Homework 5

Elliott Pryor

2 December 2021

Problem 1 Problem 1 (15 pts) 551 presentation questions: Answer any three of the following questions (551 presentation slides are posted on google drive)

[Adiesha/Reese] linear time suffix array construction:

Find the type of each character in the given string mountainsandmind$. Find the LMS character positions using the type and then find the LMS substrings.

[Jordan/Andrew] Knuth-Morris-Pratt pattern matching:

Saving information about _____ allows the KMP algorithm to avoid needless repetition and thus reduce the time complexity from quadratic (for the naive algorithm) to linear.

[Joshua Ryan/Susan] Farach's suffix tree construction:

Follow Farach's algorithm on the string S = acagaca$. Create the even tree, odd tree, and then the merged tree. For an extra challenge, circle the overmerged nodes.

[Ian/Elliott] RNA sequencing with network flows:

When using a flow network to determine RNA sequences what does a node in the flow network represent?

a) a connection between two segments of RNA

b) a segment of RNA

c) the DNA that an RNA segment was transcribed from

d) the protein that gets translated from an RNA segment

---

(1) Adiesha / Reese

```
      00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
S =    M  O  U  N  T  A  I  N  S  A  N  D  M  I  N  D  $
  =    S  S  L  S  L  S  S  S  L  S  L  S  L  S  L  L  S
       *        *     *           *     *     *        *
```

Substrings:

```
00 - SSLS
03 - SLS
05 - SSSLS
09 - SLS
```

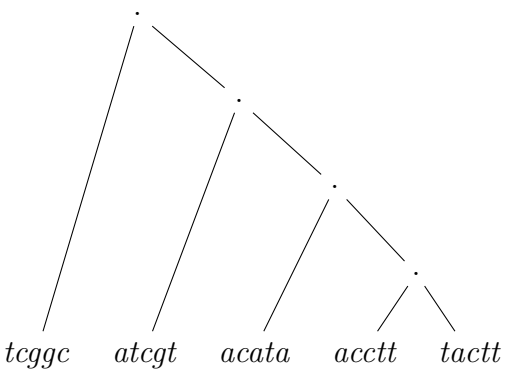
```
11 - SLS
13 - SLLS$
16 - $
```

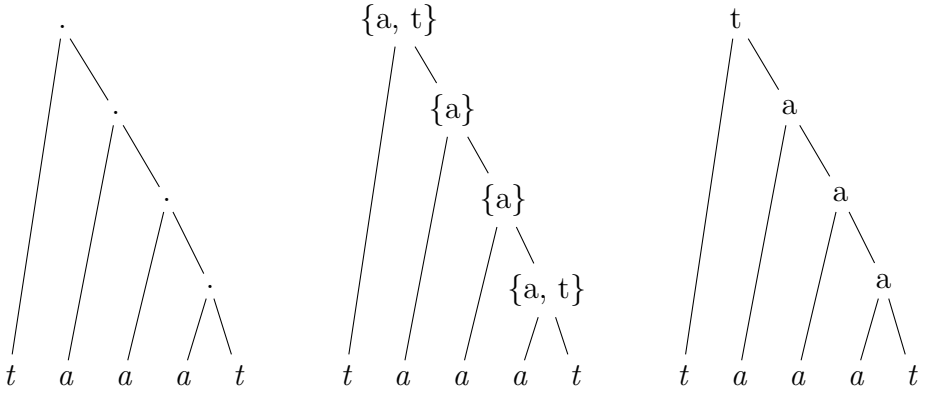
(2) Jordan / Andrew: the query string. This is the Q array from class. It stores the length of the biggest prefix at i in query string. It means we don't have to start over after matching, we can skip some.

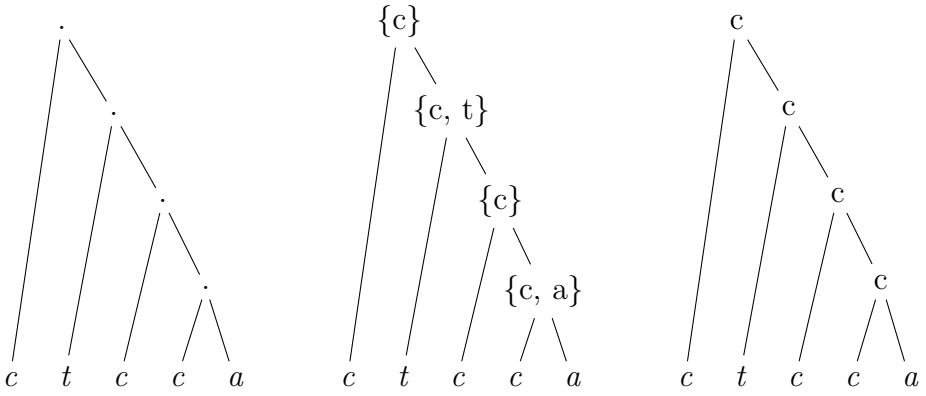
(3) Elliott / Ian: B

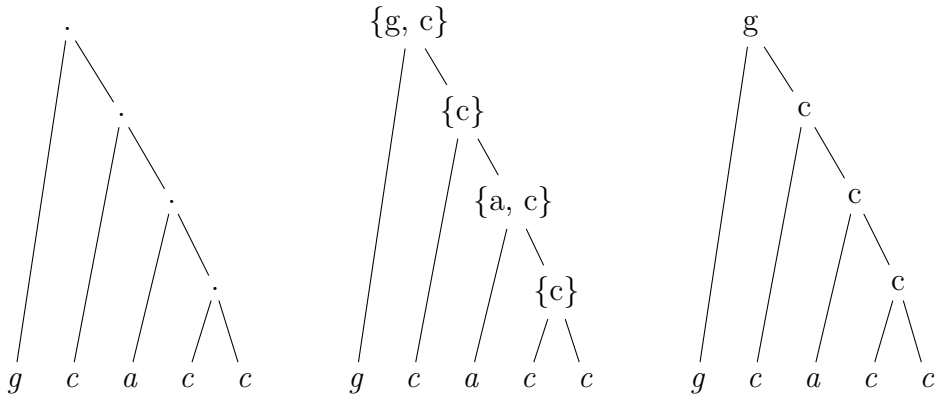
Problem 2 Fitch's algorithm: For the following five sequences, first build a binary tree where the sequences are the leaves (you can just guess the tree topology that you think leads to a good score). Then apply Fitch's algorithm to predict the ancestral sequences at each internal node in the tree.

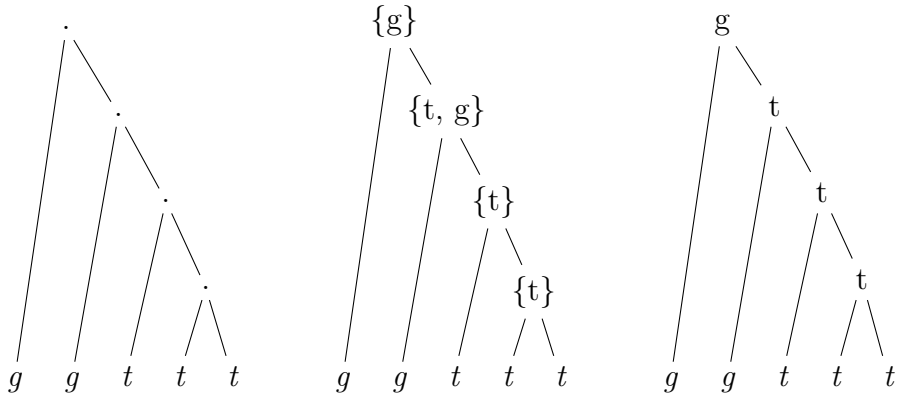
```
S1 = acctt  S2 = tcggc  S3 = tactt  S4 = atcgt  S5 = acata
```

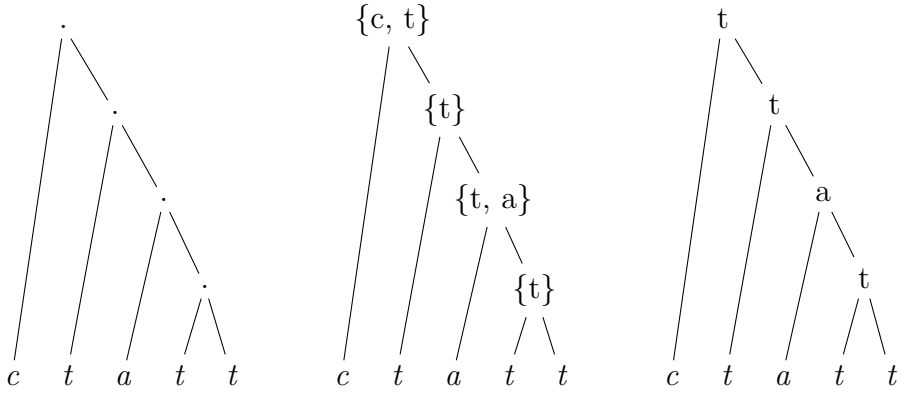


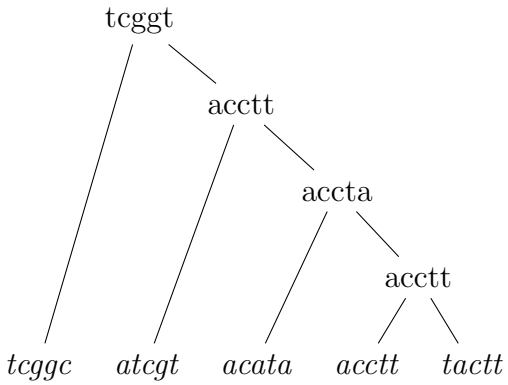





. . . . .
g c a c c
{g, c} {c} {a, c} {c}
g c a c c
g c c c c
g c a c c


. . . . .
g g t t t
{g} {t, g} {t} {t}
g g t t t
g t t t
g g t t t


. . . . .
c t a t t
{c, t} {t} {t, a} {t}
c t a t t
t t a t
c t a t t


tcggt
acctt
accta
acctt
tcggc
atcgt
acata
acctt
tactt