

There is a new alien language that uses the English alphabet. However, the order of the letters is unknown to you.

apple box abeflopz

You are given a list of strings `words` from the alien language's dictionary. Now it is claimed that the strings in `words` are **sorted lexicographically** by the rules of this new language.

If this claim is incorrect, and the given arrangement of string in `words` cannot correspond to any order of letters, return `""`.

Otherwise, return a *string of the unique letters in the new alien language sorted in lexicographically increasing order by the new language's rules*. If there are multiple solutions, return *any of them*.

**Input:** words = ["wrt", "wrf", "er", "ett", "rftt"]  
**Output:** "wertf"

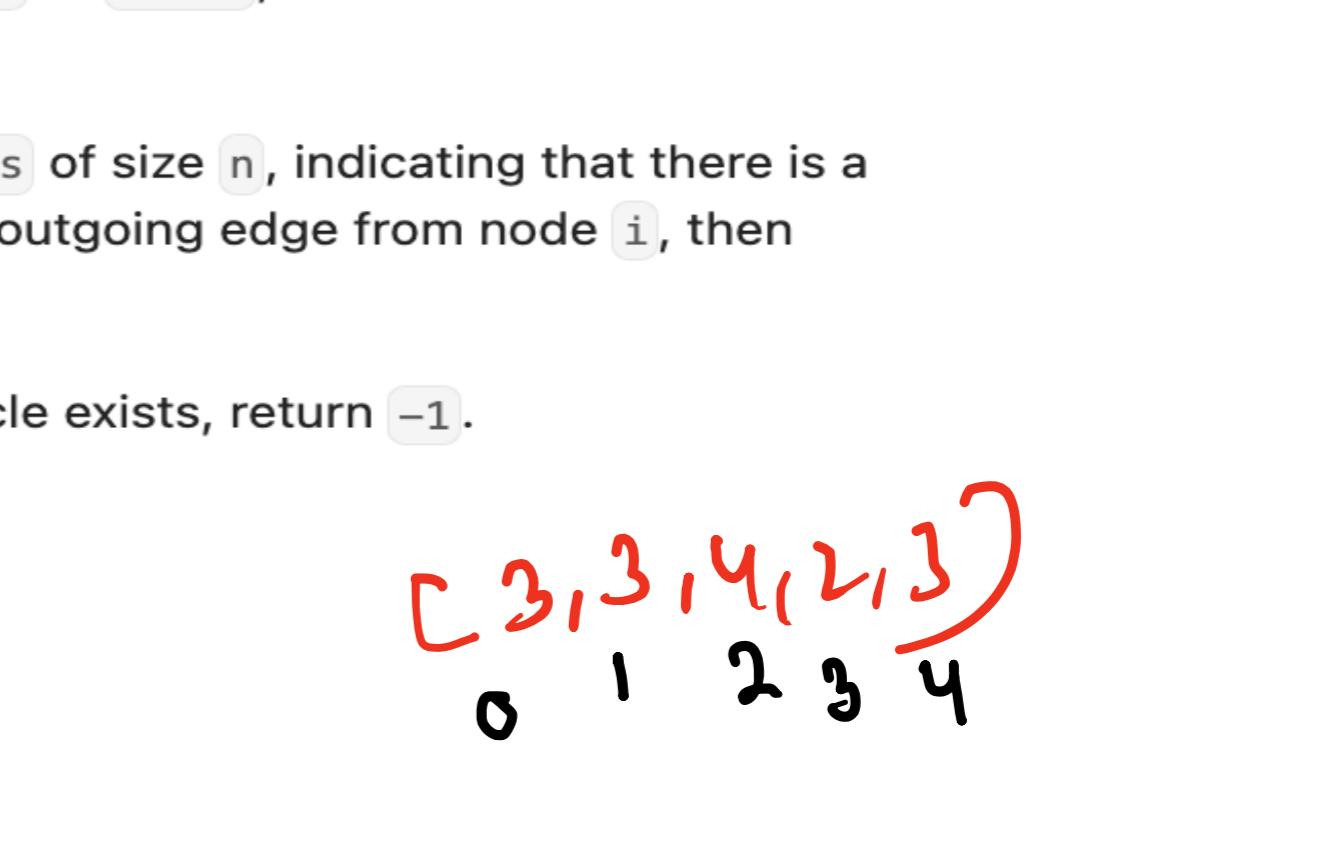
w → e → r → t → f  
 wrt < wrf  
 wrf < er  
 er < ett  
 ett < rftt

**Input:** words = ["z", "x", "z"]  
**Output:** ""

z → x

words = ["wrt", "wrf", "er", "ett", "rftt"]

wrt < wrf  
 wrf < er  
 er < ett  
 ett < rftt



You are given a **directed** graph of `n` nodes numbered from `0` to `n - 1`, where each node has **at most one** outgoing edge.

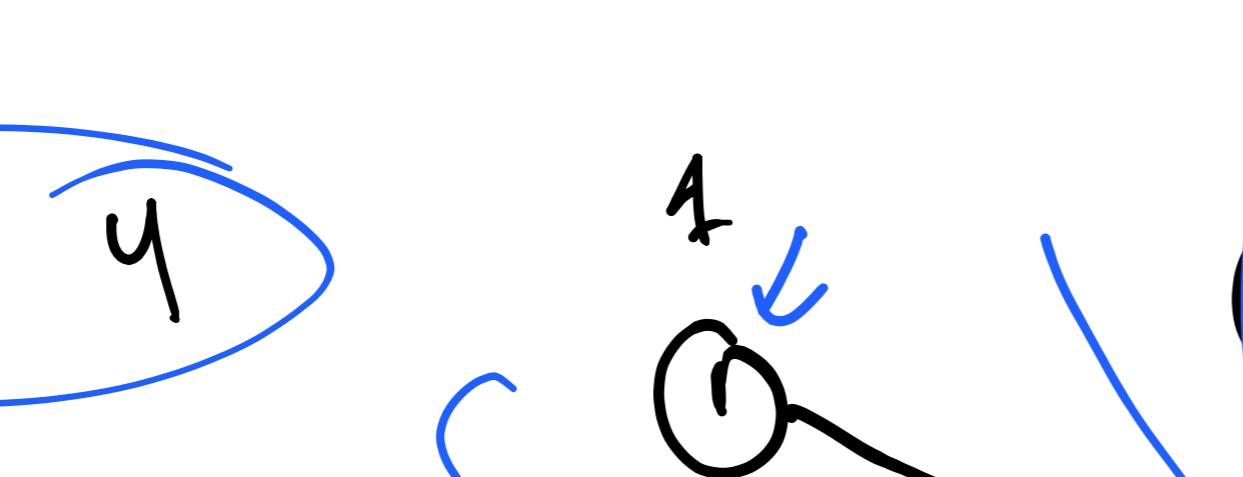
The graph is represented with a given **0-indexed** array `edges` of size `n`, indicating that there is a directed edge from node `i` to node `edges[i]`. If there is no outgoing edge from node `i`, then `edges[i] == -1`.

Return the *length of the longest cycle in the graph*. If no cycle exists, return `-1`.

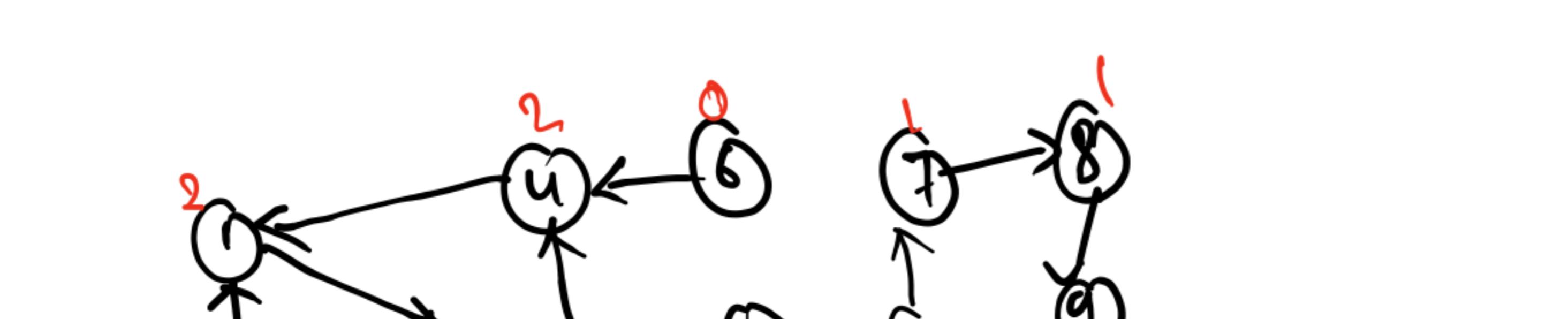
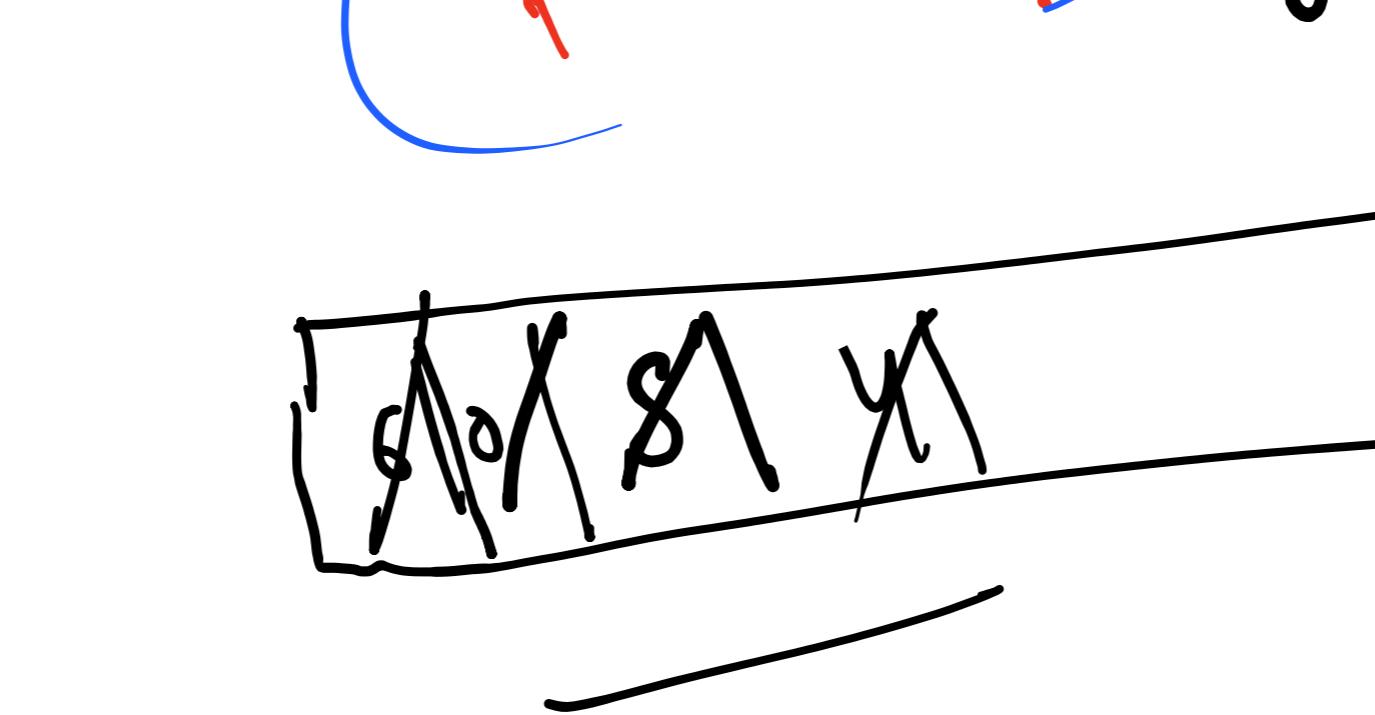
A cycle is a path that starts and ends at the **same** node.

(3, 3, 4, 2, 3)  
 0 1 2 3 4

**Example 1:**



**Input:** edges = [3, 3, 4, 2, 3]  
**Output:** 3



6 0 5 4  
 1 2 3 4 5 6 7 8 9 10

6 0 5 4  
 1 2 3 4 5 6 7 8 9 10

6 0 5 4  
 1 2 3 4 5 6 7 8 9 10

6 0 5 4  
 1 2 3 4 5 6 7 8 9 10

6 0 5 4  
 1 2 3 4 5 6 7 8 9 10

```
int ans=-1;
for (int i = 0; i < visited.length; i++) {
    if(visited[i]==false) {
        int c=1;
        int nbrs=edges[i];
        while(nbrs!=i) {
            c++;
            visited[nbrs]=true;
            nbrs=edges[nbrs];
        }
    }
}
```

6 0 5 4  
 1 2 3 4 5 6 7 8 9 10