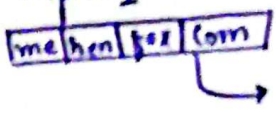
Qasim Ali
20p-0070
BAI-4A

Q#01:

Problem#1

trip-1 Fast to lahore adda



Me + hen

trip-2 Lahore adda to Fast

me | corn | fox

me, hen

← me

trip-3 Fast to Lahore add.

me | corn | Fox

me, hen, fox

me + Fox

trip-4 Lahore adda to Fast

corn | hen | me

me, hen, Fox

← me + hen

trip-5 Fast to Lahore adda

corn | hen | me

me + corn

me, corn, Fox

trip-6 Lahore add to fast

hen | me

me, corn, Fox

← me

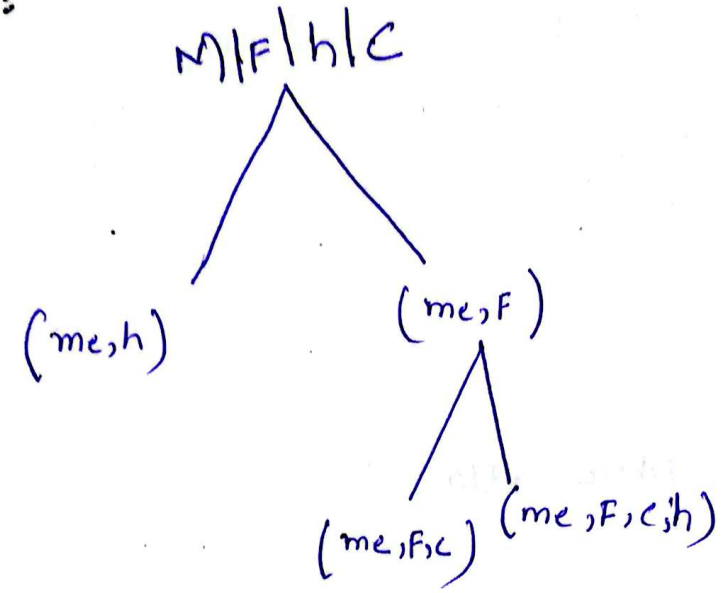
trip-7

hen | me

me + hen

hen, corn, fox, me

Solution:



moves:

M-1 Take hen
M-2 return alone
M-3 take fox
M-4 Return with hen
M-5 Take corn
M-6 return alone
M-7 Take hen.

We had a total of 7 trips.

BFS guarentees the shortest station when not checking for repeated states.

→

Part #01

1- State Representation

2: Initial State = SUV

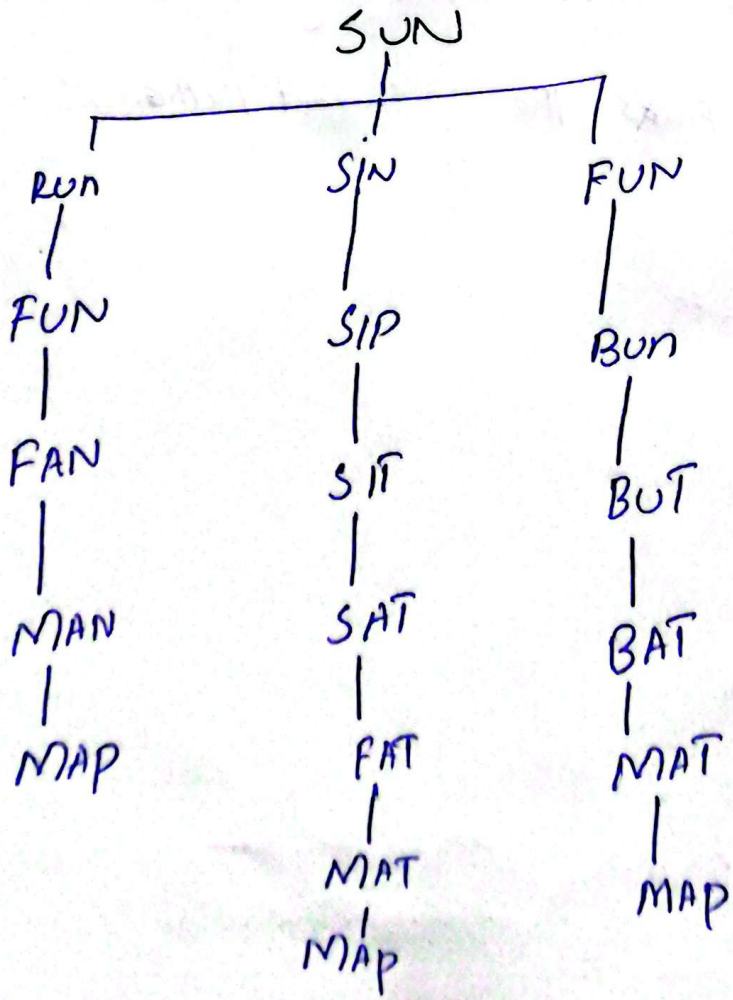
3: Successor Function: Changing one letter at a time in such a way the new word is valid.

4: path Cost Function:

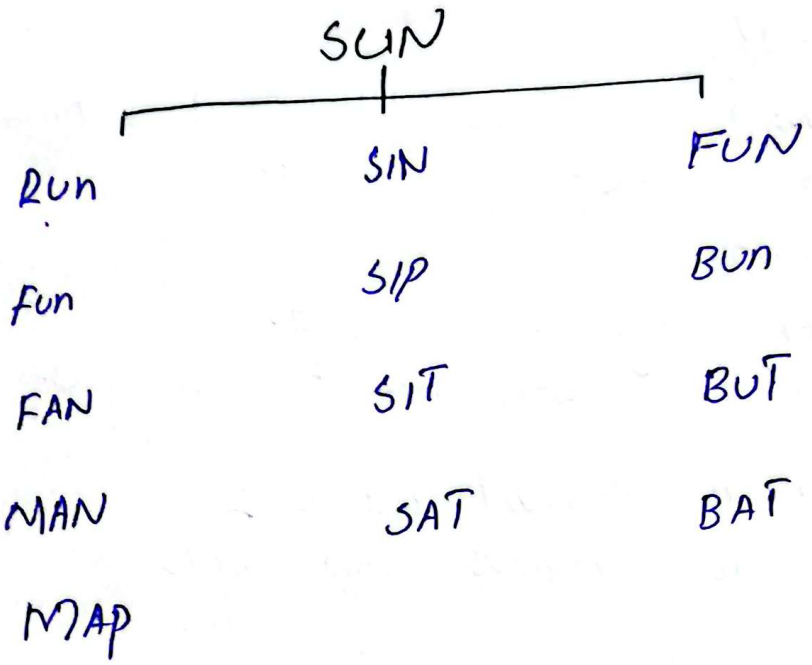
Each letter Change has a cost.

- if a letter is ~~consnt~~ consonant cost is 3.
- if a letter is consonant cost is 1.
- Returning to previous word adds 2

Part #02



1- Breadth First Search :

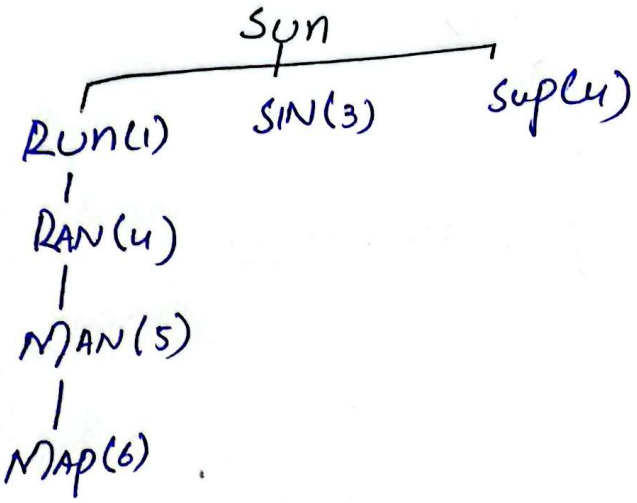


- BFS explores the trees level by level
- Yes BFS Mostly finds the Shortest paths.

2- Depth First Search :-

- DFS explores one branch deeply.
- DFS can stuck in loops.

3. Uniform cost search

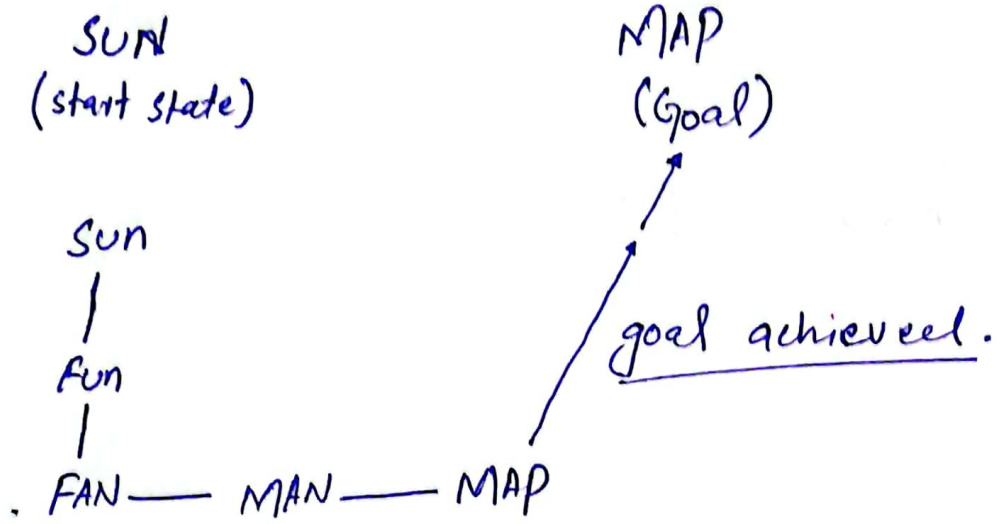


- Yes UCS guarantees the optimal path.

→

Q# 02

Problem #02



Changing one letter at a time and the transformed words must be valid.

The END