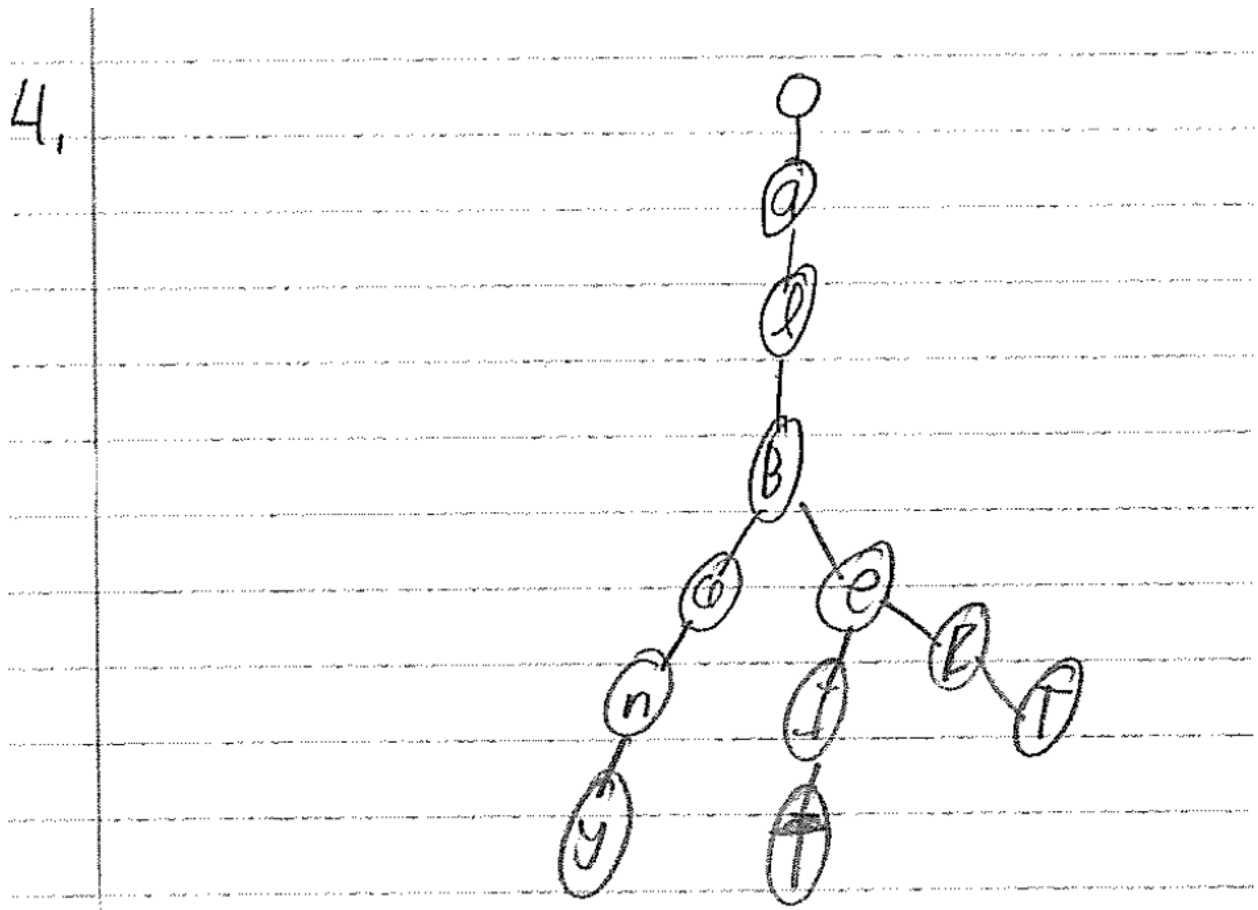
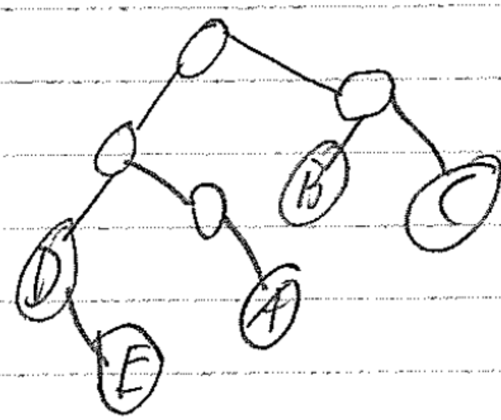
HW 5
CS 251
Dylan Mackey

1. Smallest possible strings: a, d, aa, bb, ab, ba. All of these can be represented in the regular expression except ba, **so “ba” is the answer.**
2. b*ab*ab*. This will allow for any amount of b’s to be in the string, from 0 to infinity. However there has to be exactly two a’s in the string.
3. D) 540. In the worst case scenario, each node will have 36 children, one for each letter of the alphabet. The word we are looking for has 15 letters, so 36*15 = 540.
4. B) 12

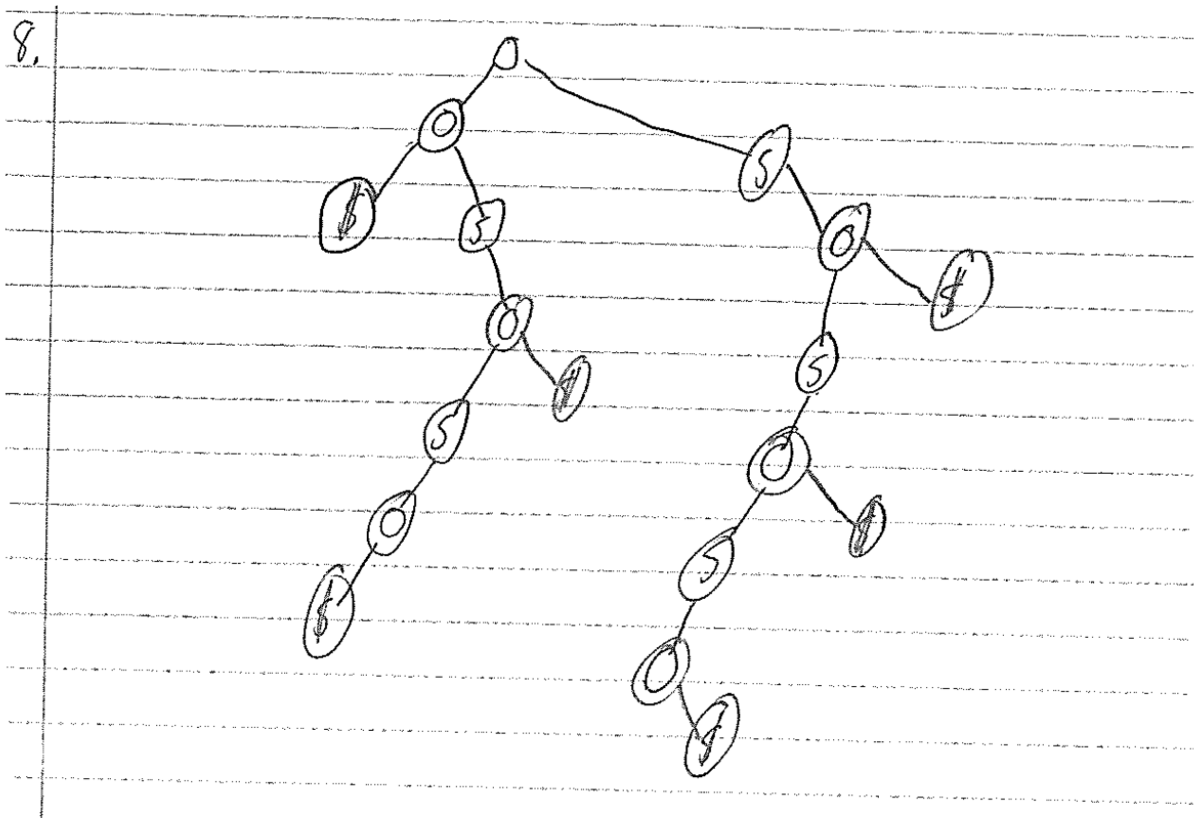


5. No, because D would be on internal node, but only leafs can store characters.

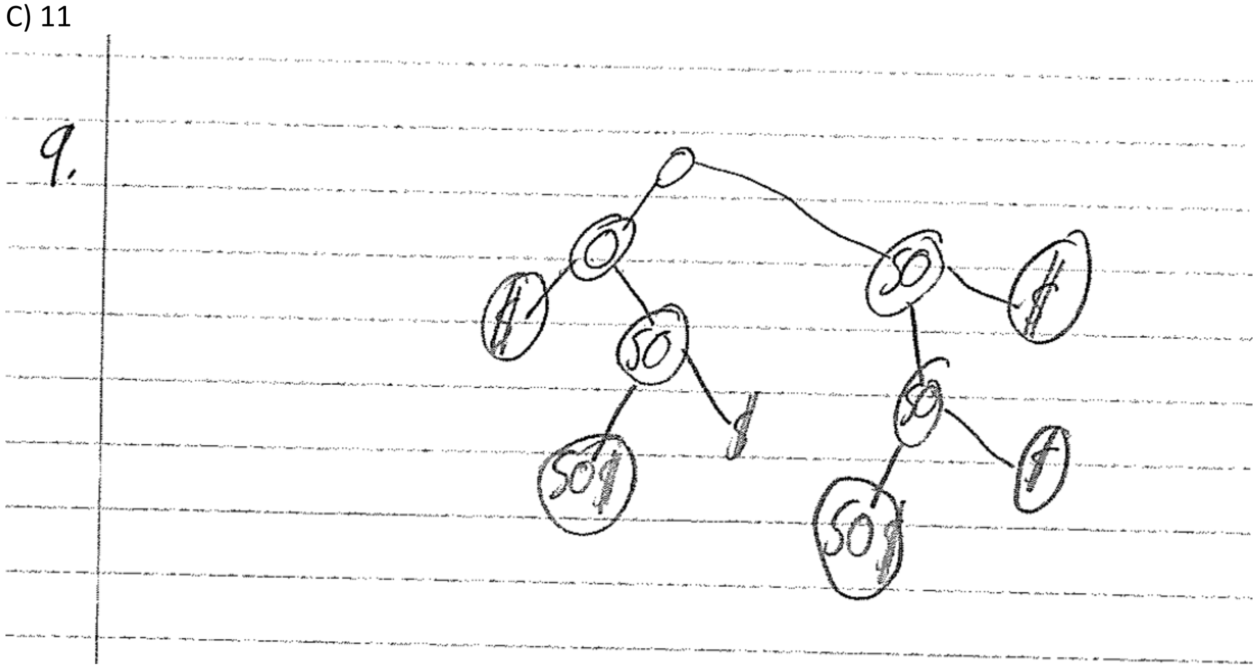
5,



6. A) True. Since the subproblems overlap, we don't use recursion. Instead, we construct optimal subproblems from bottom up
   B) False. The subproblems overlap.
   C) True. Dynamic programming uses the adjacent answers in order to solve its problem.
   D) True. As shown in A, we divide the problem into subproblems.
7. A) True. By dividing the program to a base case, it goes bottom up.
   B) True. When dividing and conquering, you break down each subproblem to a base case, and then build back up. Thus, being able to be solved independently.
   C) False. Since solution can only come from the two previous subproblems, and not any other solutions, then it is not locally improved.
   D) True. By definition, it is divided into subproblems.
8. C) 18



9. C) 11



10. Tries do not search through the entire text like BM/KMP, but rather search for a word already in a tree. Therefore, it does not have to search through the majority of characters looking for a match.