P11. Shiritori

[홍원표 21701065]

문제 분석

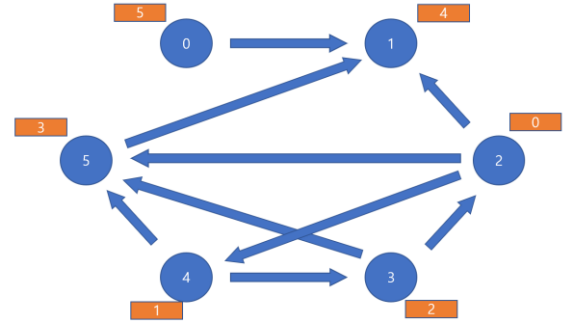
N number of words are given, (3 ≤ N ≤ 1000). If word i's last alphabet matches with word j's first alphabet, there is a connection from i to j. After making all connections, check if N number of words can all be printed using the connections. If possible, print according to the connection order. If not possible, print 0.

문제 풀이

We can create a directed graph using the connections: Change the word into a pair of first alphabet and last alphabet, and check all possible pairs of words to see if the two have connections.

After the graph is created, use BFS on one node to find distances of the rest of the nodes. From the distances of the nodes, if n-1 exists, this means that there is a way to print all words only using connections. Then, using the distances saved on each node, sort them in order and print the index.

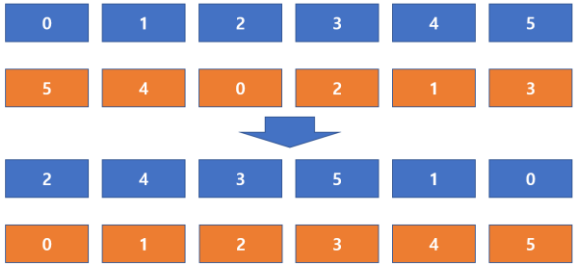
If n-1 doesn't exist, try BFS on another node, looping until BFS is done on all nodes. If n-1 still doesn't exist, this means there is no possible way to print all words using the connections.



문제 풀이 분석 [**Time Complexity: O(N^3)** **Space Complexity: O(N^2)**]

Changing the word into alphabet pair: O(N)
Creating the graph: O(N^2)
BFS on all nodes (worst case): O(N(v+e)) → since v is N, and e can possibly be N^2 → O(N^3)
Print the answer: O(NlogN)

String array for input word: O(N)
First,last alphabet pair array: O(N)
Connection Graph: O(N^2)
distance-saved array: O(N)



Discussion

I couldn't solve this problem, and I believe the reason was my approach on finding the distances, using BFS, since BFS has worst case of O(N^3). I've tried DFS and topological DFS as well, but the time complexity was the same, if not, only increased.