

# Ternary Search Tree

Advance String tree data structure

# What is ternary search tree?

- A ternary search tree (TST) is a prefix base tree-based data structure used for efficiently storing and searching strings. Unlike binary trees, each node in a TST has three children: one for characters less than the current node's character, one for characters equal to it, and one for characters greater than it. TSTs are particularly useful for tasks like spell checking, autocomplete, and symbol tables due to their efficient handling of string operations and memory usage.

# Why we learn ternary search tree data structure?

- Efficient Searching
- Memory efficiency
- prefix searching
- ordered operations
- versatility

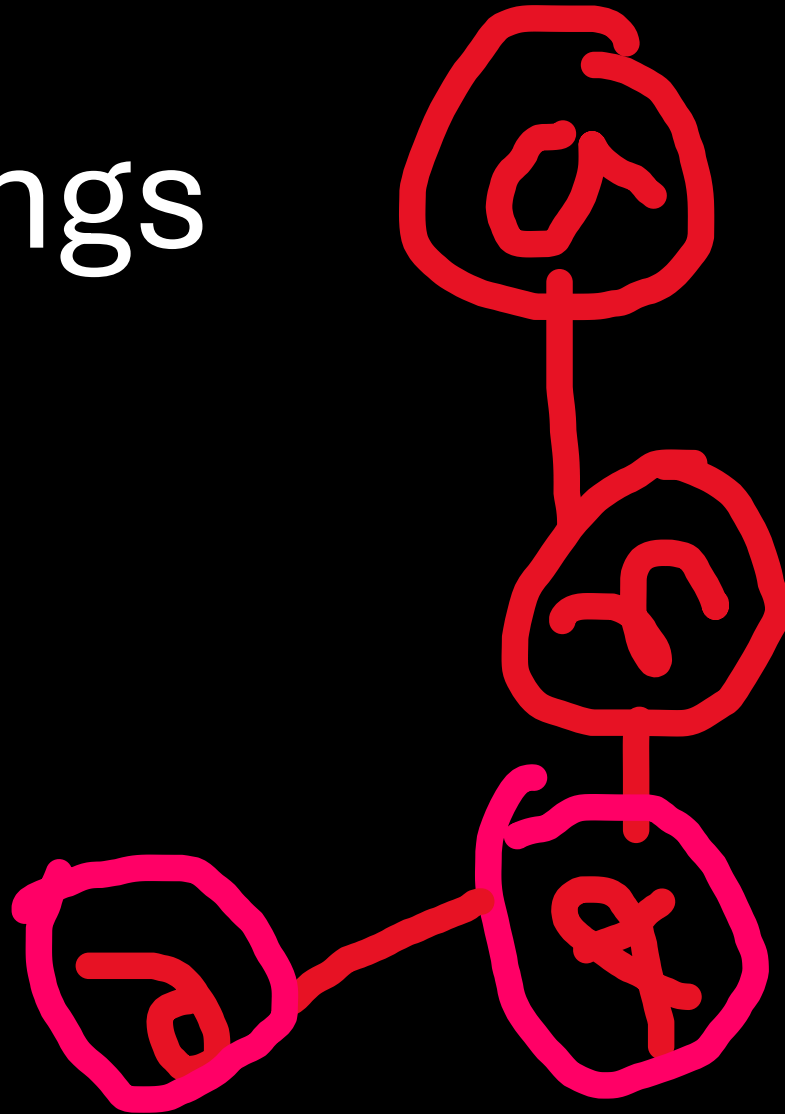
# In summary

- We use ternary search trees when we need to efficiently store and search for strings, especially in scenarios where memory efficiency, fast searching, and prefix-based operations are crucial. Common applications include spell checkers, autocomplete features, symbol tables in compilers, IP routing tables, and more. The benefit of employing a ternary search tree lies in its balance between memory efficiency and fast searching capabilities, making it a versatile choice for a wide range of string-related tasks.

# A simple visualization of ternary search tree

Added strings

1. Ant
2. And



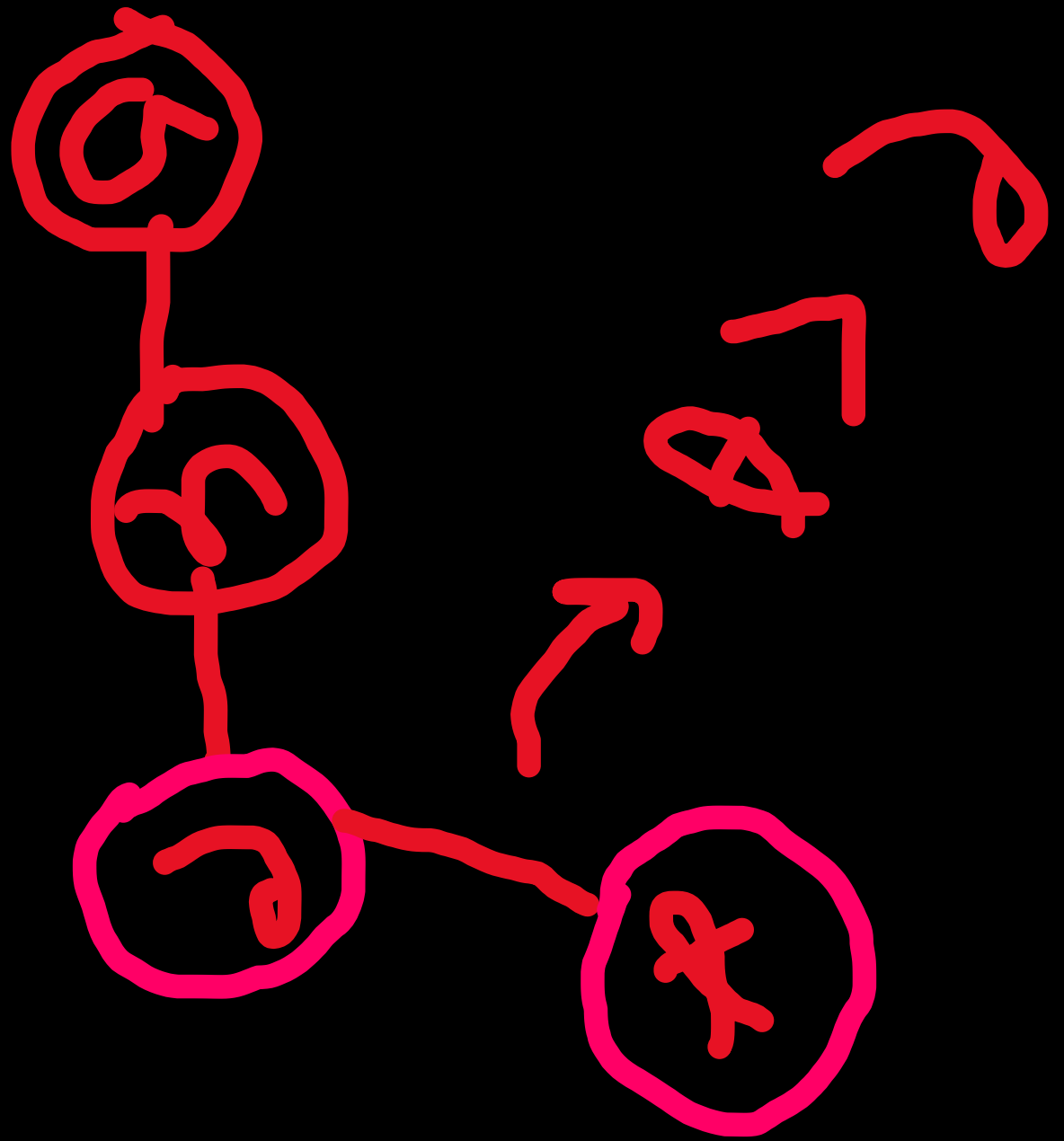
# Node properties of ternary search tree

- Has a character data.
- Has a left child(smaller char than parent node).
- Has a right child(greater char than parent).
- Has a middle child(carry out the same string character).
- A marked up Boolean value to represent this is the end character of the string or not.

Same character  
of a word set as the  
middle node.

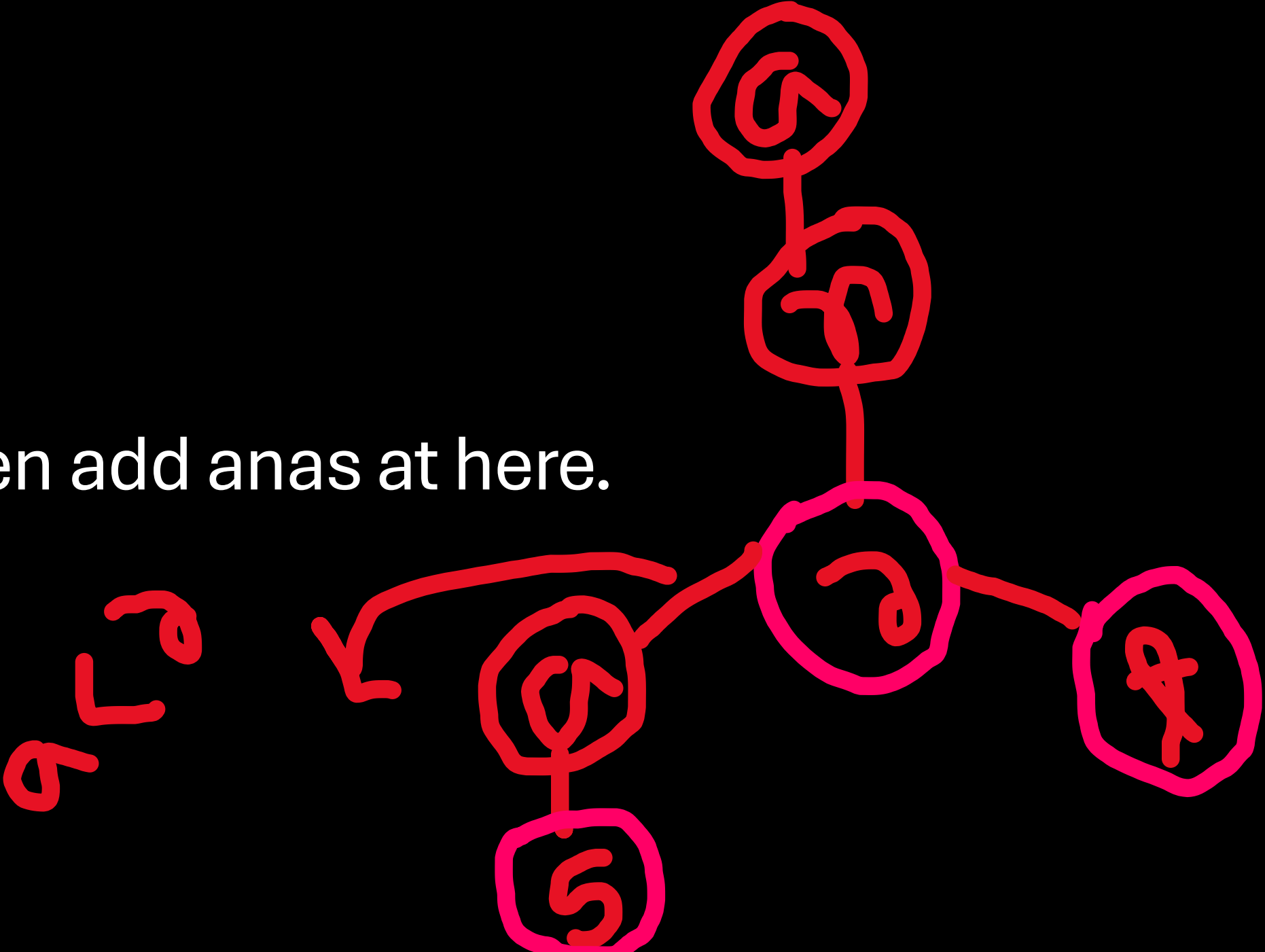


When try to add ant  
at here.





When add anas at here.



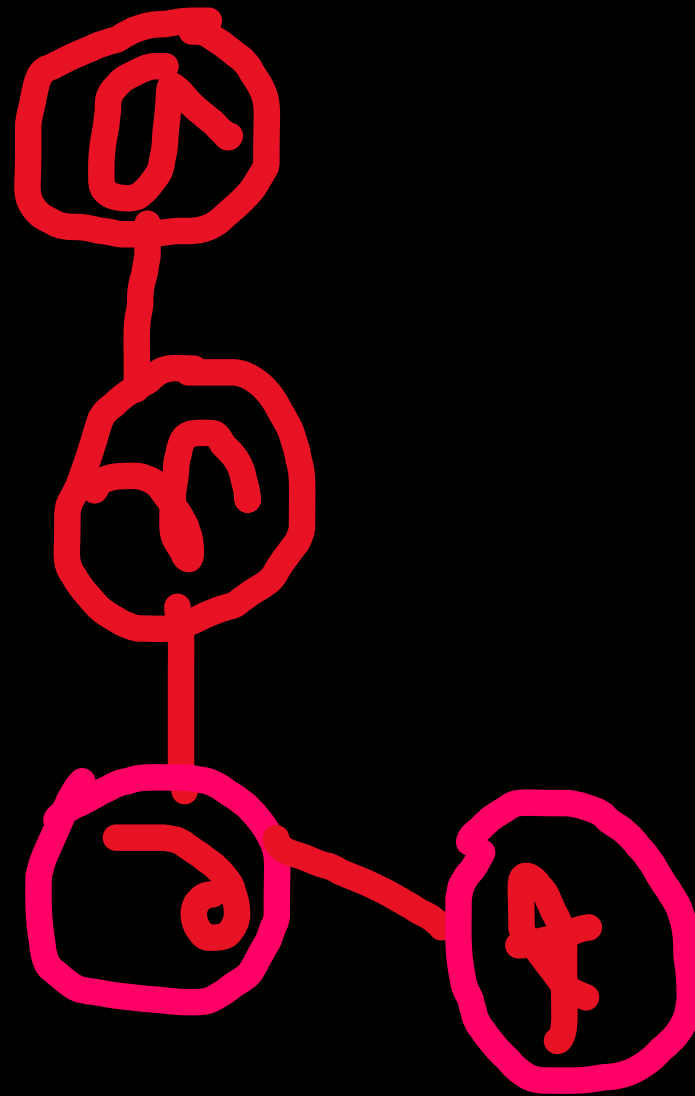
**Let's solve an example to  
understand the insertion.**

and, ant, cat, car, rat, ram, anas, anam, at, cow, tie,  
tide, am.

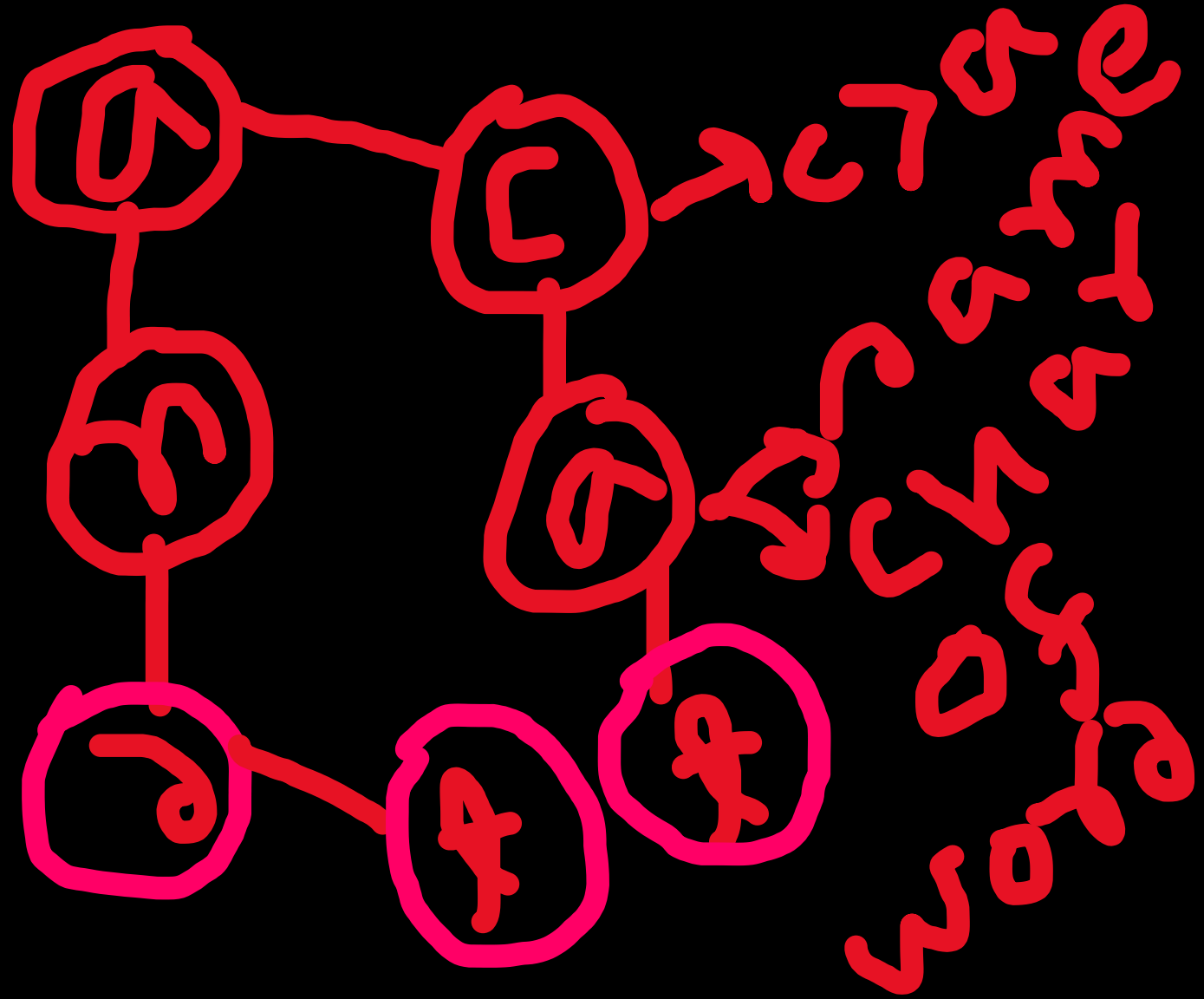


Add and at here.

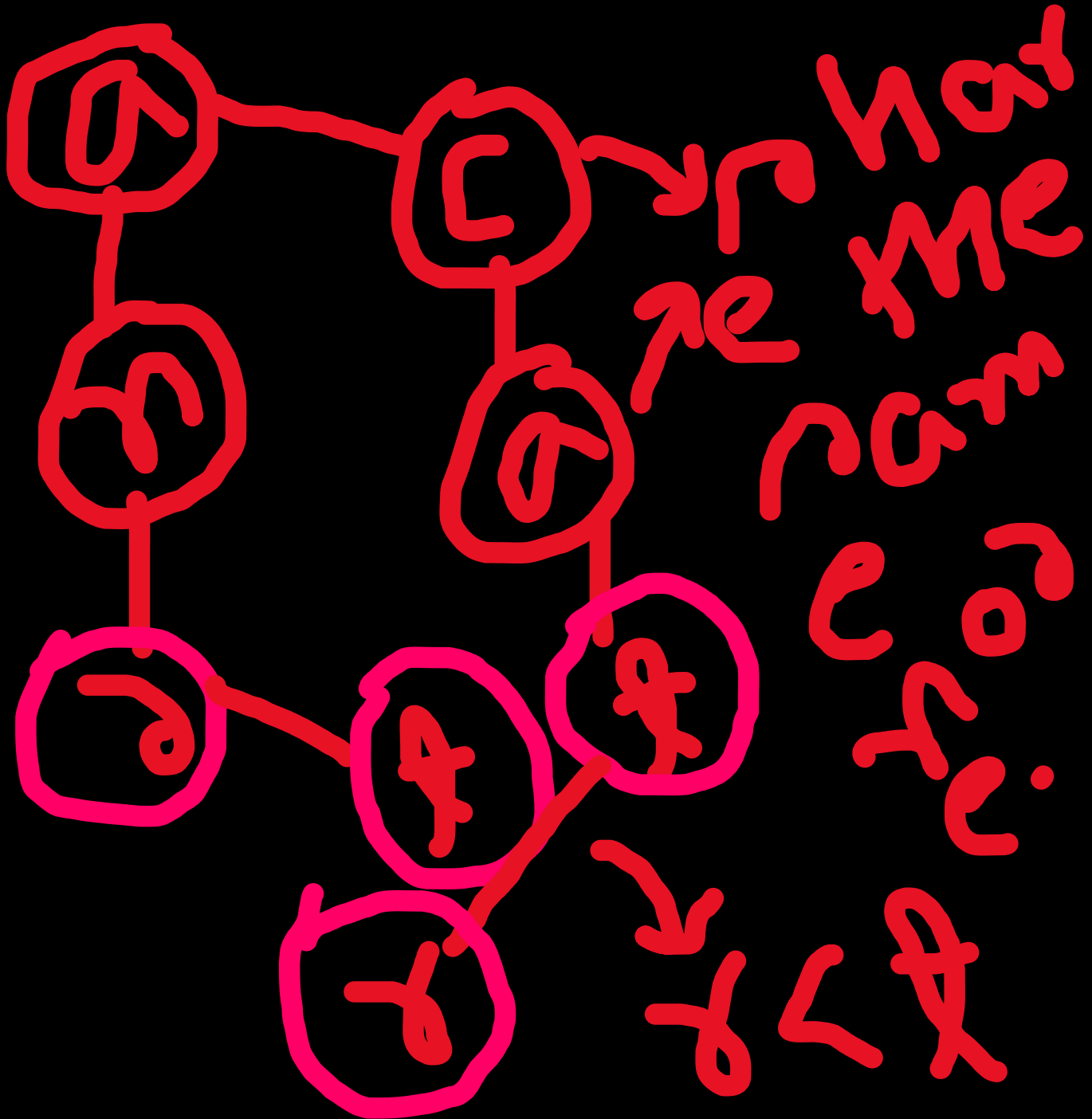
Add ant at here.



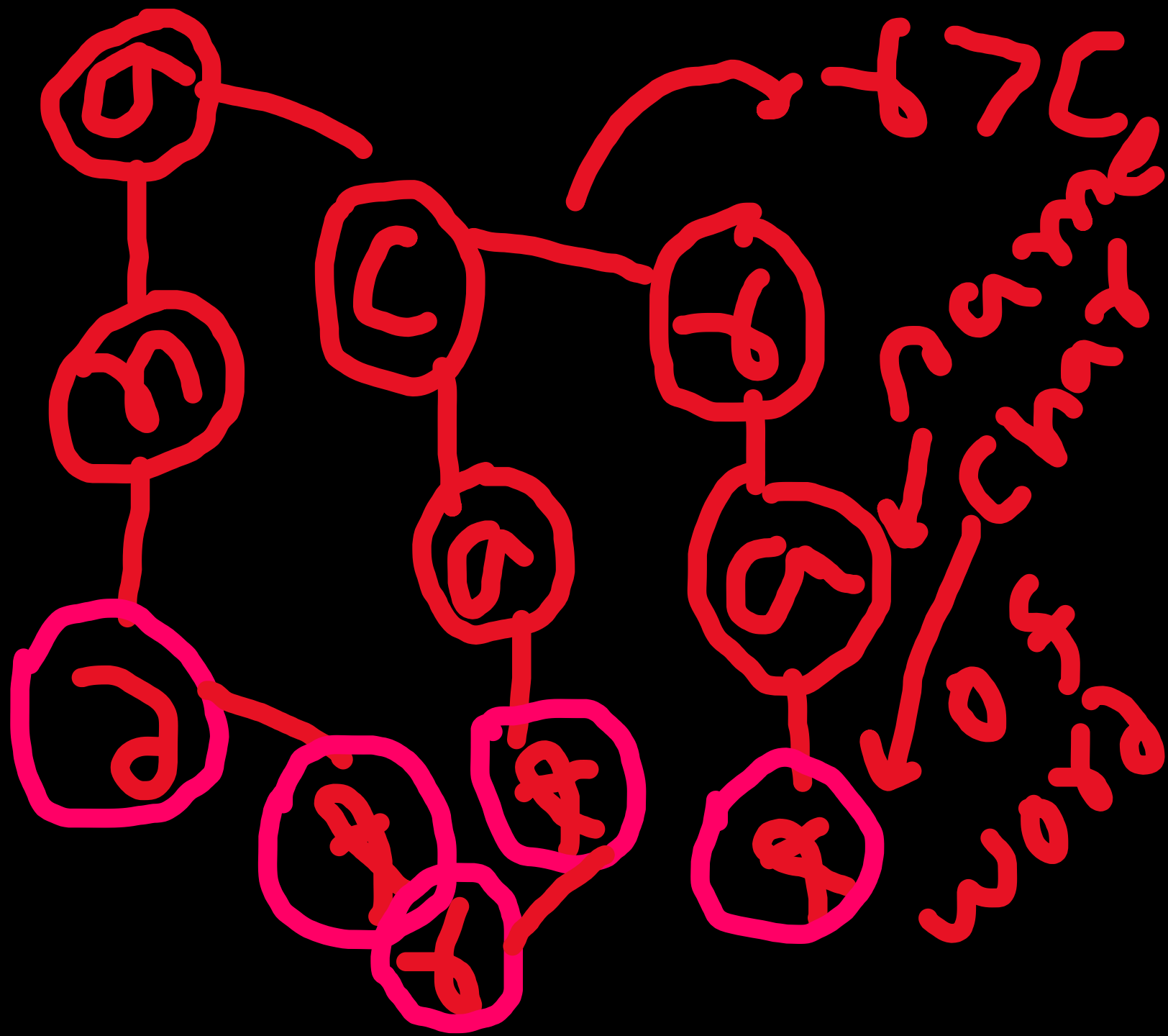
Add cat at here.



Add car at here.

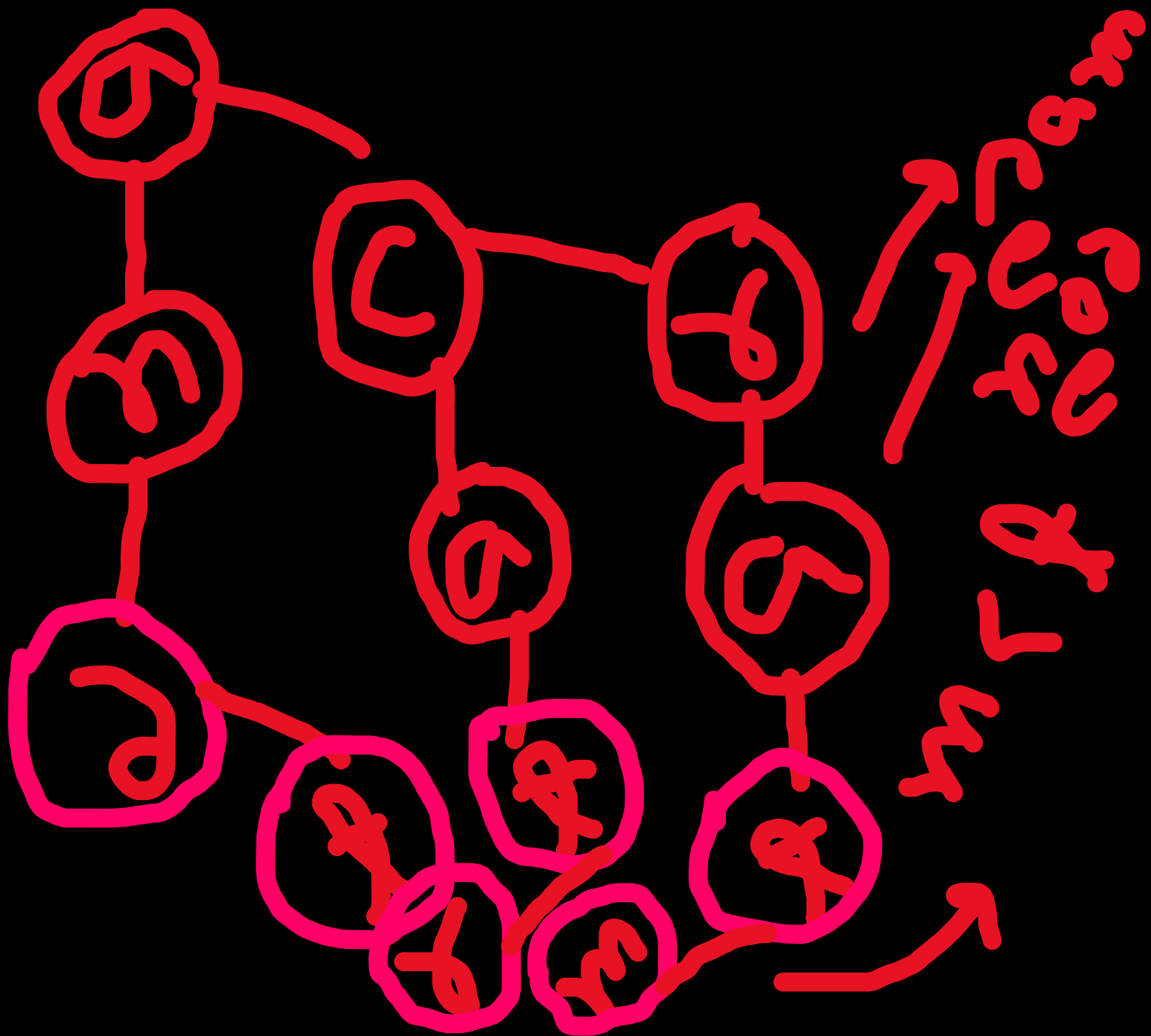


Add rat at here.





Add ram at here.





Add anas at here.



Add anam at here.

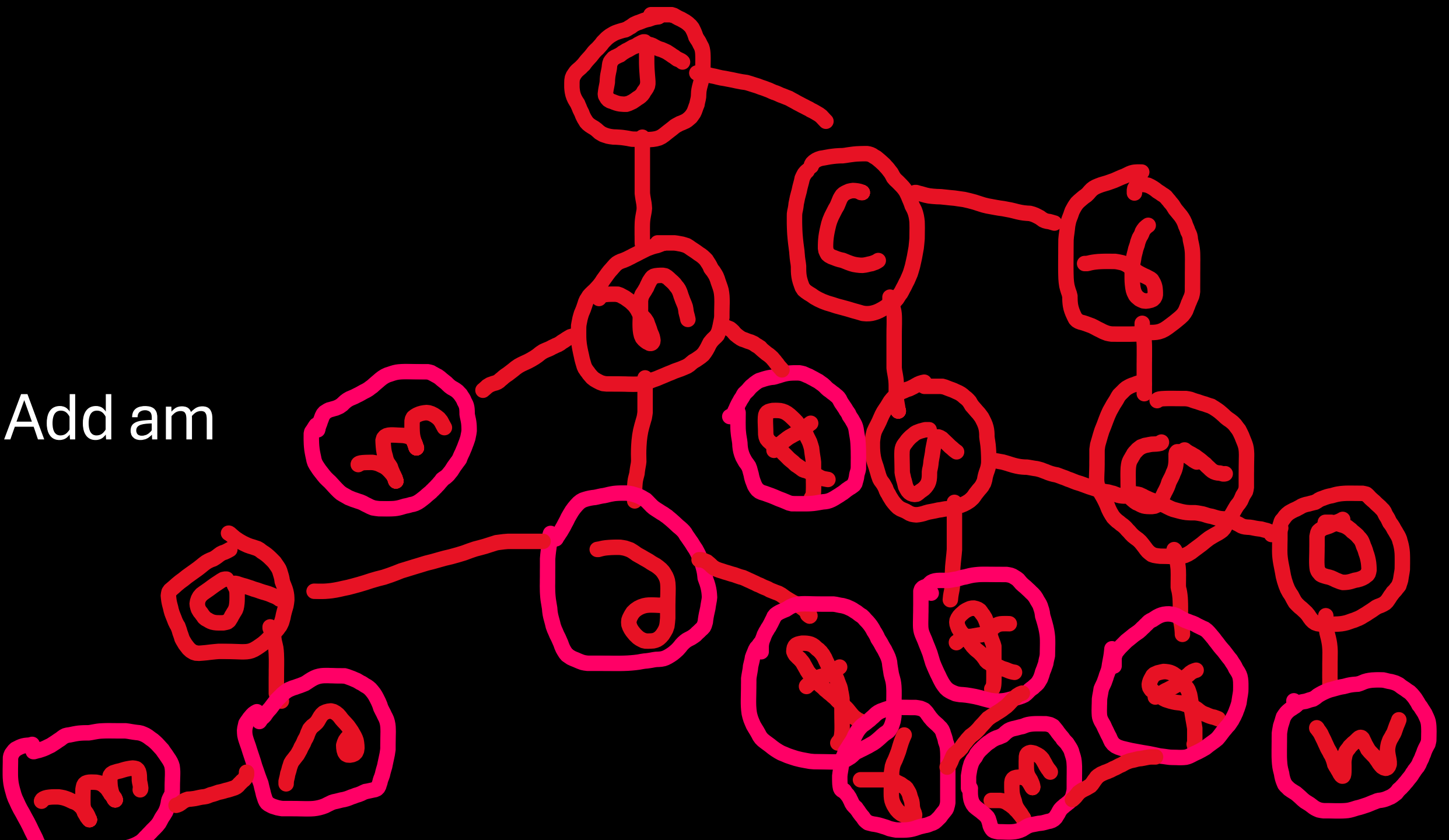




Add cow at here.



Add am






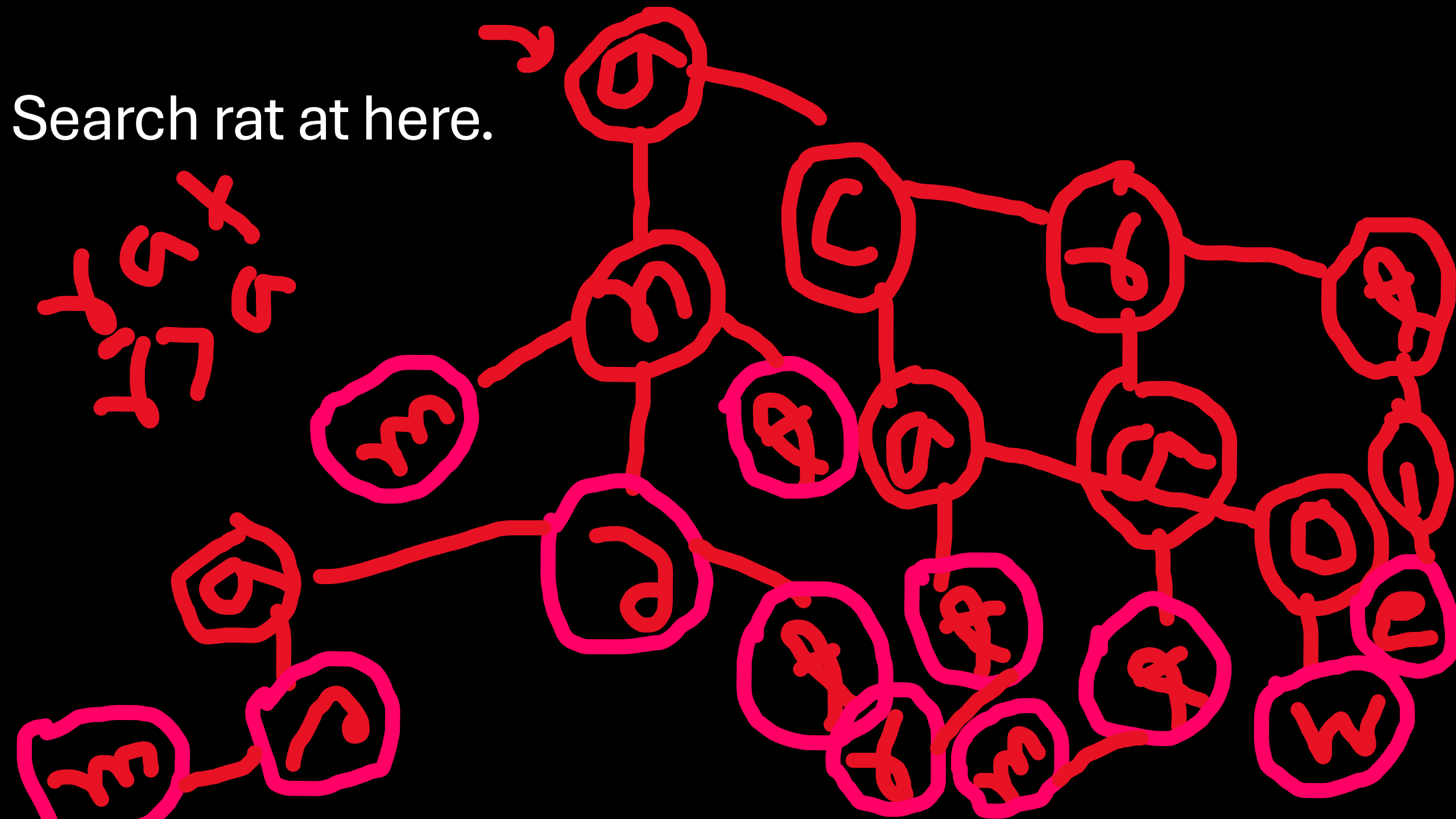
Please try to add tide at  
here.



**Now come to see the operations  
of searching.**

- For searching 
  - we will follow the same rules as bst.
  - If our searching strings character is greater than the present node, then go right.
  - If our present string character is smaller than the present nod, go left.
  - If they are equal go to the middle and continue searching until find the end of the character of the string and that is marked as true at there.

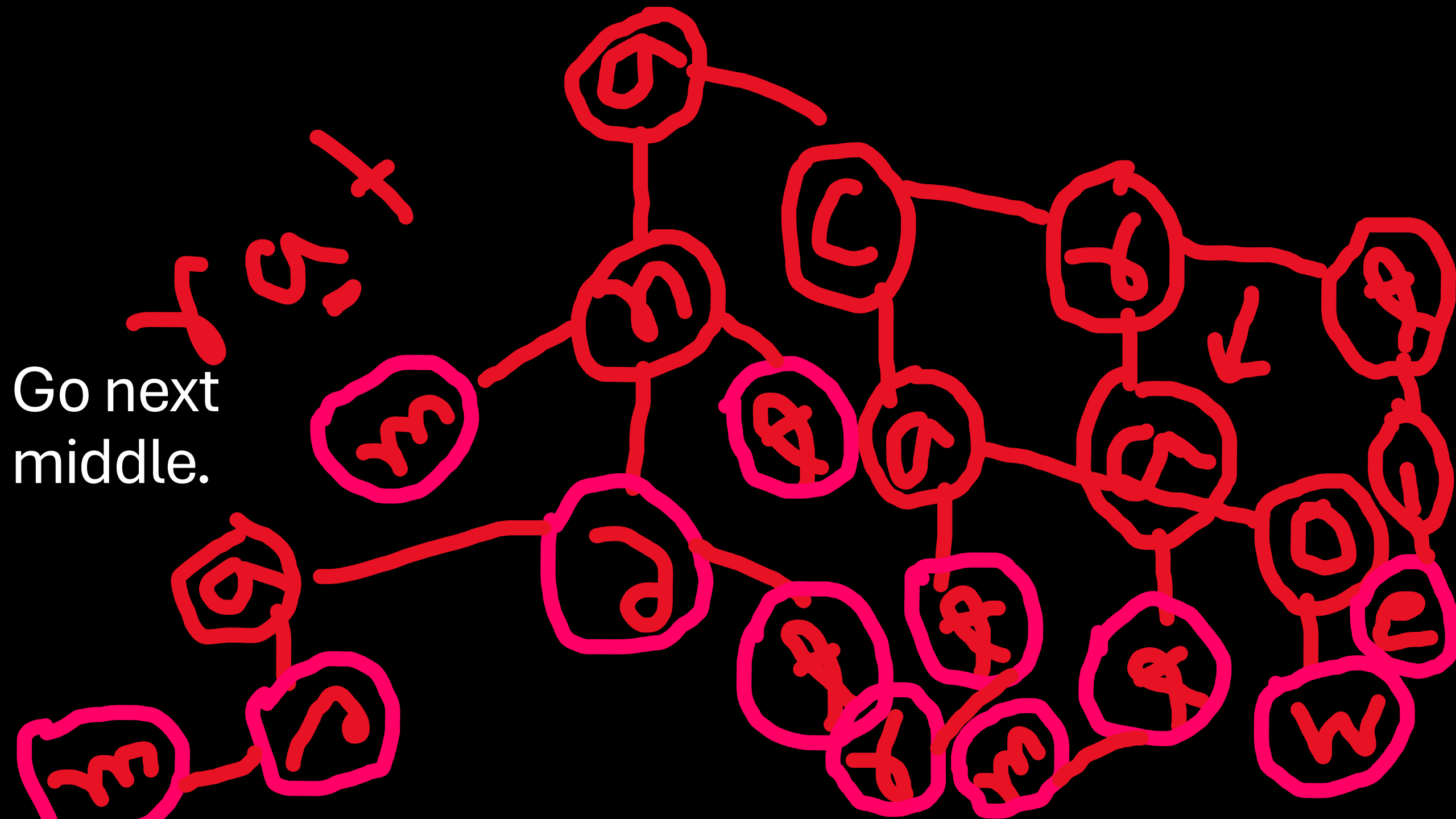
Search rat at here.



Both of them same

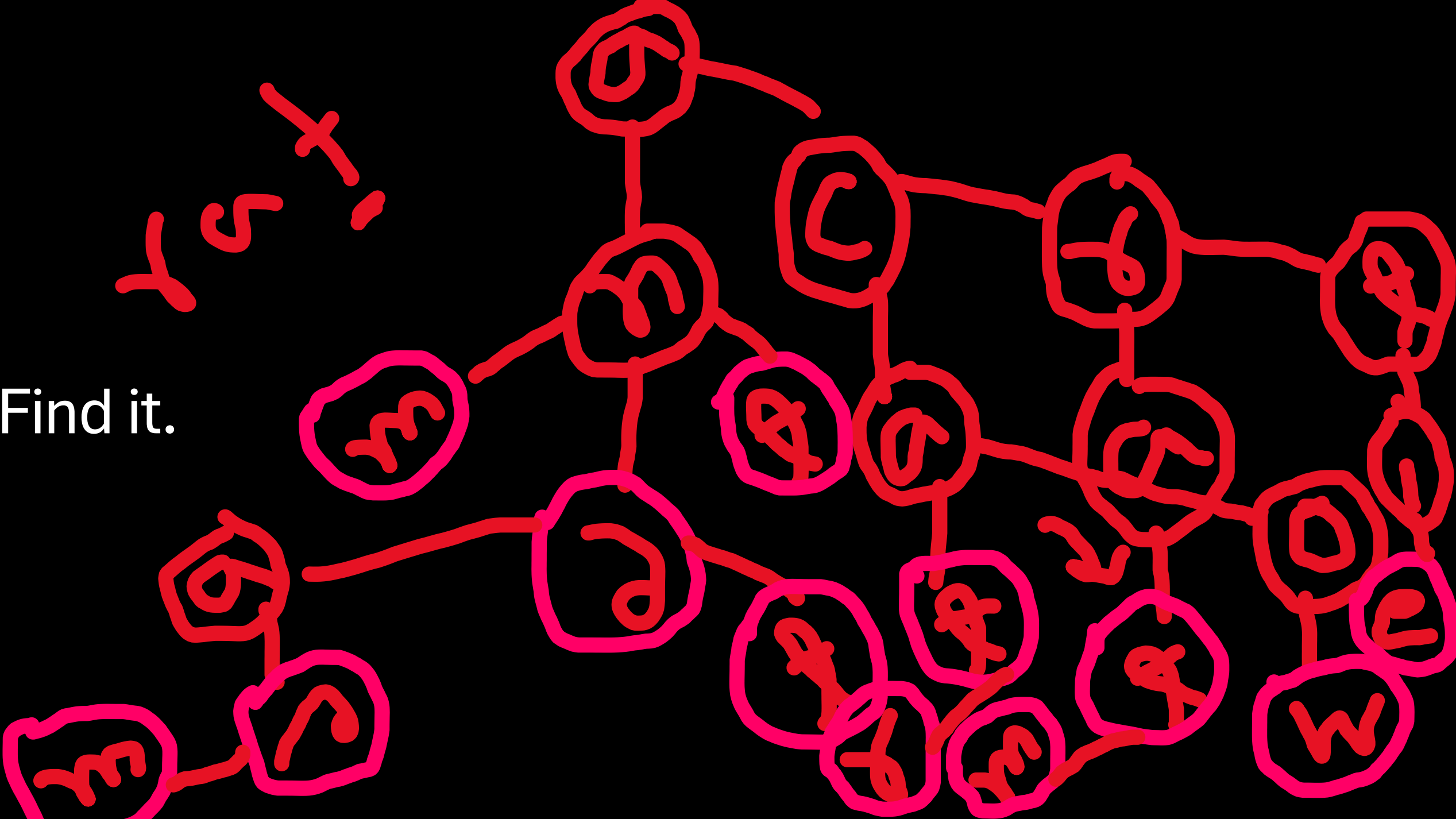


Go next  
middle.



Find it.

Test!



Now come for the delete operations.

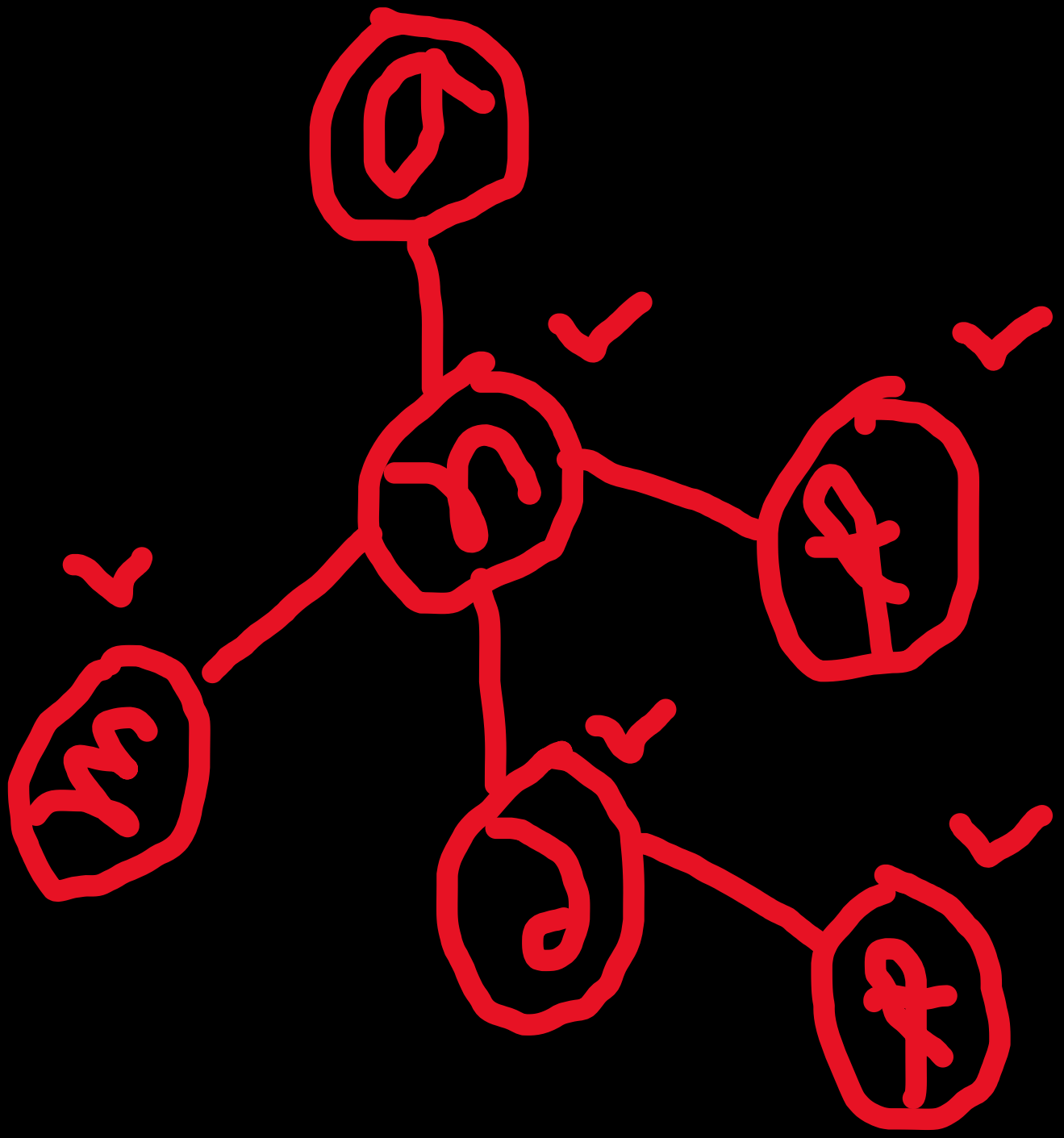
- This is as like as same as the bst and trie data structure.
  - First time find the word we want to delete. Then mark the end character as false if this is prefix of other string.
  - If no other string use that node, then just delete it.



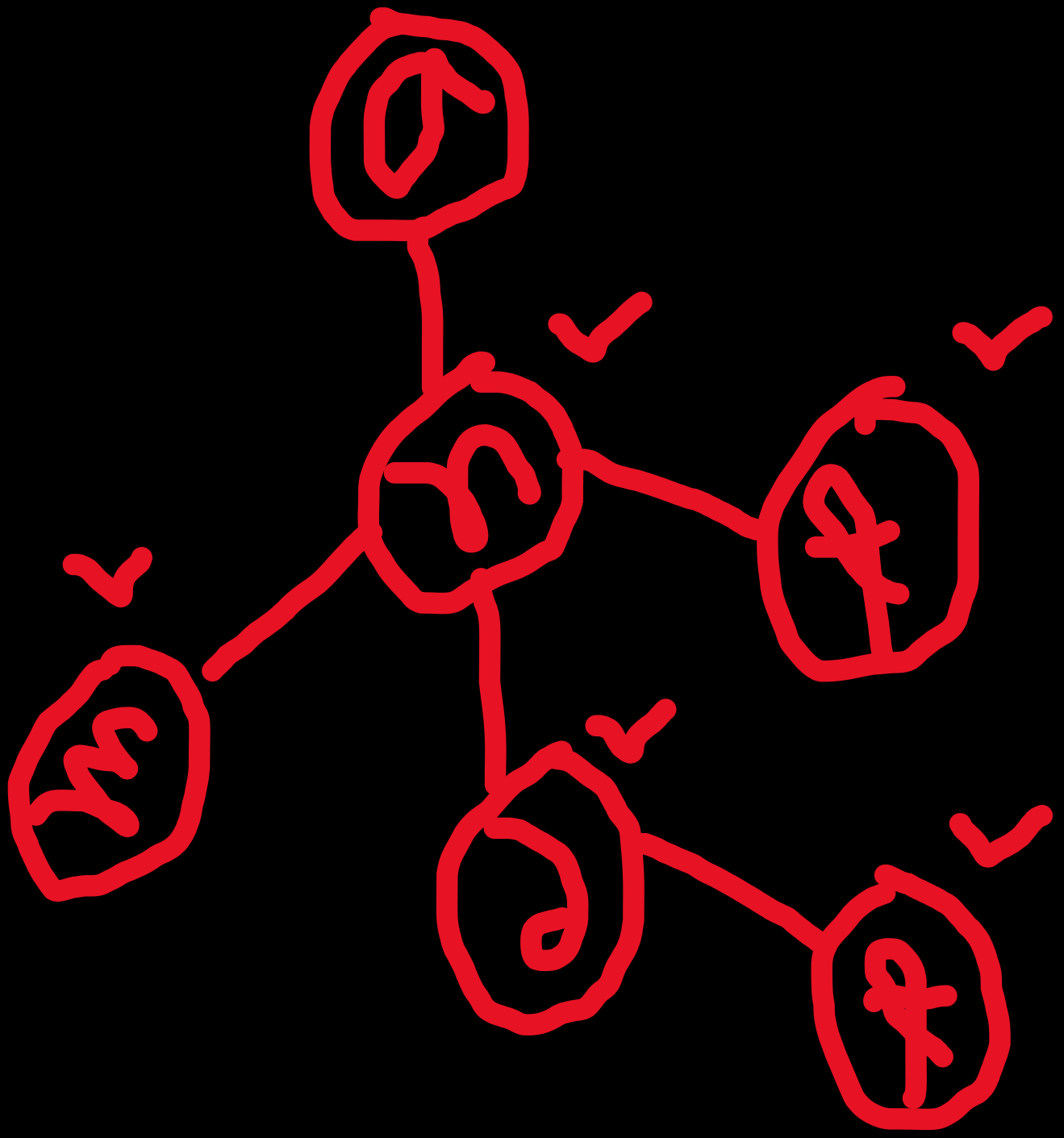
Let's solve an example.

Handwritten numbers and symbols arranged in two columns:

- Column 1 (left): 5, 6, 9, 9, 9
- Column 2 (right): 2, 3, 3, 4, 5

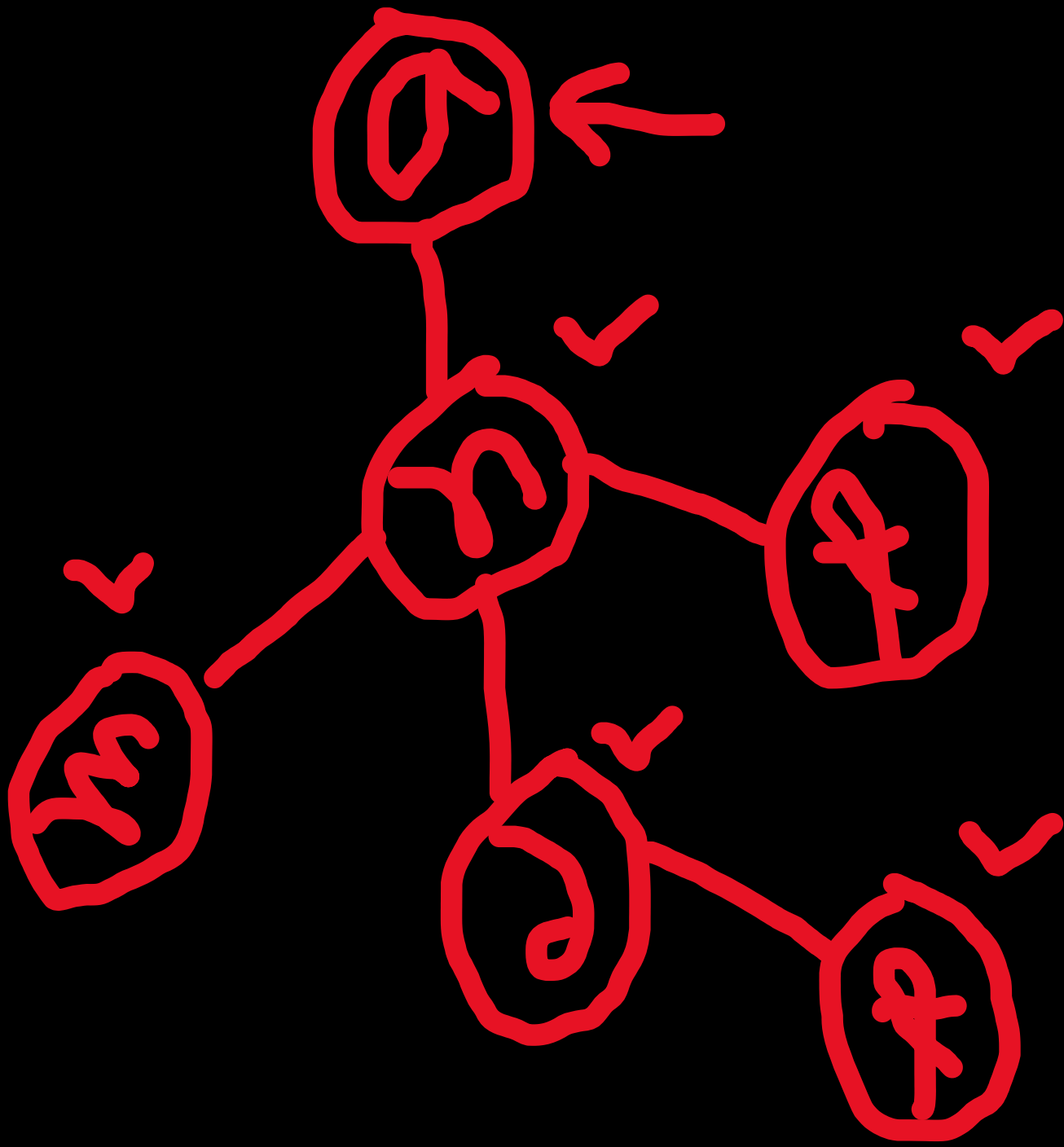


Handwritten notes in red ink, organized into two columns. The right column contains the characters 'a', 't', 'e', 'r', 'n', 'e', 't', 'i', 't', 'y', 'i', 'n', 'g', 't', 'h', 'e', 'i', 'n', 't', 'e', 'r', 'n', 'e', 't'. The left column contains the characters 'i', 'n', 't', 'e', 'r', 'n', 'e', 't', 'i', 'n', 'g', 't', 'h', 'e', 'i', 'n', 't', 'e', 'r', 'n', 'e', 't'. Each character is written in a stylized, cursive font.

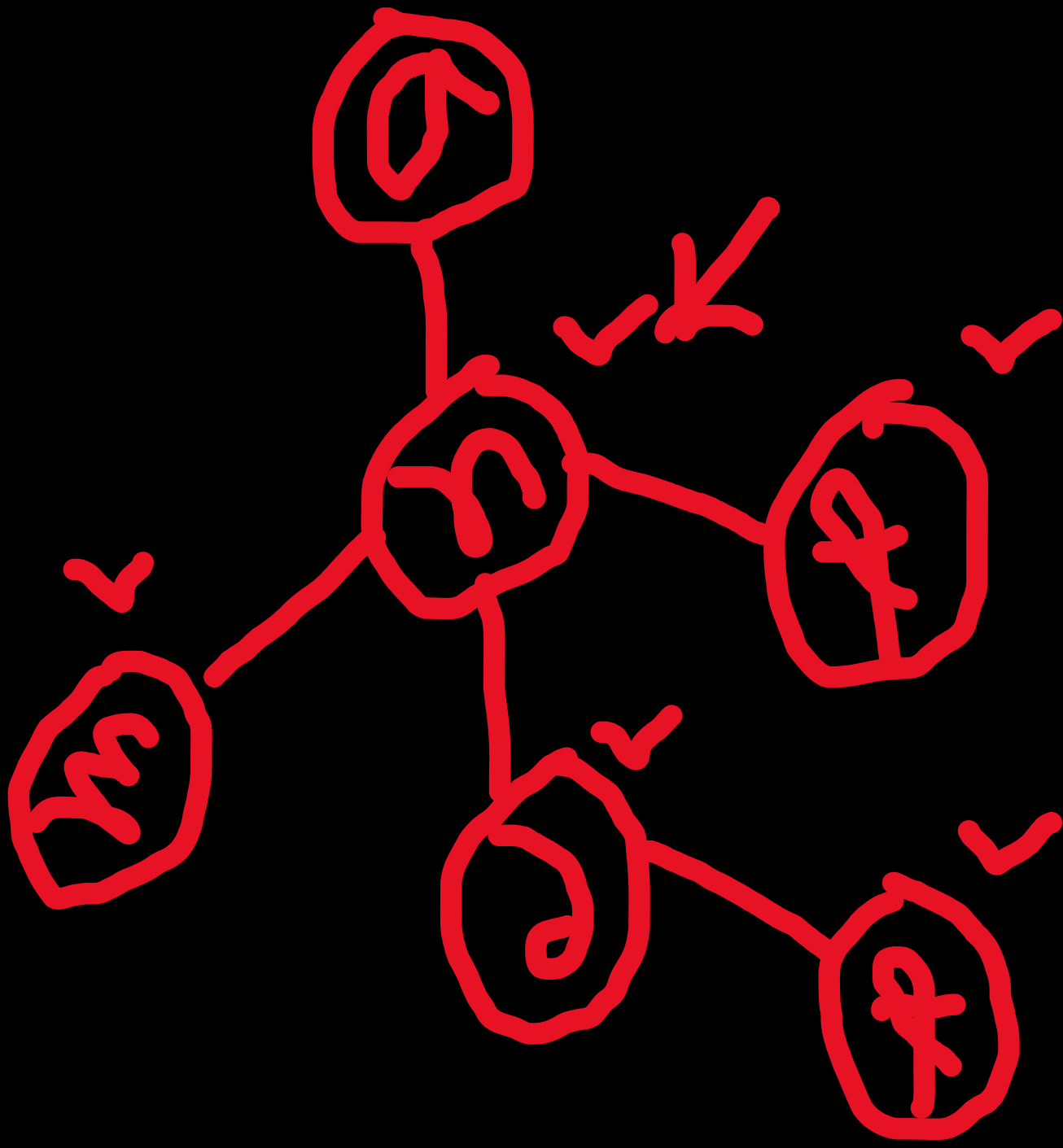


Handwritten numbers and symbols arranged in two columns:

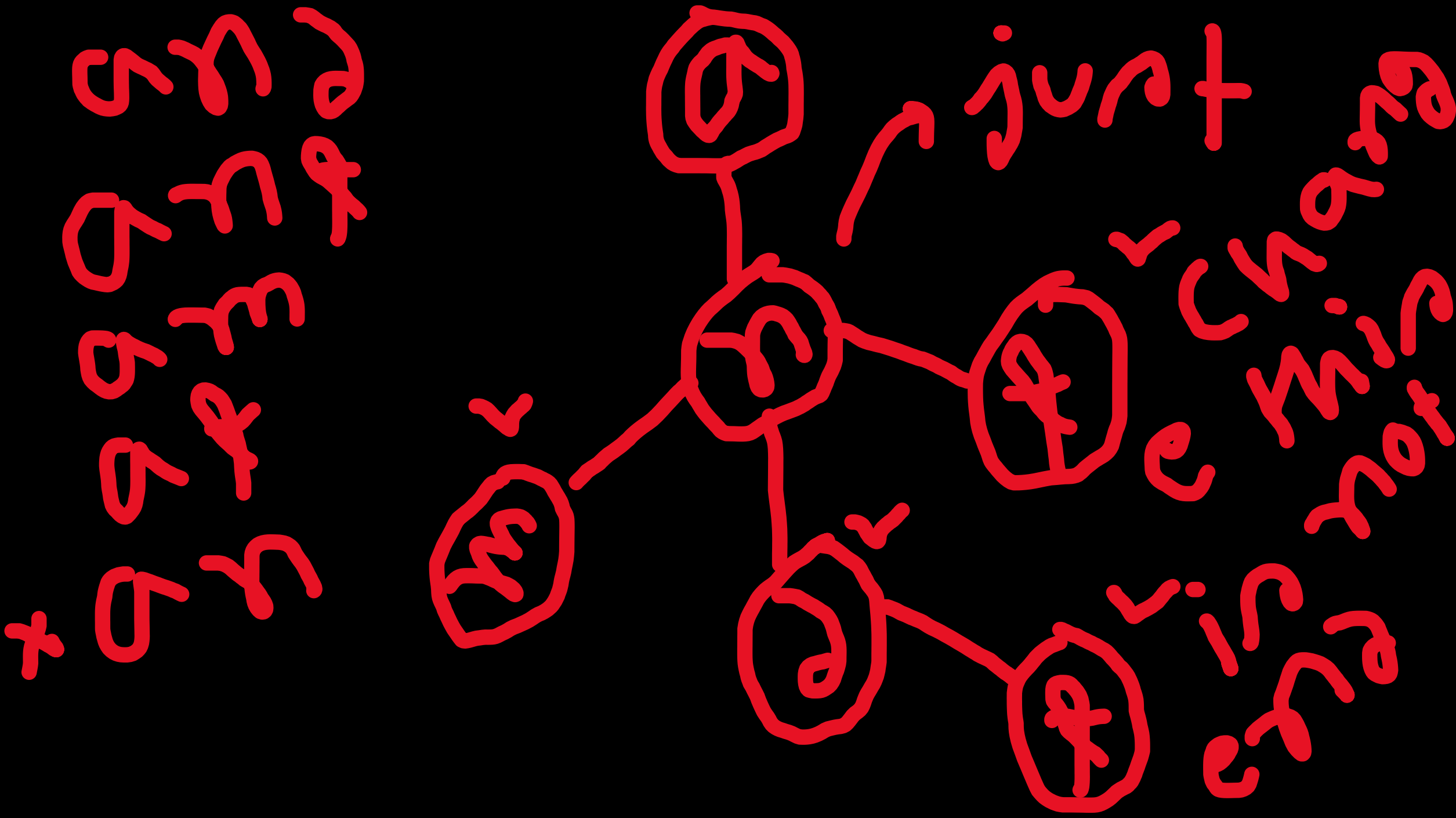
- Column 1 (left): 5, 6, 9, 9
- Column 2 (right): 2, 4, 3, 4



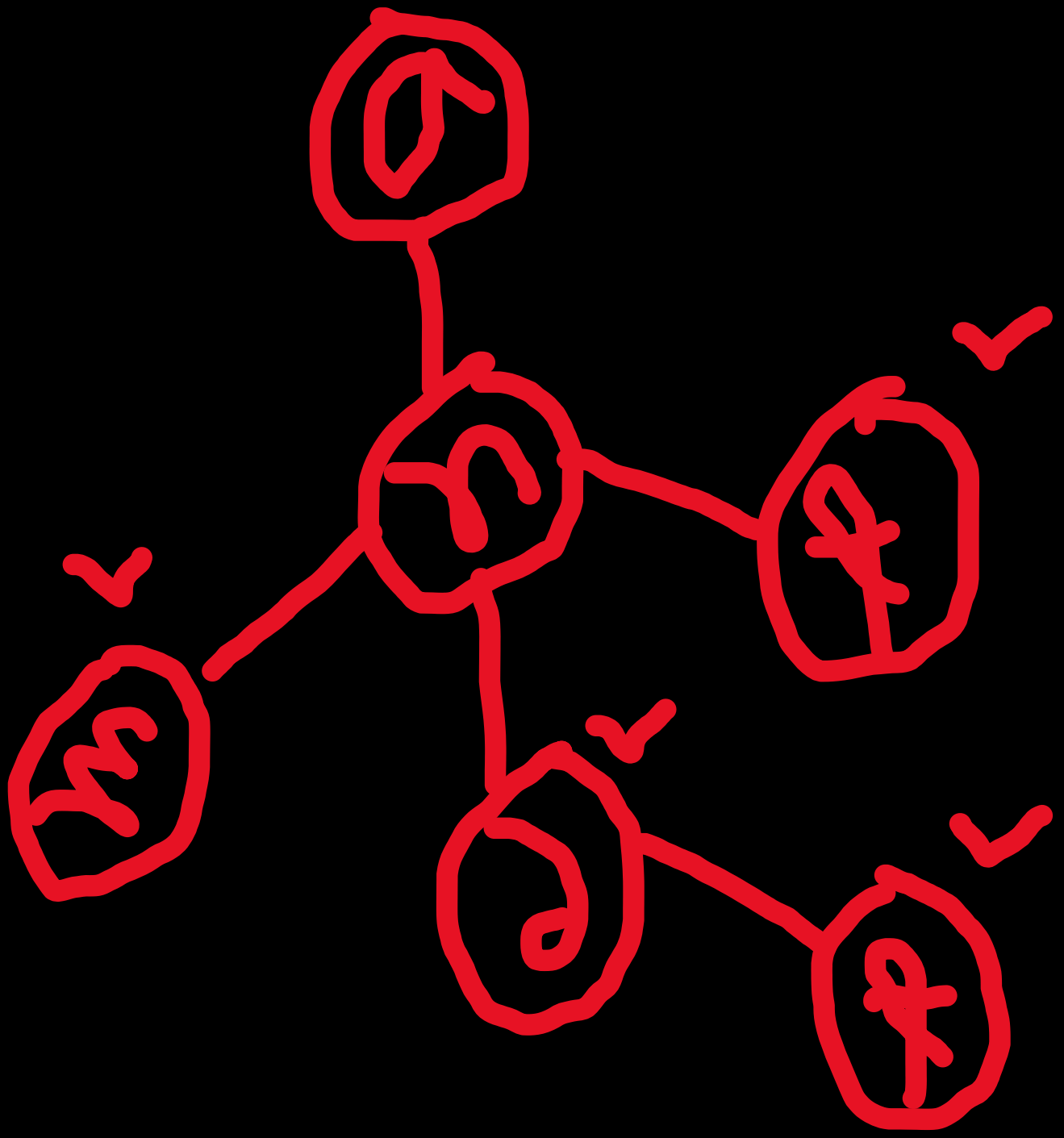
9 9 9 9 9  
3 4 3 4 3  
5 6 5 6 5



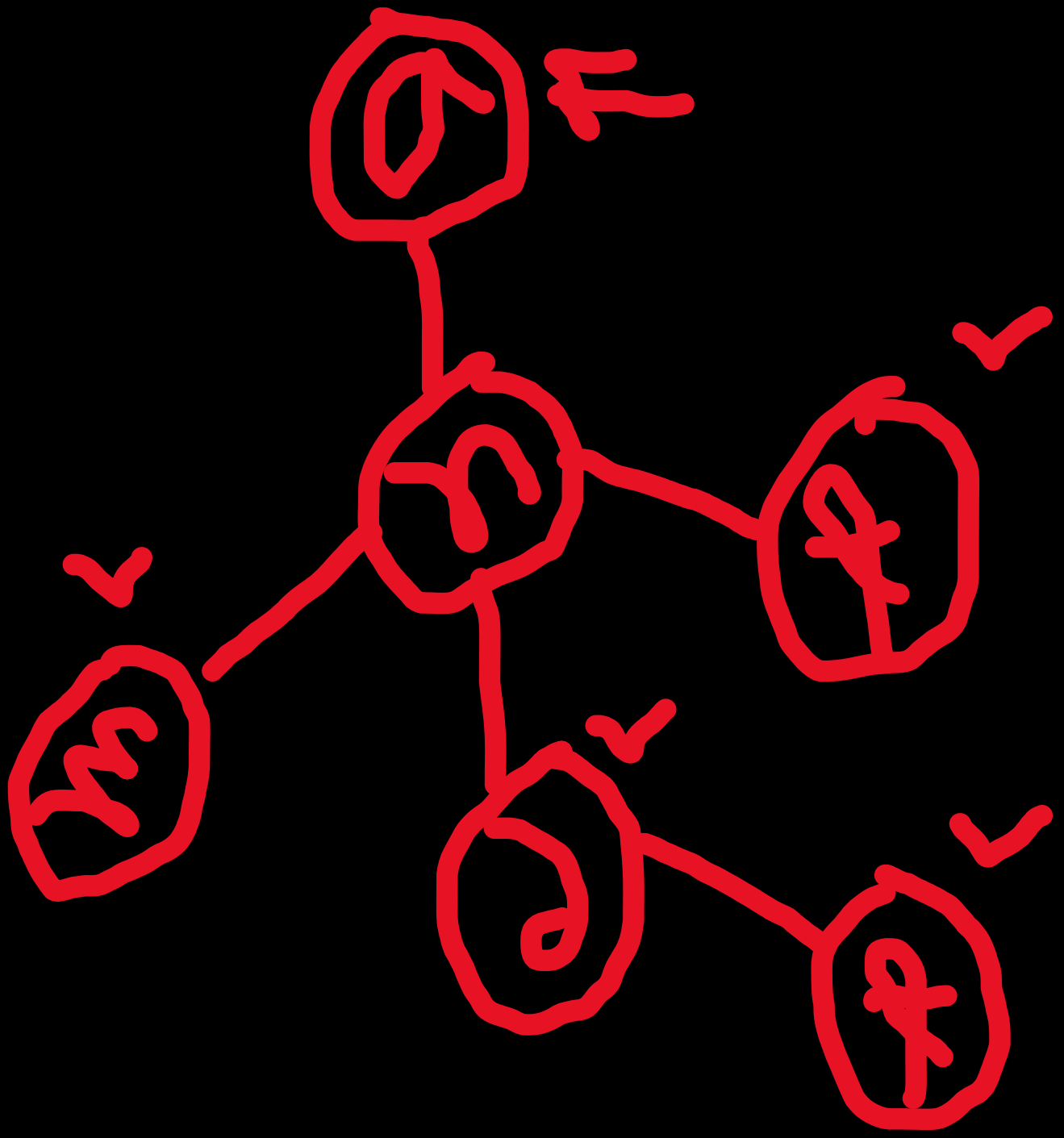
9 9 9 9 9  
5 6 5 6 5  
7 8 7 8 7



Handwritten notes in red ink, organized into two columns. The right column contains the characters 'a', 'c', 'e', 'g', 'i', 'k', 'm', 'o', 'q', 's', 'u', 'w', 'y'. The left column contains the characters 'b', 'd', 'f', 'h', 'j', 'l', 'n', 'p', 'r', 't', 'v', 'x', 'z'. Each character is accompanied by a small checkmark.

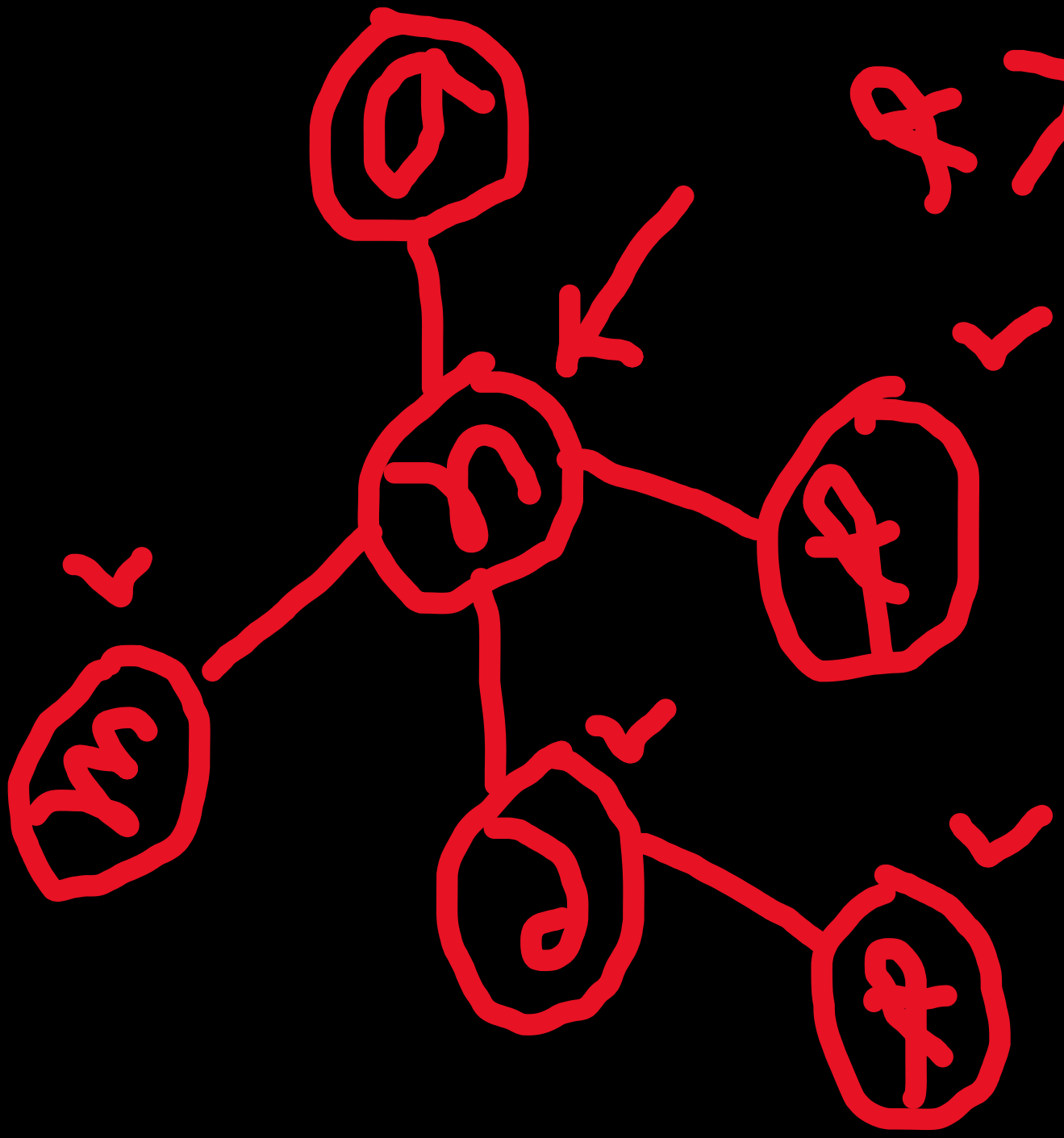


Handwritten notes in red ink, organized into two columns. The right column contains the numbers 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50. The left column contains the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50.





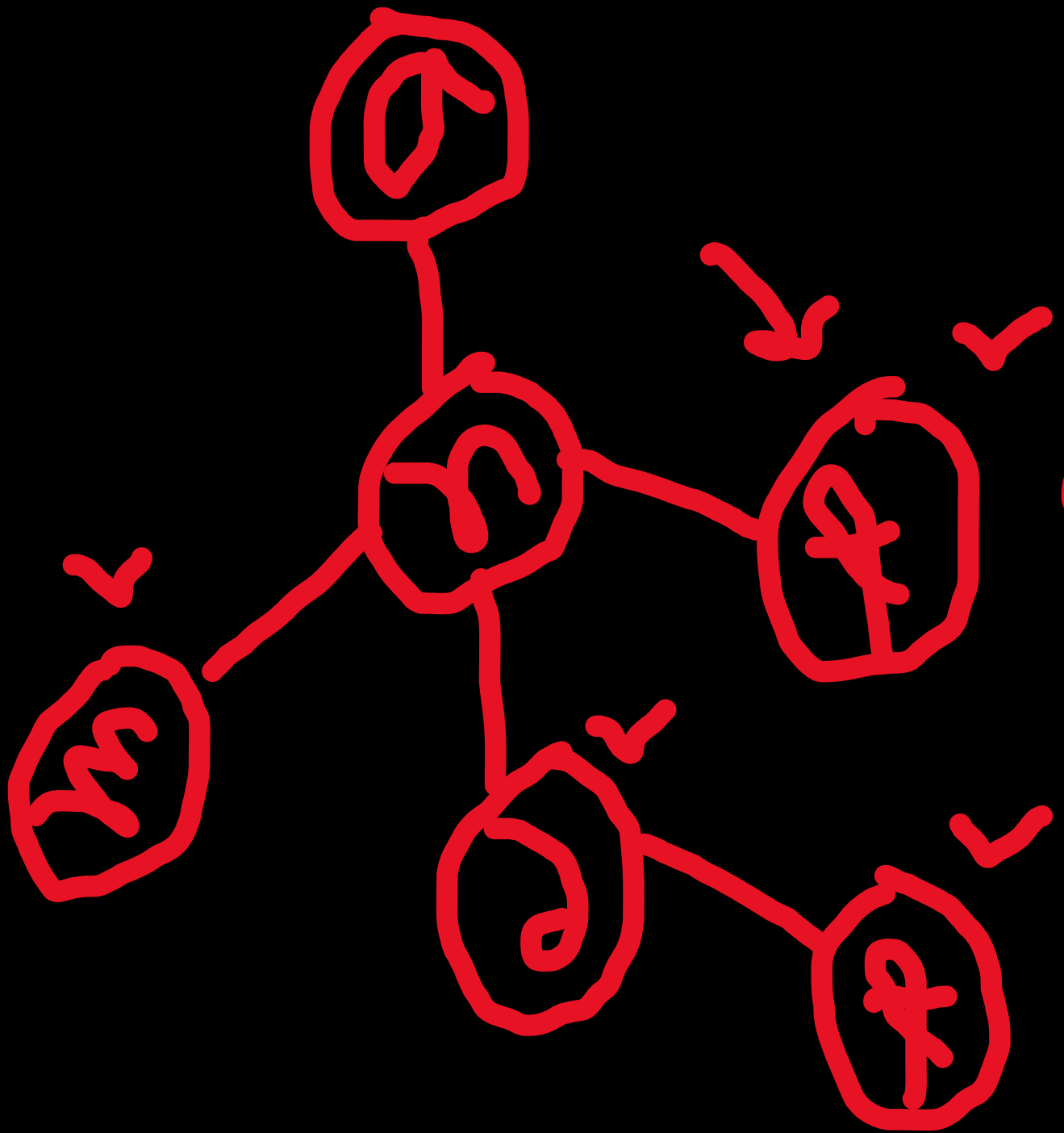
Handwritten notes on the left side of the page, consisting of several vertical columns of scribbled characters and symbols.



Handwritten notes on the right side of the page, including the word 'height' written vertically and several arrows pointing downwards.

Handwritten notes in Hindi, possibly a list of items or a sequence of characters:

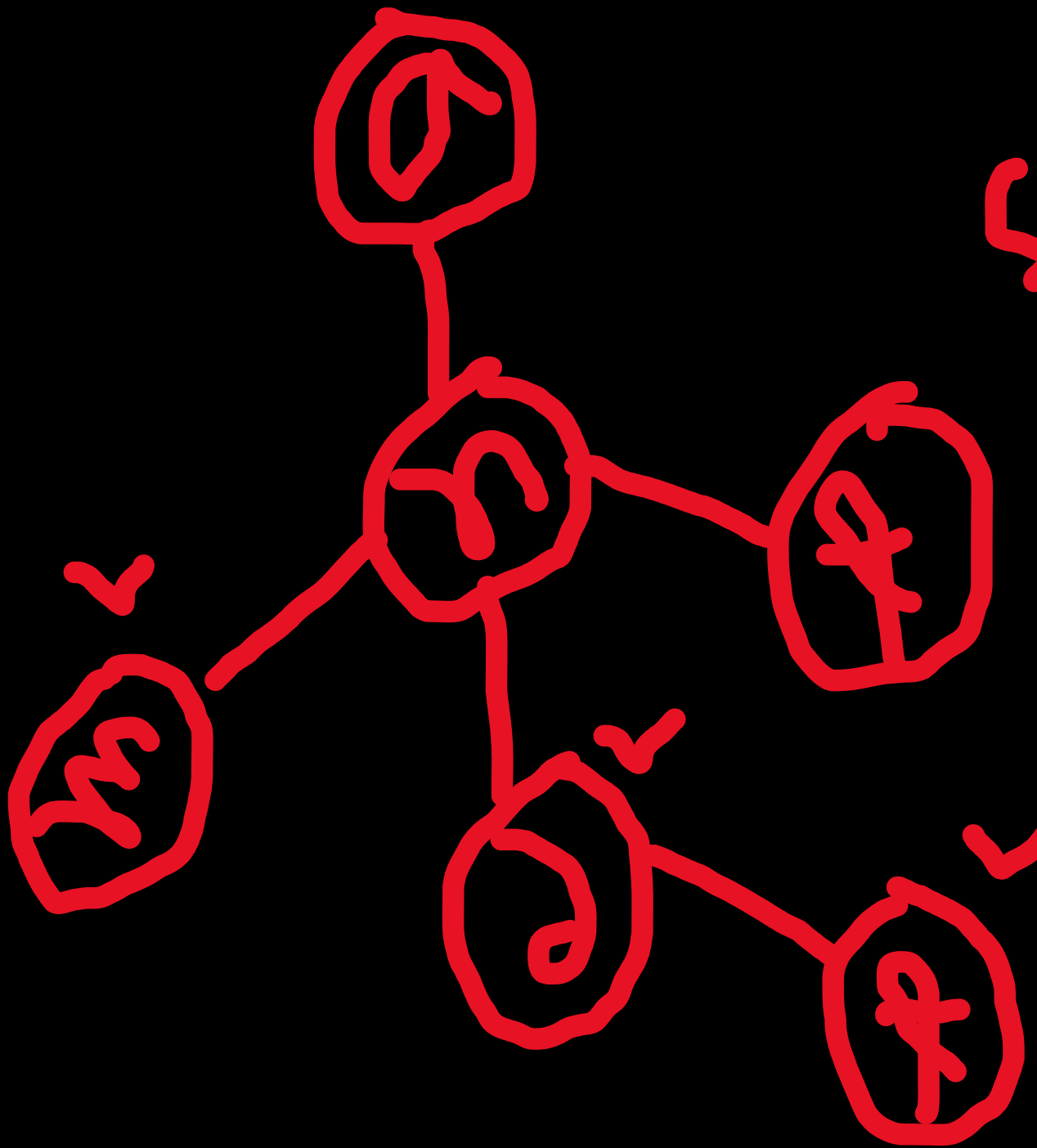
२  
३  
४  
५  
६  
७  
८  
९  
१०  
११  
१२  
१३  
१४  
१५  
१६  
१७  
१८  
१९  
२०  
२१  
२२  
२३  
२४  
२५  
२६  
२७  
२८  
२९  
३०  
३१  
३२  
३३  
३४  
३५  
३६  
३७  
३८  
३९  
४०  
४१  
४२  
४३  
४४  
४५  
४६  
४७  
४८  
४९  
५०  
५१  
५२  
५३  
५४  
५५  
५६  
५७  
५८  
५९  
६०  
६१  
६२  
६३  
६४  
६५  
६६  
६७  
६८  
६९  
७०  
७१  
७२  
७३  
७४  
७५  
७६  
७७  
७८  
७९  
८०  
८१  
८२  
८३  
८४  
८५  
८६  
८७  
८८  
८९  
९०  
९१  
९२  
९३  
९४  
९५  
९६  
९७  
९८  
९९  
१००



Handwritten notes in Hindi, possibly a list of items or a sequence of characters:

१  
२  
३  
४  
५  
६  
७  
८  
९  
१०  
११  
१२  
१३  
१४  
१५  
१६  
१७  
१८  
१९  
२०  
२१  
२२  
२३  
२४  
२५  
२६  
२७  
२८  
२९  
३०  
३१  
३२  
३३  
३४  
३५  
३६  
३७  
३८  
३९  
४०  
४१  
४२  
४३  
४४  
४५  
४६  
४७  
४८  
४९  
५०  
५१  
५२  
५३  
५४  
५५  
५६  
५७  
५८  
५९  
६०  
६१  
६२  
६३  
६४  
६५  
६६  
६७  
६८  
६९  
७०  
७१  
७२  
७३  
७४  
७५  
७६  
७७  
७८  
७९  
८०  
८१  
८२  
८३  
८४  
८५  
८६  
८७  
८८  
८९  
९०  
९१  
९२  
९३  
९४  
९५  
९६  
९७  
९८  
९९  
१००

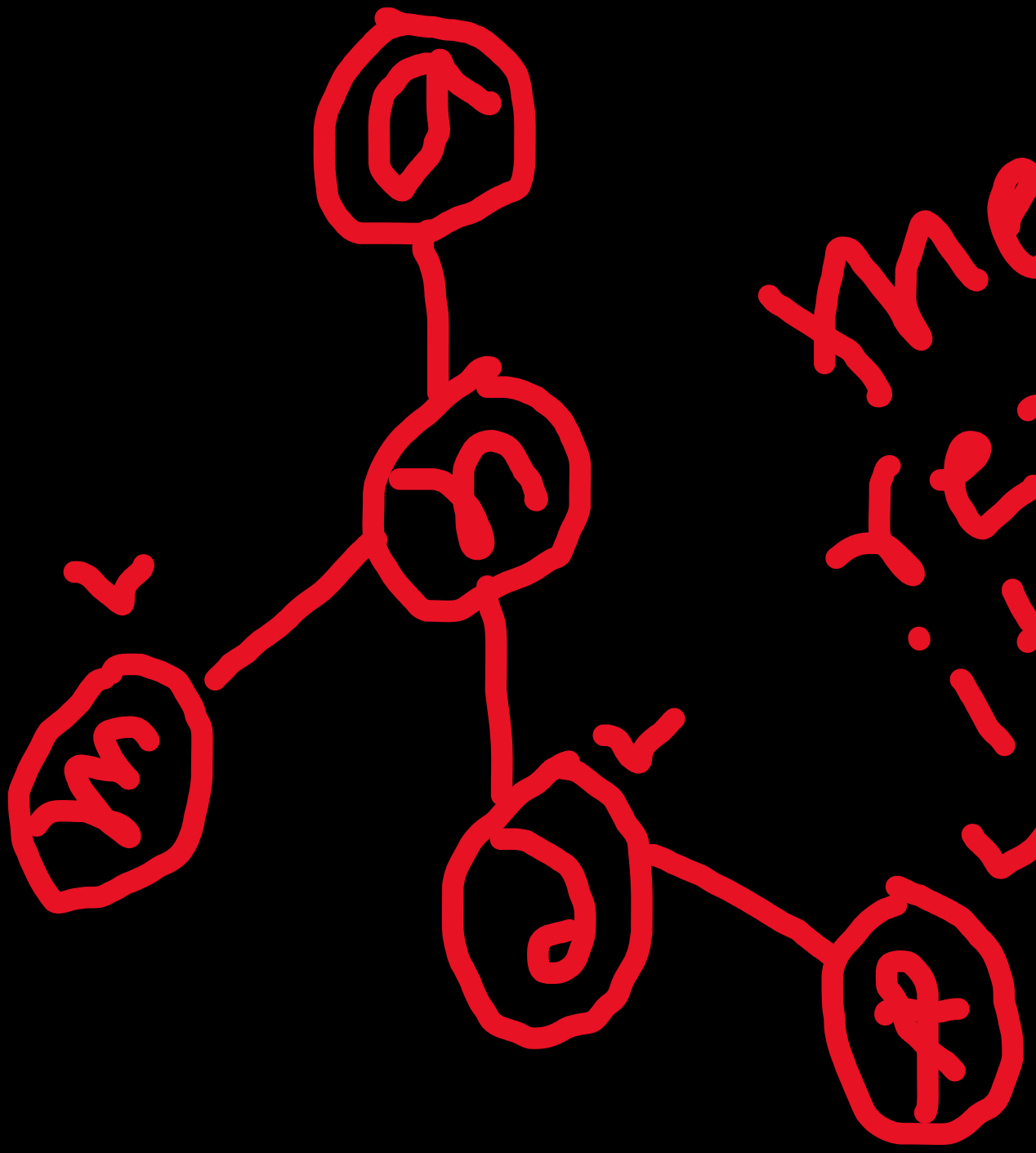
Handwritten notes on the left side of the page, including symbols like  $\sim$ ,  $\times$ , and various letters and numbers.



Handwritten notes on the right side of the page, including symbols like  $\sim$ ,  $\times$ , and various letters and numbers.

Handwritten notes on the left side of the page, organized into three columns:

- Column 1:  $\times$ ,  $9$ ,  $9$ ,  $4$
- Column 2:  $9$ ,  $9$ ,  $9$ ,  $9$
- Column 3:  $3$ ,  $3$ ,  $3$ ,  $3$



Handwritten notes on the right side of the page, organized into two columns:

- Column 1:  $3$ ,  $2$ ,  $2$ ,  $2$
- Column 2:  $3$ ,  $3$ ,  $3$ ,  $3$

Now, let's go to the  
implementations