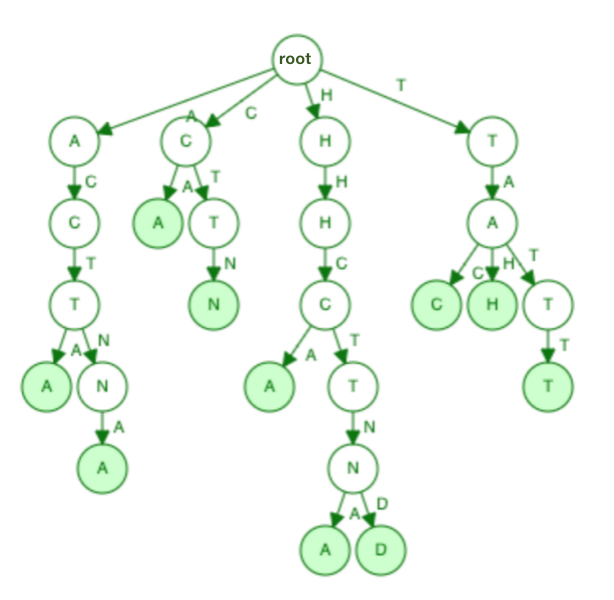
**Problem 1. The bus.**

Sol: As we can see from the Trie, it has 10 leaves.



**Problem 2. Bus stops**

Text := **tactnahhctndhhctna**

**First iteration:**

Input letter t, go through Trie with node t, then keep going we found pattern **tac.**

**Second iteration:**

Input letter a, go through Trie with node a, then sweep through, we found pattern **actna.**

**Third iteration:**

Input letter c, go through Trie with node c, then sweep through, we found pattern **ctn.**

**Fourth iteration:**

Input letter t, go through Trie with node t, then sweep through, we couldn’t find any pattern.

**Fifth iteration:**

Input letter n, go through Trie, no node n are found.

**Sixth iteration:**

Input letter a, go through Trie with node a, then sweep through, we couldn’t find any pattern.

**Seventh iteration:**

Input letter h, go through Trie with node h, then sweep through, we found pattern **hhctnd.**

**Eighth iteration:**

Input letter h, go through Trie with node h, then sweep through, we couldn’t find any pattern.

**Ninth iteration:**

Input letter c, go through Trie with node c, then sweep through, we found pattern **ctn.**

**Tenth iteration:**

Input letter t, go through Trie with node t, then sweep through, we couldn’t find any pattern.

**Eleventh iteration:**

Input letter n, no node n are found.

**Twelfth iteration:**

Input letter d, no node d are found.

**Thirteenth iteration:**

Input letter h, go through Trie with node h, then sweep through, we found pattern **hhctna.**

**Fourteenth iteration:**

Input letter h, go through Trie with node h, then sweep through, we couldn’t find any pattern.

**Fifteenth iteration:**

Input letter c, go through Trie with node c, then sweep through, we found pattern **ctn.**

**Sixteenth iteration:**

Input letter t, go through Trie with node h, then sweep through, we couldn’t find any pattern.

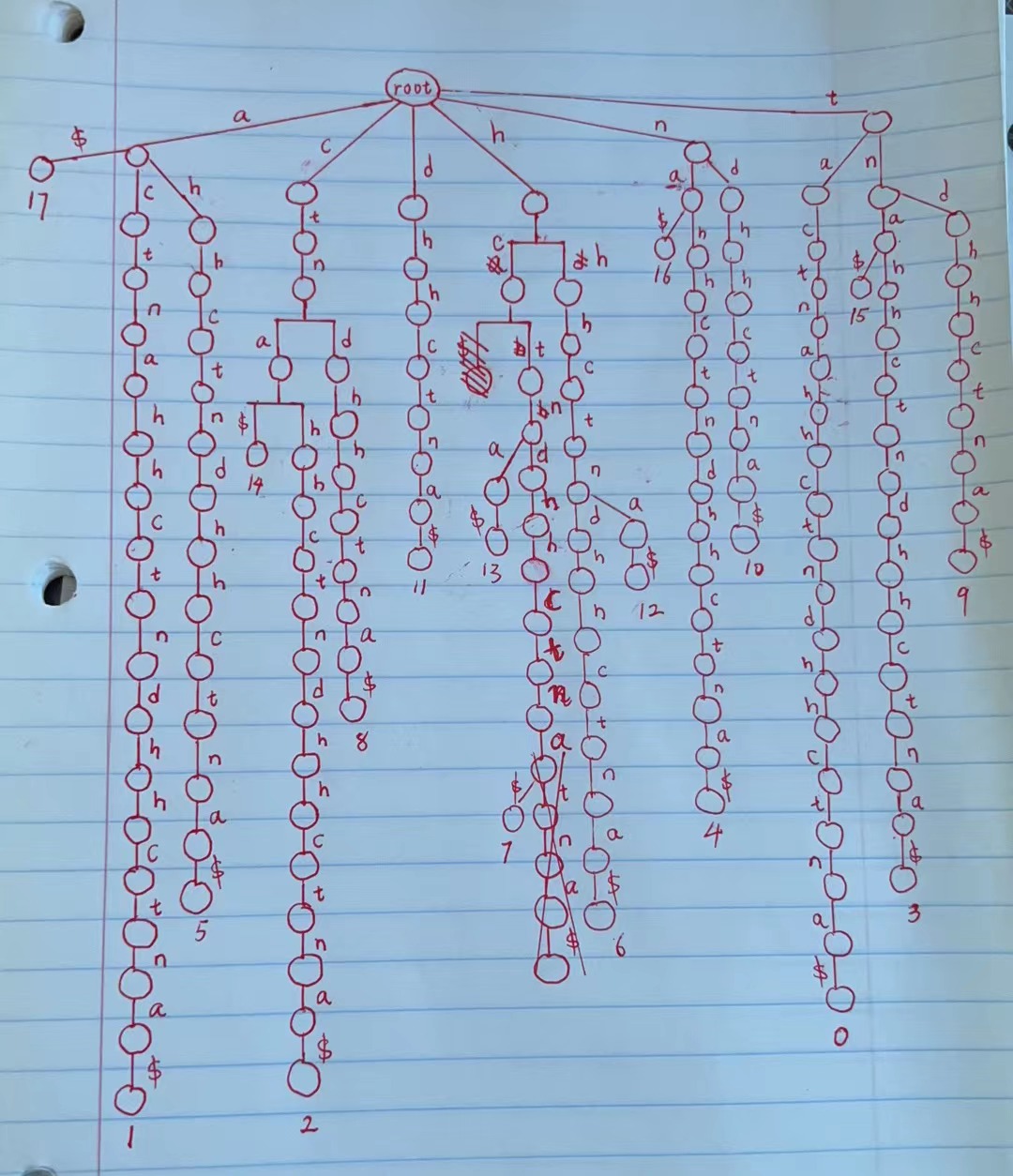
**Seventeenth iteration:**

Input letter n, no node n are found.

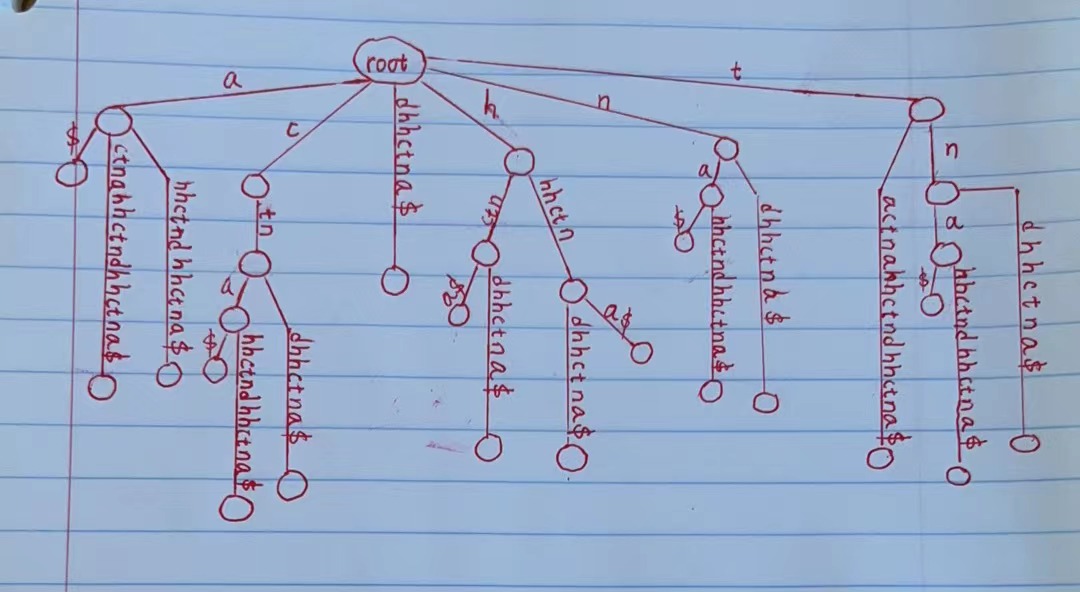
**Eighteenth iteration:**

Input letter a, go through Trie with node a, then sweep through, we couldn’t find any pattern.

In total of **7** matches were found by running the Triematch algorithm.

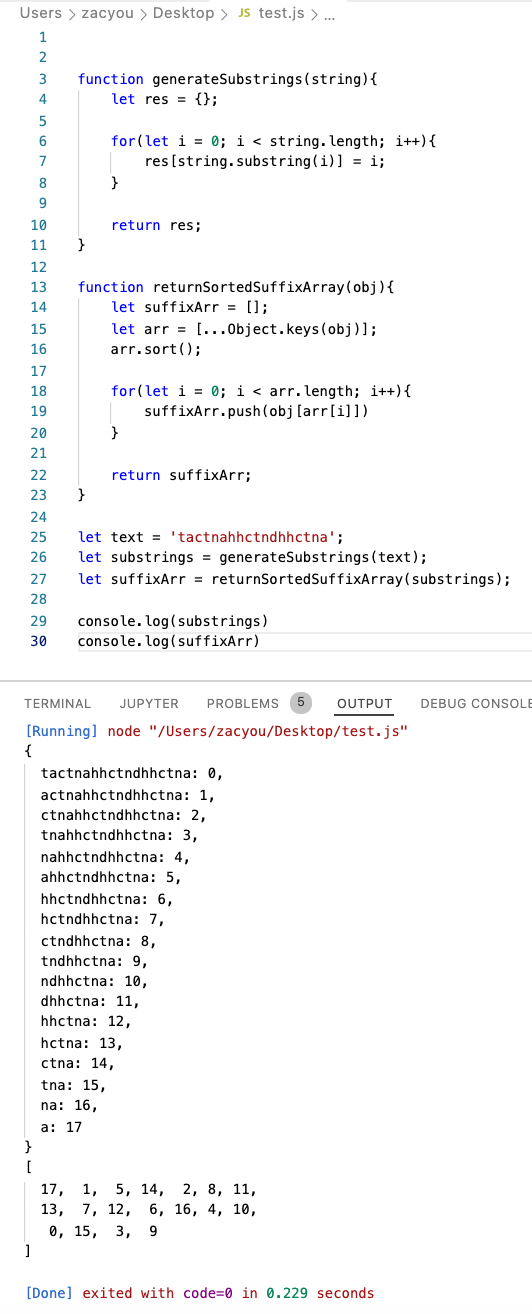


**Problem 3. Suffix trees**



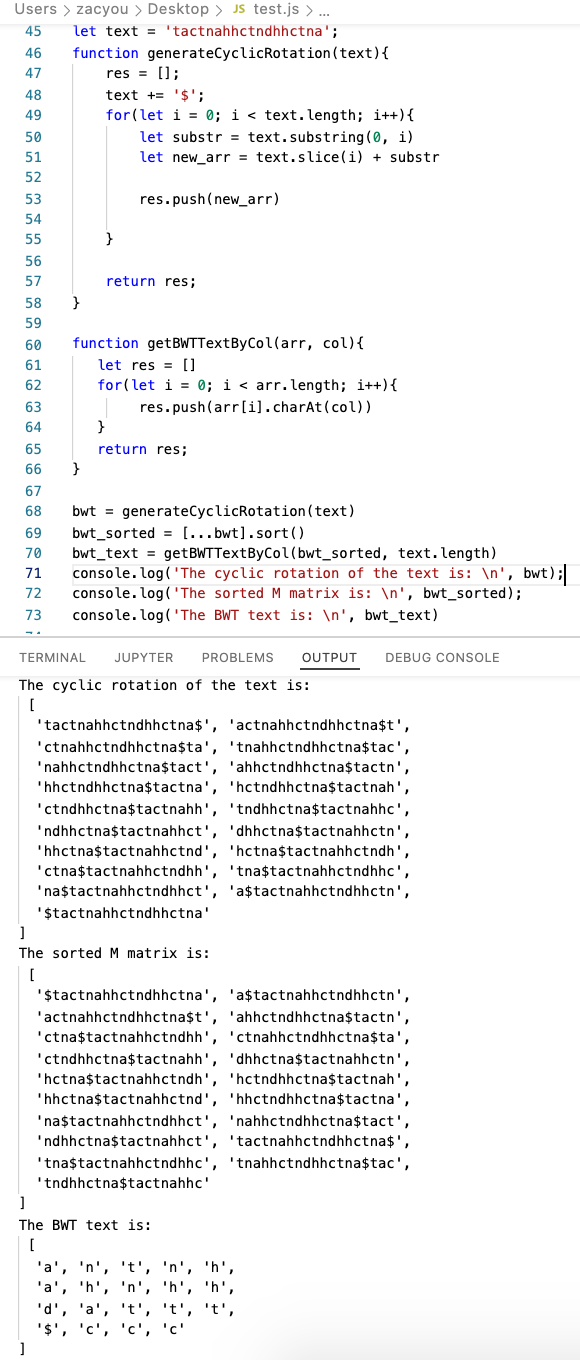
It has 18 leaves. No the number of leaves is not the same as the Trie in problem 1 because the Trie is built based on the patterns; therefore, its number of leaves equals to the number of patterns. The suffix tree, on the other hand, is built upon the entire string input, which has 18 characters.

**Problem 4. Suffix arrays**

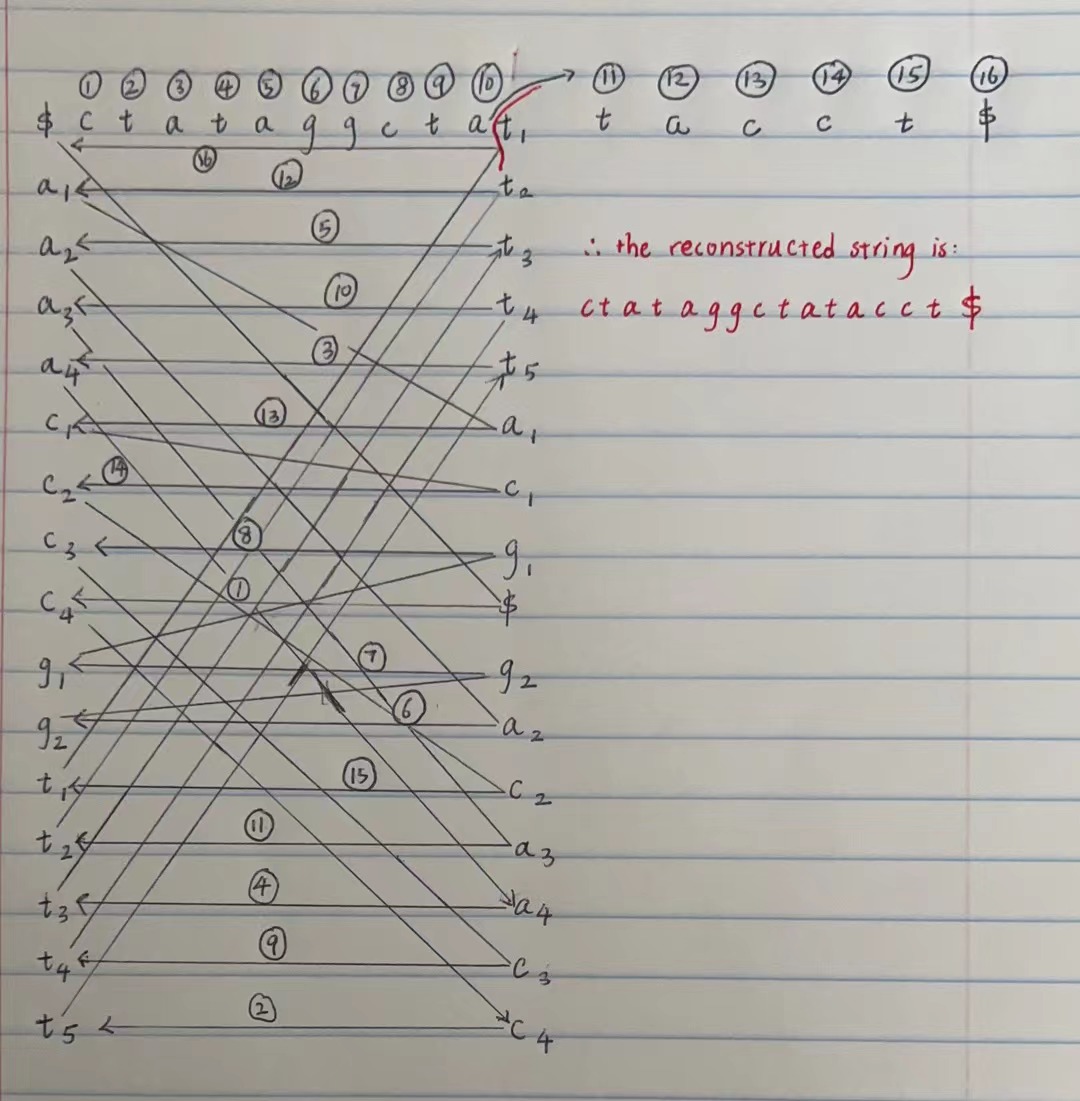


**Problem 5. BWT**

As the result below, the first matrix printed is the cyclic rotation matrix and the second matrix is the M matrix which is sorted, and the BWT text is the last output.

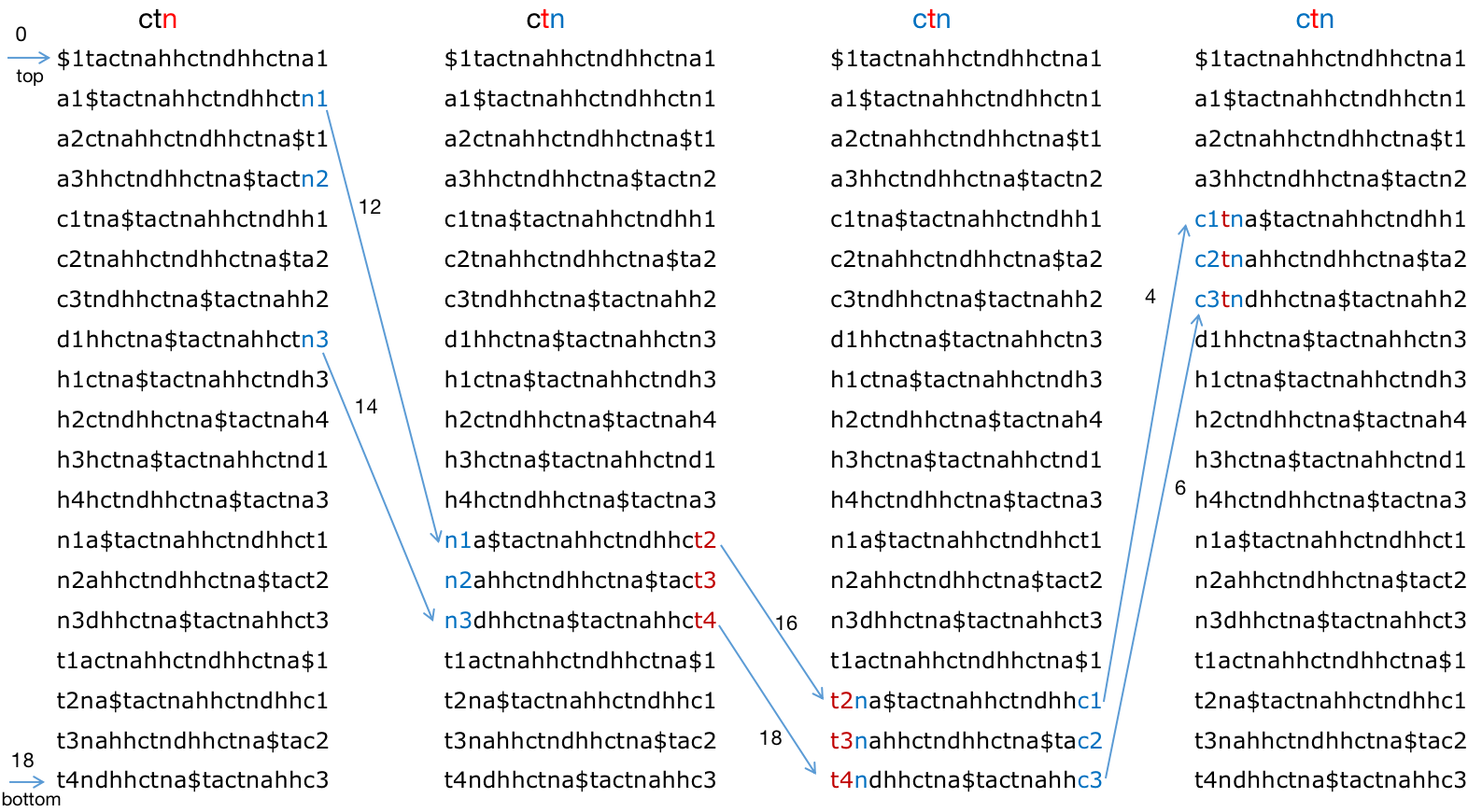


**Problem 6. BWT traversal**

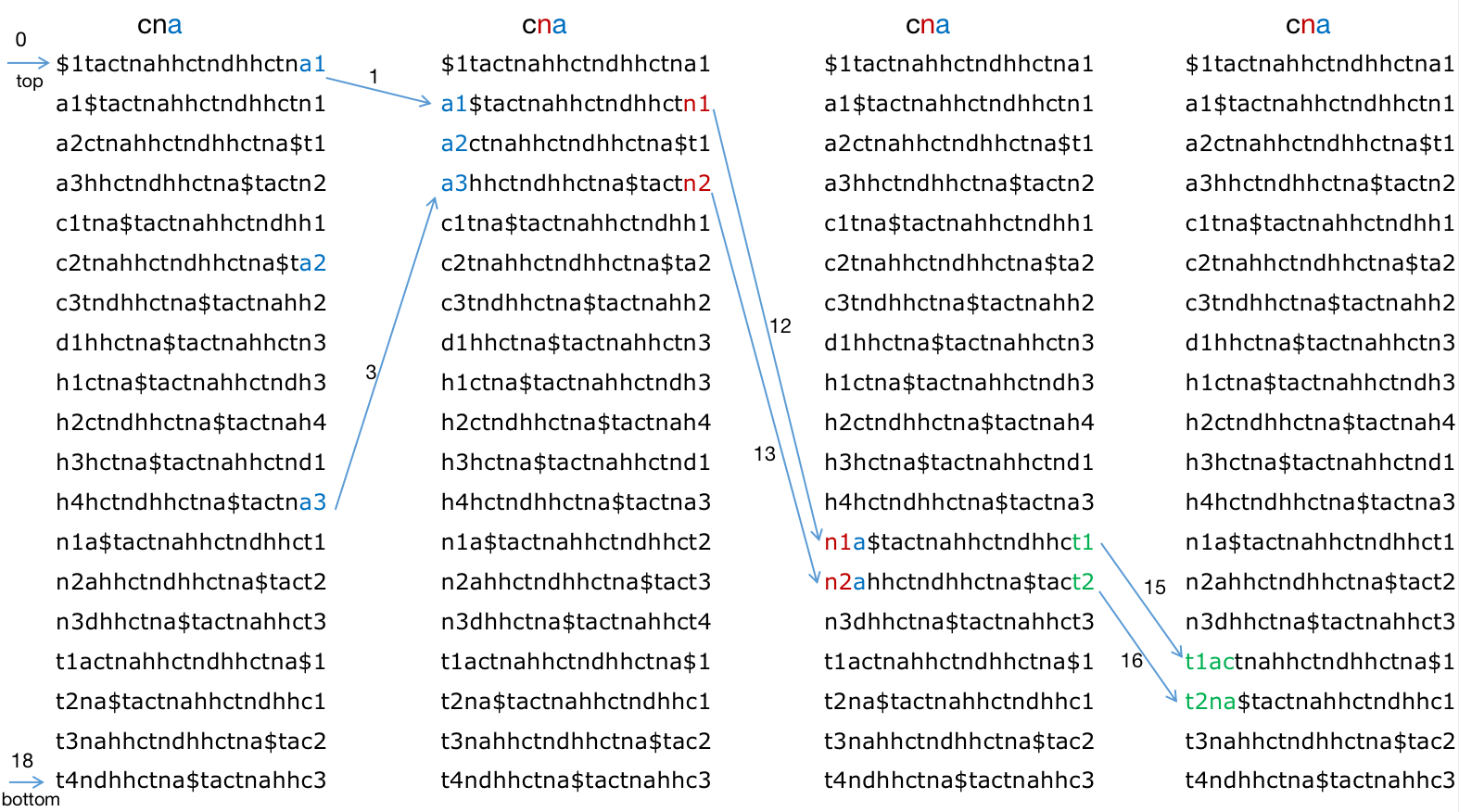


**Problem 7. BWMatching**

|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **FirstColumn** | **LastColumn** | **LastToFirst(i)** |
| 0 | $ | a1 | 1 |
| 1 | a1 | n1 | 12 |
| 2 | a2 | t1 | 15 |
| 3 | a3 | n2 | 13 |
| 4 | c1 | h1 | 8 |
| 5 | c2 | a2 | 2 |
| 6 | c3 | h2 | 9 |
| 7 | d1 | n3 | 14 |
| 8 | h1 | h3 | 10 |
| 9 | h2 | h4 | 11 |
| 10 | h3 | d1 | 7 |
| 11 | h4 | a3 | 3 |
| 12 | n1 | t2 | 16 |
| 13 | n2 | t3 | 17 |
| 14 | n3 | t4 | 18 |
| 15 | t1 | $ | 0 |
| 16 | t2 | c1 | 4 |
| 17 | t3 | c2 | 5 |
| 18 | t4 | c3 | 6 |



Pattern ‘ctn’ appears 3 times.



No pattern ‘cna’ is found.