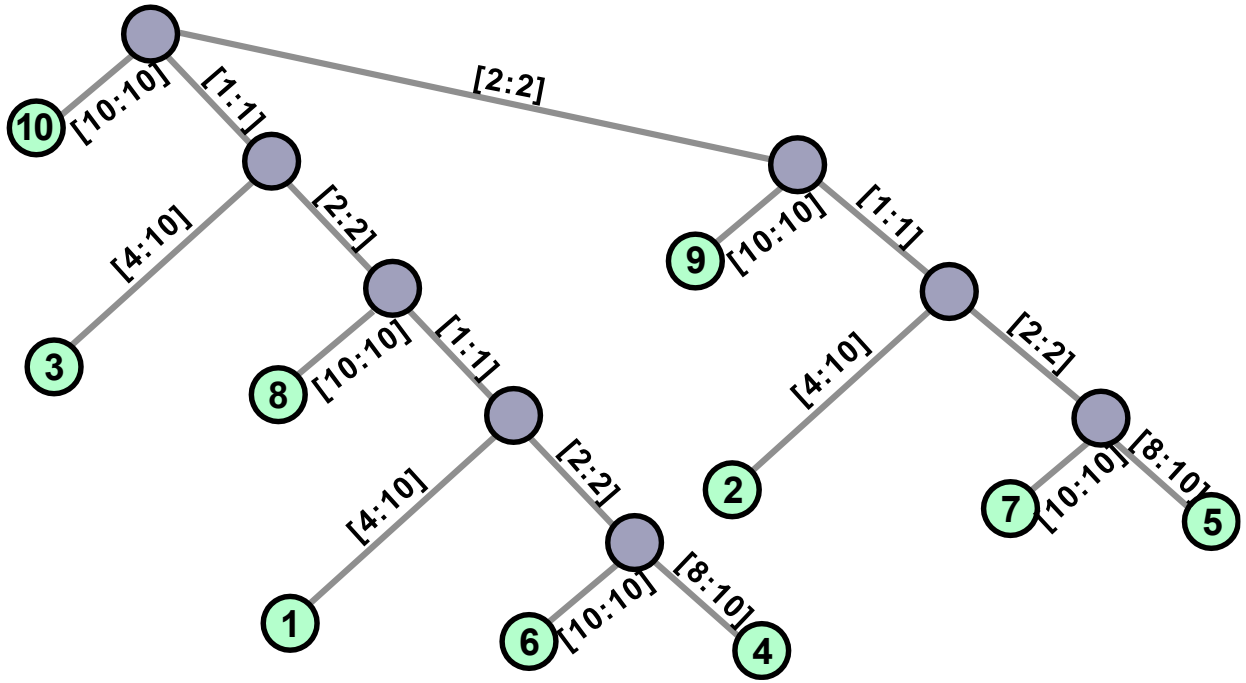
1. Bestimmen Sie für den String `abrakadabra` und `abaababab` den jeweiligen komprimierten Suffixbaum. Ist der komprimierte Suffixbaum nicht eindeutig, geben Sie bitte alle an.
   Hinweis: Es ist ausreichend, eine alternative Beschriftung in anderer Farbe anzubringen oder separat alternative Beschriftungen zu kennzeichnen.

Aufgabe 1 (5 Punkte):

1. Bestimmen Sie für den String `abrakadabra` und `abaababab` den jeweiligen komprimierten Suffixbaum. Ist der komprimierte Suffixbaum nicht eindeutig, geben Sie bitte alle an.
   Hinweis: Es ist ausreichend, eine alternative Beschriftung in anderer Farbe anzubringen oder separat alternative Beschriftungen zu kennzeichnen.



1	2	3	4	5	6	7	8	9	10
A	B	A	A	B	A	B	A	B	$



[1:1]=[3:3]=[4:4]=[6:6]=[8:8]
[2:2]=[5:5]=[7:7]=[9:9]

Aufgabe 1 (5 Punkte):

2. Bestimmen Sie für die Strings `AGTT`, `TAC` und `GTAG` den komprimierten generalisierten[1] Suffixbaum.

[1]Der *generalisierte* Suffixbaum für k Strings $S_1, \ldots, S_k$ ist der Suffixbaum für $S_1\$_1S_2\$_2 \ldots S_k\$_k$, wobei $\$_1, \$_2, \ldots, \$_k$ paarweise verschiedene Zeichen sind, die nicht im verwendeten Alphabet vorkommen.

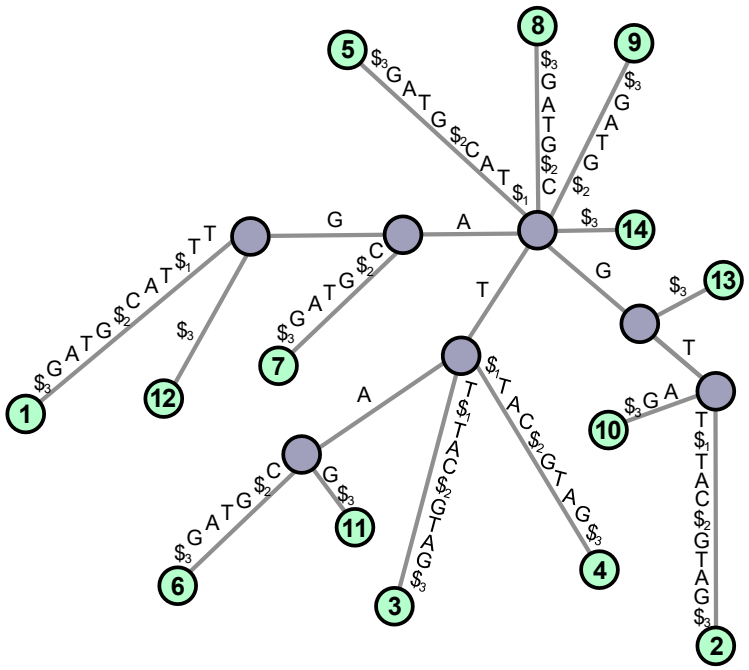
1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	G	T	T	$\$_1$	T	A	C	$\$_2$	G	T	A	G	$\$_3$

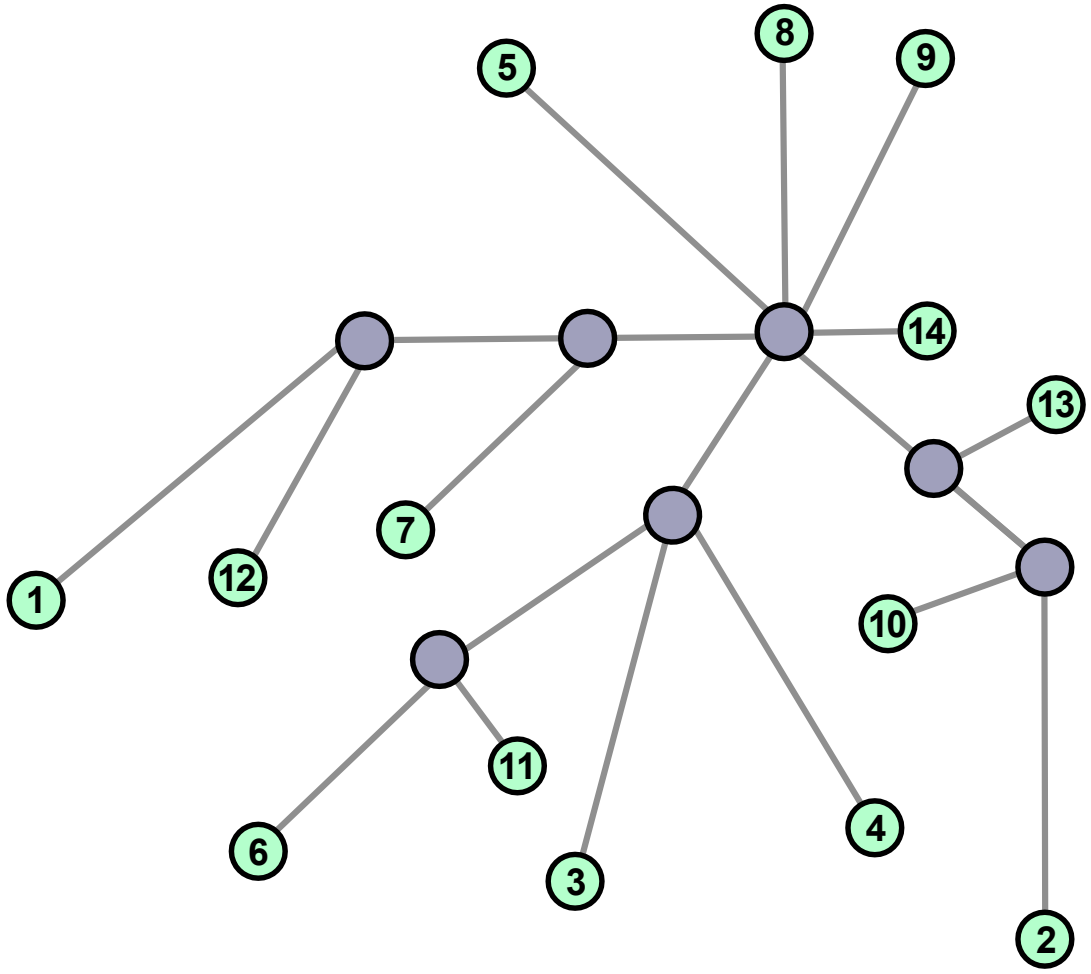
Aufgabe 1 (5 Punkte):

2. Bestimmen Sie für die Strings `AGTT`, `TAC` und `GTAG` den komprimierten generalisierten[1] Suffixbaum.

[1]Der *generalisierte* Suffixbaum für k Strings $S_1, \ldots, S_k$ ist der Suffixbaum für $S_1\$_1S_2\$_2 \ldots S_k\$_k$, wobei $\$_1, \$_2, \ldots, \$_k$ paarweise verschiedene Zeichen sind, die nicht im verwendeten Alphabet vorkommen.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	G	T	T	$\$_1$	T	A	C	$\$_2$	G	T	A	G	$\$_3$



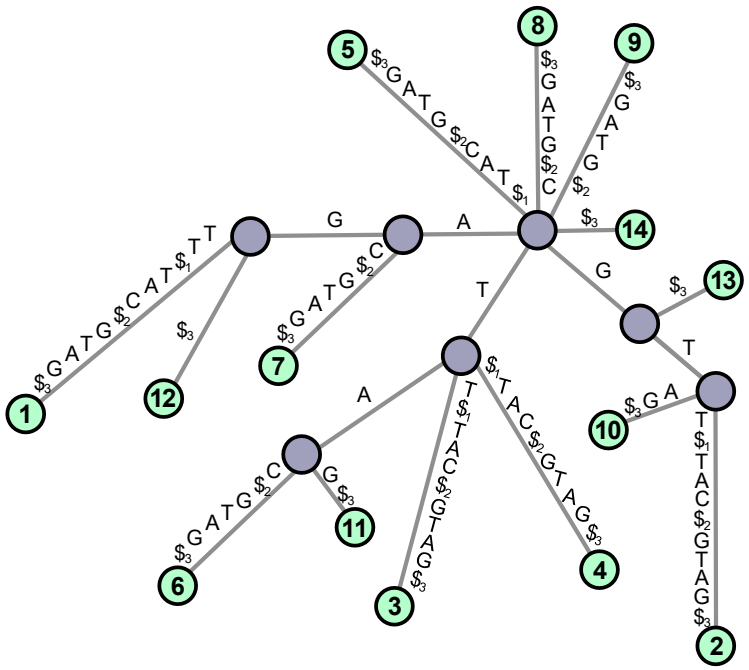


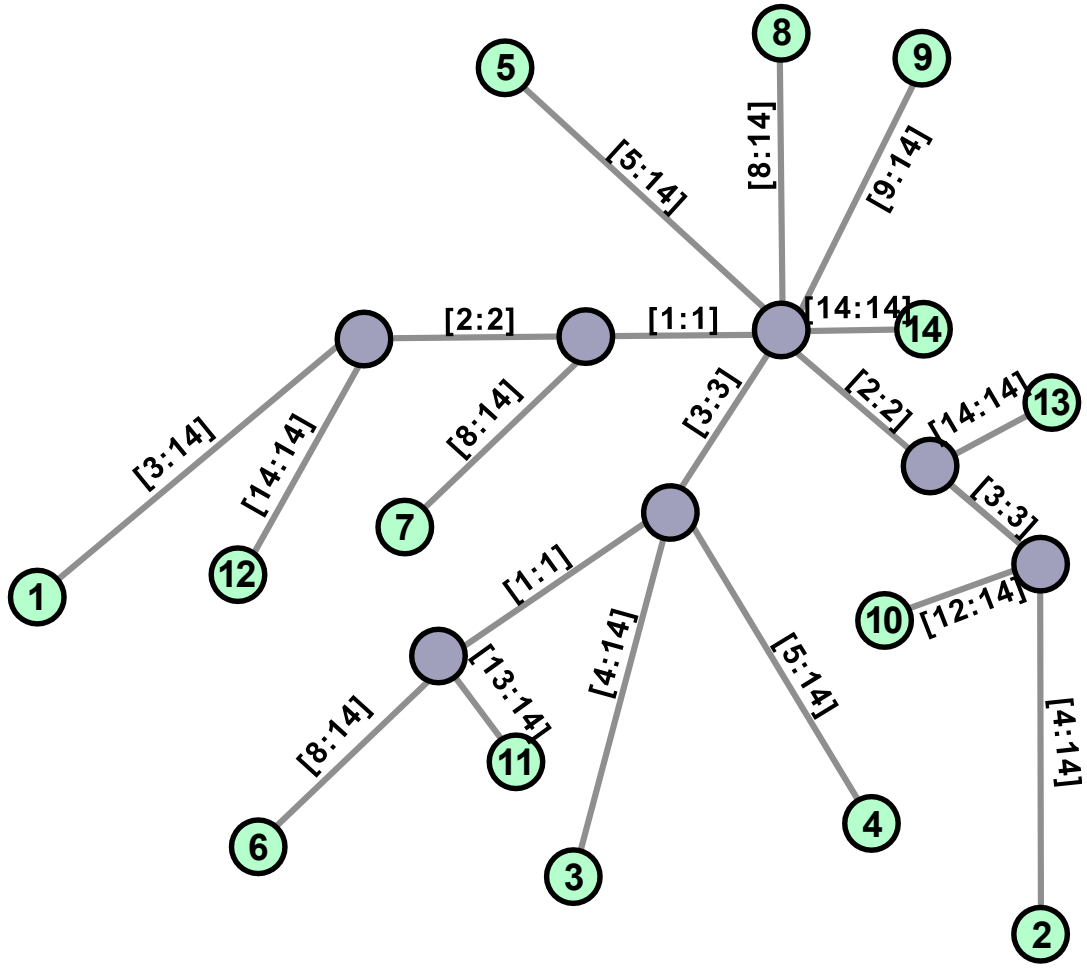
Aufgabe 1 (5 Punkte):

2. Bestimmen Sie für die Strings `AGTT`, `TAC` und `GTAG` den komprimierten generalisierten[1] Suffixbaum.

[1]Der *generalisierte* Suffixbaum für k Strings $S_1, \ldots, S_k$ ist der Suffixbaum für $S_1\$_1S_2\$_2 \ldots S_k\$_k$, wobei $\$_1, \$_2, \ldots, \$_k$ paarweise verschiedene Zeichen sind, die nicht im verwendeten Alphabet vorkommen.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	G	T	T	$\$_1$	T	A	C	$\$_2$	G	T	A	G	$\$_3$





[1:1]=[7:7]=[12:12]
[2:2]=[10:10]=[13:13]
[3:3]=[4:4]=[6:6]=[11:11]

Aufgabe 2 (5 Punkte): Wie kann man mit Hilfe eines Suffixbaumes in einem Text der Länge n den längsten Teilstring finden, der in diesem Text auch rückwärts vorkommt? In welcher asymptotischen Laufzeit ist dies möglich?

Aufgabe 2 (5 Punkte): Wie kann man mit Hilfe eines Suffixbaumes in einem Text der Länge n den längsten Teilstring finden, der in diesem Text auch rückwärts vorkommt? In welcher asymptotischen Laufzeit ist dies möglich?

<u>Algorithmus und Laufzeit:</u>

1. Erzeuge generalisierten Suffixbaum für den Text T und den reversen Text T^R, also einen Suffixbaum B für $T\$_1T^R\$_2$.
   → Laufzeit $O(n)$

2. Berechne für jeden Knoten v die Stringtiefe $s(v)$.
   → Laufzeit $O(n)$

3. Markiere innere Knoten, die für Teilstrings aus beiden Texten stehen.
   → Laufzeit $O(n)$

4. Suche nach markierten Knoten v mit der größten Stringtiefe $s(v)$.
   → Laufzeit $O(n)$