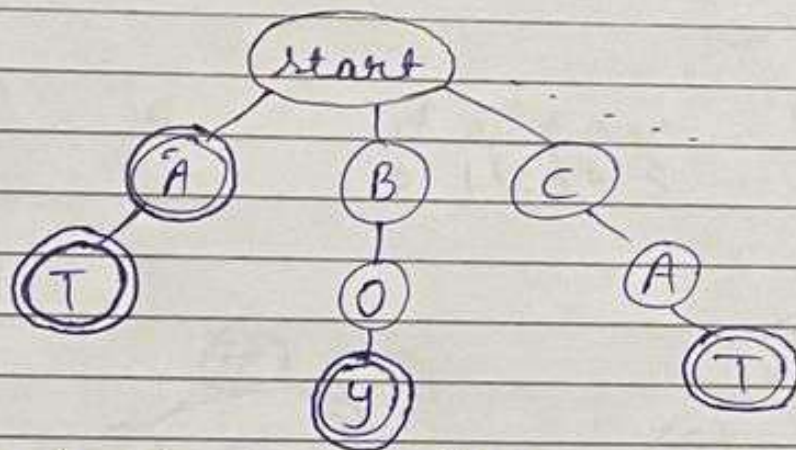# * Word Composition problem *

1) The given question is similarily like dictionary so, the tree data structure will work much better than other data structure. where trie looks like

Where the double circled letters tells that a word is terminating there, So the words in trie are (A, AT, BOY, CAT)

2) So we will start by reading input by ~~lines~~ using scanner.

3) Lets dryrun the code for Input-01.txt where it has

cat
cats
catsdogcats
dogs
dogcatsdog
hippopotamuses
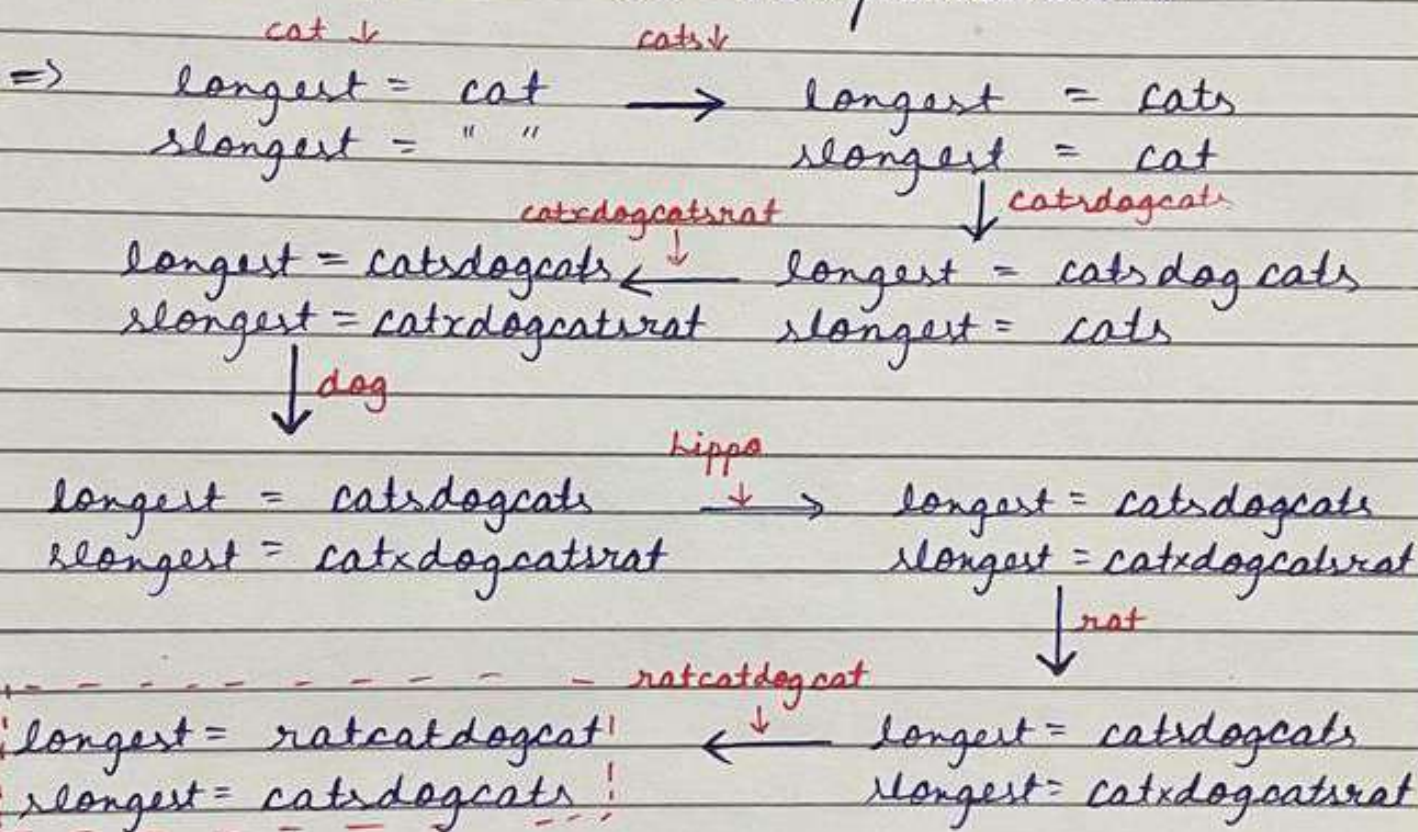rat
ratcatdogcat

⇒ Data will be inserted in the trie

⇒ For each word in the file, the code finds suffix and adds in pair class

for e.g "catsdogcats" suffixes like "cats" and "dogcats" and , pairs will be

  ∴) (catsdogcats, dogcats)
  :) (catsdogcats, cats)
Similarily, it finds "cats" in trie and stored in compound word.

⇒
cat ↓
longest = cat
slongest = " "
→
cats ↓
longest = cats
slongest = cat

catsdogcatsrat
longest = catsdogcats ←↓ longest = catsdogcats
slongest = catsdogcatsrat   ↓catsdogcats slongest = cats

↓ dog

longest = catsdogcats
slongest = catsdogcatsrat
—↓ Lippo →
longest = catsdogcats
slongest = catsdogcatsrat

↓ rat

ratcatdogcat
longest = ratcatdogcat ←↓ longest = catsdogcats
slongest = catsdogcats   slongest = catsdogcatsrat

So output will be
  Longest compound word = ratcatdogcat
second longest compound word = catsdogcats