

1.) Time Complexity of algorithm that builds graph

Pseudo Code: for words: $O(n^2)$
nested for loop for words:
if neighbors
vector<vector<int>> → Alist[i].push(j)
Alist[j].push(i):

2.) Time Complexity of determining if a ladder exist

Pseudo Code: for Connected Components: $O(n^2)$
nested for loop for Connected Components:

3.) Time Complexity of finding a path between two words

Pseudo Code: while (!q.empty) // worst entire list n
q.front().neighbors

$n \times n = n^2$ while (All Neighbors !visited) worst all neighbors n
 $= O(n^2)$

4.) Time Complexity of Shortest ladder

Pseudo Code: while (!q.empty()) worst Shortest is the last
n

n x n = $O(n^2)$ for (Neighbors) worst = last index n

5.) Time Complexity of longest ladder

Pseudo Code:

for Words:

n

while (!q.empty())

worst = n

while (!neighbors == Visited) worst = n

n x n x n

$O(n^3)$

→ have to check longest for all
the words $n^2 \times n = n^3$