COMP3506/COMP7505—Algorithms and Data Structures

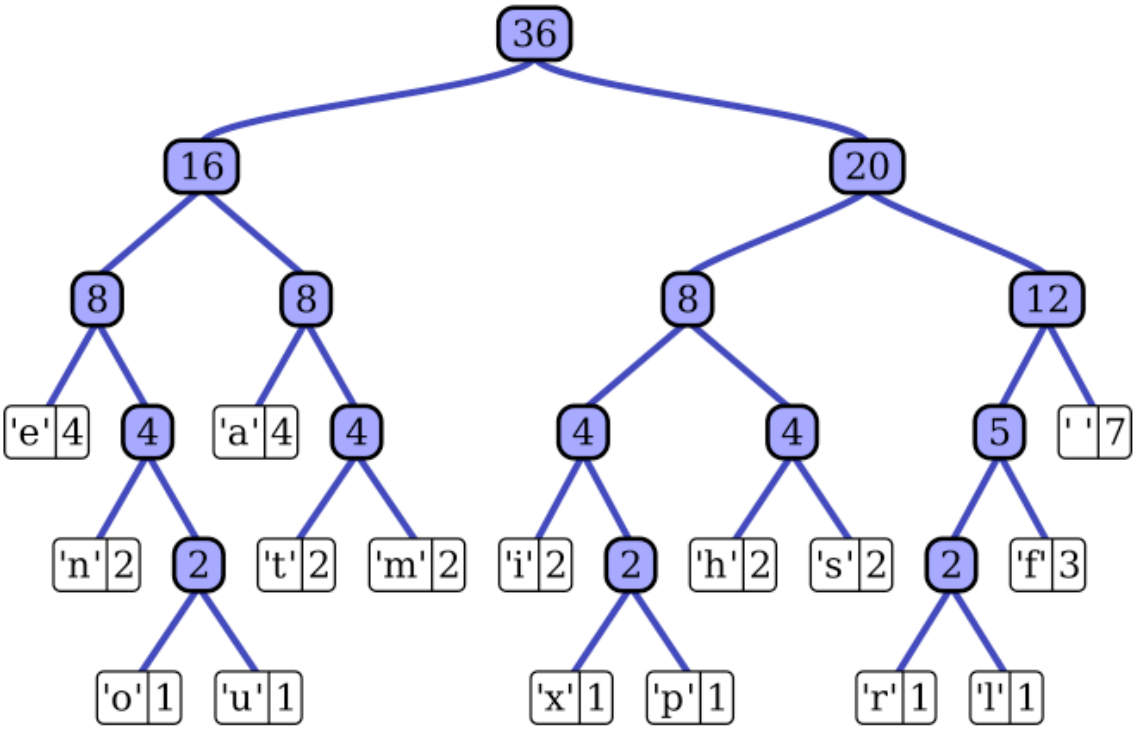
School of Information Technology and Electrical Engineering

Week 12 Tutorial Sample Solutions

Question 1.

Below is a Huffman tree generated from the exact frequencies of the text

`"this is an example of a huffman tree"`



Describe the steps involved in creating a Huffman tree from text as exemplified above. In your answer explain what the labels on the nodes of the tree mean.

(1) Compute the freq for each character in text (this is the number with each label on the external nodes)
(2) Initialize a PQ (or list but then performance may be worse)
(3) Insert into PQ each char as a single node binary tree annotated with the freq
(4) Main loop: take the two trees with smallest freq out of PQ, and add the concatenation of the two back in (until only a single tree is left)
Each node of the tree holds the count of occurrences (i.e. freq) of the chars that are positioned below it.

Using the Huffman tree above, determine the binary code and count the number of bits required to transmit the text "`fill pool`" (do not forget the space separating the words).

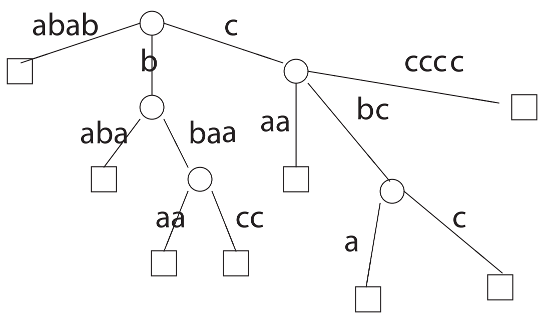
CODE= 1101 1000 11001 11001 111 10011 00110 00110 11001
#BITS= 41

Question 2.

Draw a *compressed* trie for the following set of strings:

`{abab, baba, ccccc, bbaaaa, caa, bbaacc, cbcc, cbca}`

Solution



What is the longest prefix of the string "`cgtacgttcgtacg`" that is also a suffix of this string?

Solution

`cgtacg`

Question 3.

Draw a figure (similar to that in lecture) illustrating the comparisons done by Knuth-Morris-Pratt pattern matching algorithm for the text "aaabaadaabaaa" and pattern "aabaaa". Also show the last failure function table.

j	0	1	2	3	4	5
P(j)	a	a	b	a	a	a
F(j)	0	1	0	1	2	2

j = *F*(*j* – 1)

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
a	**a**	**a**	**b**	**a**	**a**	**d**	**a**	**a**	**b**	**a**	**a**	**a**
a **1**	a **2**	~~b~~ **3**	a	a	a			*j* = *F*(2 – 1) = 1				

	a	a **4**	b **5**	a **6**	a **7**	~~a~~ **8**		$j = F(5 - 1) = 2$				
				a	**a**	~~b~~ **9**	a	a	a	$j = F(2 - 1) = 1$		
					a	~~a~~ **10**	b	a	a	a	$j = 0$	
$j == 0$, so $i = i + 1$						~~a~~ **11**	a	b	a	a	a	$j = 0$
							a 12	**a 13**	**b 14**	**a 15**	**a 16**	**a 17**

Question 4

Fill in the following table and show the possible longest common sub-sequences.

	-1	G	A	T	T	A	C	A
-1	0	0	0	0	0	0	0	0
A	0	0	***1***	1	1	***1***	1	***1***
C	0	0	1	1	1	1	***2***	2
A	0	0	***1***	1	1	***2***	2	***3***
T	0	0	1	***2***	***2***	2	2	3
T	0	0	1	***2***	***3***	3	3	3
A	0	0	***1***	2	3	***4***	4	***4***
G	0	***1***	1	2	3	4	4	4

Refer to the following excerpt, from GT Ch 12, p572.

> Given the table of $L[i,j]$ values, constructing a longest common sub-sequence is straightforward. One method is to start from $L[n,m]$ and work back through the table, reconstructing a longest subsequence from back to front. At any position $L[i,j]$, we can determine whether $x_i == y_j$. If this is true, then we can take x_i as the next character of the sub-sequence (noting that x_i is ***before*** the previous character we found, if any), moving next to $L[i-1, j-1]$. If $x_i != y_j$, then we move to the larger of $L[i, j-1]$ and $L[i-1,j]$. (See Figure 12.13.) We stop when we reach a boundary cell (with $i == -1$ *or* $j == -1$). This method constructs a longest common subsequence in $O(n + m)$ additional time.

	-1	G	A	T	T	A	C	A
-1	0	0	0	0	0	0	0	0
A	0	0	***1***	1	1	***1***	1	***1***
C	0	0	1	1	1	1	***2***	2
A	0	0	***1***	1	1	***2***	2	***3***

T	0	0	1	***2***	***2***	2	2	3
T	0	0	1	***2***	***3***	3	3	3
A	0	0	***1***	2	3	***4***	4	***4***
G	0	***1***	1	2	3	4	4	4