CS 213
Mid-Semester Exams Solutions
2021

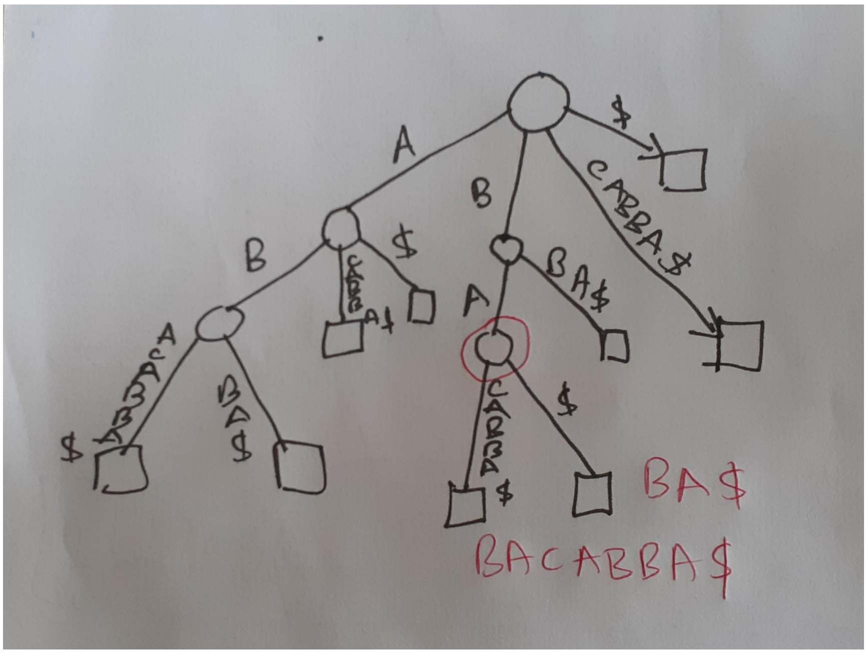
Time: 1 hour, 55 minutes.
Marks=[4+5+2+2+6+6+5]=30 marks.

Instructions: Please make sure that the images you load are clean, well oriented and have the Question Number and no other work on it. Marks are reserved for this. For code snippets, please add in-line comments. If the TA finds the solution poorly written or presented, he/she is authorized to deduct 1 mark per question.

1. *Build the compacted (i.e., with no nodes with one child) suffix tree for the word “ABACABBA$”. Show the edge labels clearly. Draw a square box around nodes where words terminate.*

   Locate the string “BA” in this tree. Mark it with a STAR. Use the tree to identify and list the suffixes where “BA” appears.

   Submit the final tree as a single image as above. [4 marks]



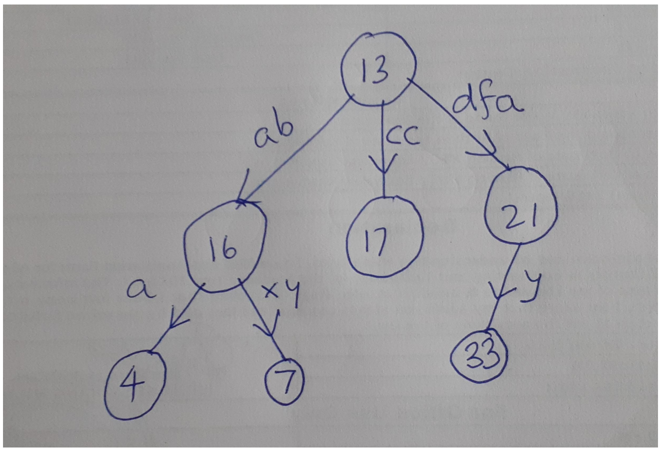
For the Tree:
0 - Incorrect tree
1 - Partially correct tree
2 - Correct tree
Many may/may not use the $ to make a single edge branch from root. Both are fine.

Locating BA and reading out the suffixes:
0 - No correct suffix
1 - one correct suffix
2 - Both correct

2. *A trie node is represented in the following structure.*

   trienode.Label
   trienode.NumberOfChildren (say k) (integer)
   trienode.PointersToChildren[1,...,k] (array of pointers)
   trienode.PointersToEdgeLabels[1,...,k] (array of strings)

   Thus for example, node 16 below will have:
   trienode.Label=16
   trienode.NumberOfChildren=2
   trienode.PointersToChildren=[p4,p7]
   trienode.PointersToEdgeLabels=[a,xy]



Write a recursive pseudo-code snippet for S=Prefix(T) which produces a list of all possible prefixes of words which are stored in T. Write 1-line descriptions of all functions used, before you begin with the main program. Apply the code for the tree above starting at node 13. Execute only the first level of recursion, and show the output lists returned by the first calls and the final answer. Attach a single picture with the code snippet and its execution written below the code. [5 marks]

Functions used:

C=Concat(A,B): Given two sets of strings, for every staring x in A and y in B, put z=xy in C.
S=SPrefix(x): given a string x, produce the set S of all prefixes of x.
U=Union(S,T): union of the sets of strings S and T

Main code:

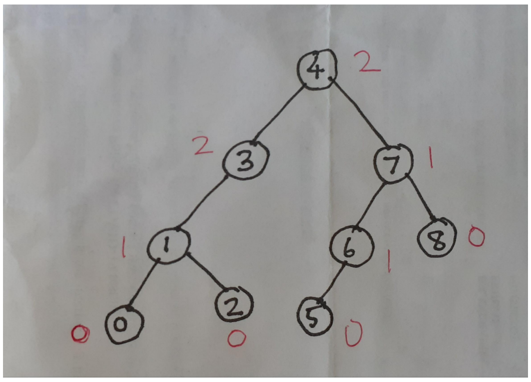
```
Function S=Prefix(T)
k=T.NumberOfChildren.
S={};
For i=1:k
    x=T.PointertoEdgeLabel[i];
    Ti=T.PointerToChildren[i];
    Si=Prefix(Ti);
    S=Union(S,Concat(x,Si));
    S=Union(S,SPrefix(x));
End;
Endfunction.
```

Functions descriptions 1 marks
Function 2 marks

End of k	1	2	3
x	ab	cc	dfa
SPrefix(x)	ab,a	c,cc	dfa,df,d
Si	a,x,xy	-	y
S	aba, abx,abxy,ab,a	aba, abx, abxy, ab, a, c,cc	aba, abx, abxy, ab, a, c, cc,dfay, dfa,df,d

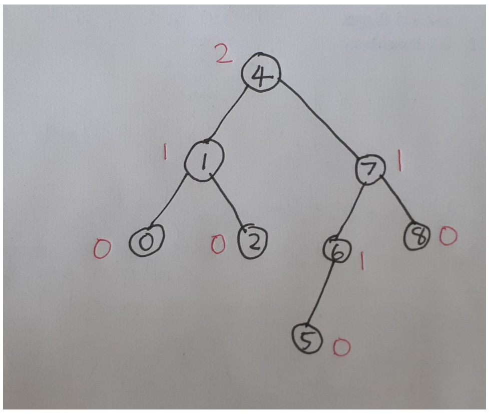
Table above in some form: 2 marks.

3. *For every node v in a tree T, define low(v) as the length of the shortest path from v to a leaf node. Consider the binary search tree T below. Label each node v with its low(v). Submit image. [2 marks]*



0 or 2 marks.

4. *Delete the node “3” and compute the new BST and the new low values. Submit image. [2 marks]*



0 or 2 marks.

5. *For a pattern P[1...n], define h(i) as the smallest 1<=k<=i-1 such that P[1]=P[k+1],..., P[i-k]=P[i] and P[i-k+1]!=P[i+1]. If no such match can be found, we assign h(i)=i.* ***Note that this definition is as in the tutorial problem (and different from Prof. Garg’s notes).*** *Fill in the table as below. [4 marks]*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
P	x	y	x	x	z	x	x	y	y
i=1	x	**y**							
h(i)=1									
i=2	x	y	**x**						
h(i)=2									
i=3	x	y	x	**x**					
h(i)=2			x	y					
i=4	x	y	x	x	**z**				
h(i)=3				x	**y**				
i=5	x	y	x	x	z	**x**			

h(i)=5									
i=6	x	y	x	x	z	x	**x**		
h(i)=5						x	**y**		
i=7	x	y	x	x	z	x	x	**y**	
h(i)=7							x	**y**	
i=8	x	y	x	x	z	x	x	y	**y**
h(i)=6							x	y	**x**
i=9	x	y	x	x	z	x	x	y	y
h(i)=9									

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
P	x	y	x	x	z	x	x	y	y
h(i)	1	2	2	3	5	5	7	6	9

Marks :

- ***0 - mostly incorrect values***
- ***2 - Half the values are correct***
- ***4 - Almost all values correct***

Explain in 2 lines, how did you use the presence of z in the 5th location? [2 marks]

Since z appears at the 5th place, we have the following 3 cases:
i<5: no change
i=5 condition will not be satisfied, so h(i)=5
i>5 h(i) must be greater than or equal to 5

Marks :

- ***0 - No explanation given/Filled wrong values for i=5 in the table***
- ***1 - Partially correct explanation***
- ***2 - Almost correct explanation***

6. *There is an employment portal MyJobi.Com in which you can post your profile. It then collects all the offers and sends you one offer O(i) every day. You must either accept the offer or reject it. If you accept, the game stops. If you reject, you will get another offer O(i+1) the next day. Consider the following algorithm which aims to accept a job offer, given offers O[1,..,n]*

hold=O(1); select=0;

```
For i=2:n
    If O(i)>hold;
      select=O(i);
      Break;
    End;
End;
```

Thus, the strategy is to accept the first offer better than the offer on Day 1. Note that you may end up not accepting any offer, .i.e., end up with select=0

Assume that O is a random permutation of {1,2,...,n}. Let p[i] be the probability that select=i. Then
What is p[0]? [2 marks]

P[0] is the probability that you reject everyone. That is when the first job was n. Thus the probability is 1/n.

[0,2] marks.

What is the probability that the game ends on Day 2? [4 marks]

This happens when the second number i is higher than the first i. The total ways are n(n-1). The ordering required gives us n(n-1)/2. Thus, probability is 1/2

[0,2,4] marks. 2 marks for correct answer. 4 for correct reasoning.

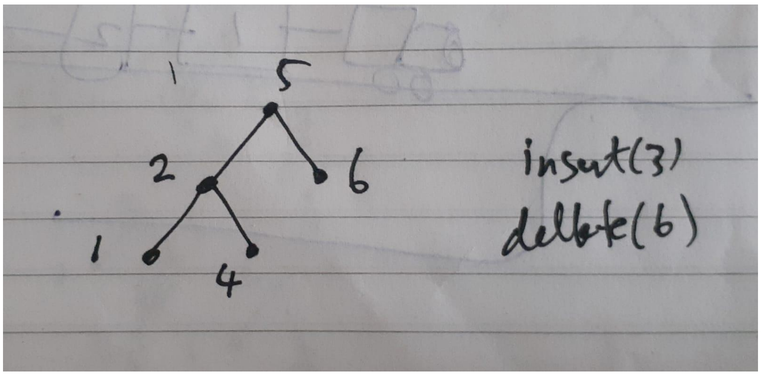
Can you generalize? Try Day 3. That is a nice number. [0 marks]

7. *Let T be an AVL tree such that the root value is b. There are three values a<b<c such that a is NOT in the tree, while c is in the tree. Suppose we do (A) insert (a) followed by delete(c) on T to get tree TA, OR (B) delete(c) followed by insert (a) on T to get TB. Will TA be the same as TB?*

   Answer YES/NO with 4 lines of explanation. [5 marks]

   In general, NO, during insertion or deletion, the rotates may affect the root node, In this case, it depends on how you have implemented the rotates. See the example below.

   YES: with some explanation (without considering rotation) 1 mark.
   YES: with explanation containing the case of rotation, but arguing that it will not matter 2,3 marks.
   NO: without explanation 1 mark.
   NO: with mention of rotations: 2,3 marks.
   NO: with example and argument. 5 marks.



insert(3)
delete(6)