Assignment #4

Due on Thursday 2/28/2019

1. We would like to use the Huffman Coding algorithm to find an optimal coding scheme for the following characters with the corresponding frequencies

c	$c.freq$
A	75
B	11
C	39
D	30
E	146
F	33
G	48
H	41

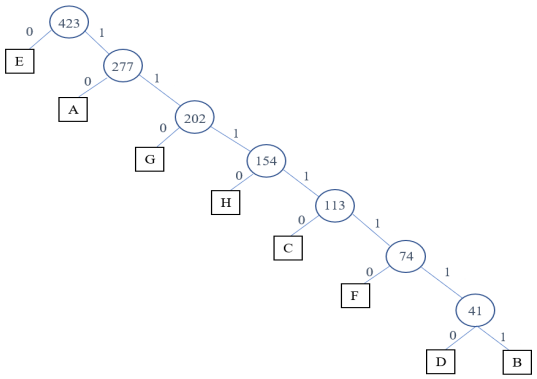
   (a) Construct the Huffman Tree according to the table above.

   (b) Find prefix codes for the characters A through H.

   (c) Use the codes found in (b) to encode the message "HEADACHE".

   (d) Is the encoding you computed in (c) optimal? Why or why not?

Answer: (a)



(71).png (71).png

(b): Letter code

A	10
B	1111111
C	11110
D	1111110
E	0
F	111110
G	110
H	1110

(c): 1110 0 10 1111110 10 11110 1110 0

(d):Yes it is optimal, becuase we put the most frequent number at the place would cost us least space to code.

2. In the *longest common subsequence* algorithm we discussed in class, we formulated the recursive formula based on *prefixes* of the two inputs, i.e., $X[1..i]$ and $Y[1..j]$.

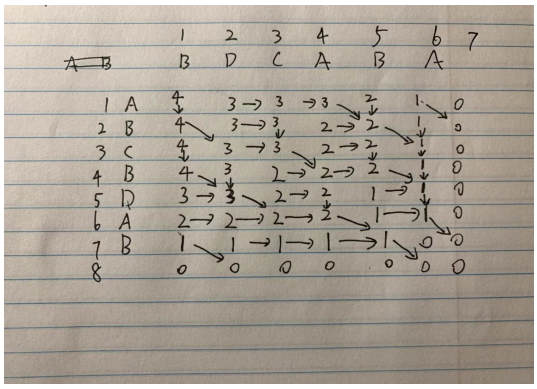
   (a) Rewrite the recursive formula using *suffixes* instead of prefixes, i.e., $X[i..m]$ and $Y[j..n]$.
   Answer:

$$LCS(X[i..m], Y[j..n]) = \begin{cases} 0 & \text{m=0,n=0} \\ LCS(X[i+1..m], Y[j+1..n]) & \text{if } X_i = Y_j \\ max[LCS(X[i+1..m], Y[j..n])LCS(X[i..m], Y[j+1..n])] & \text{if } X_i \neq Y_j \end{cases} \quad (1)$$

   (b) Develop a bottom-up dynamic programming algorithm based on the recursive formula in (a). Describe the algorithm and write a pseudo code.
   Answer: pseudo code:
   LCS(X,Y)
   if m=0 or n=0
   then return 0
   if X[i]=Y[j]
   then return LCS(X[i+1..m],Y[j+1..n])
   if X[i] $\neq$ Y[j]
   then return$max[LCS(X[i+1..m], Y[j..n]), LCS(X[i..m], Y[j+1..n])]$

   Algorithm: First if one array has size of 0, we just return 0. Then, We compare two first letters X[i] and Y[j] in each array. If they are the same, we reduce the size for both array by one. If they are not same, we would try to return the function with greater value between LCS(X[i+1..m],Y[j..n]) and LCS(X[i..m],Y[j+1..n])

(c) Use the algorithm developed in (b) to solve the problem for the inputs $X = \langle A, B, C, B, D, A, B\rangle$ and $Y = \langle B, D, C, A, B, A\rangle$. Show the final tabulation of the answer showing both the length of the answer and the arrows similar to Figure 15.8 on page 395 of the textbook.



3. *Longest increasing subsequence:* Given an input sequence $X = \langle x_1, x_2, ..., x_n \rangle$ of numbers, an *increasing subsequence* of X is a subsequence where the values are increasing in value. We would like to find the *longest increasing subsequence*. For simplicity, assume that all the numbers in X are distinct.

   (a) Develop a recursive formula for the problem above and prove the correctness of the formula, i.e., show that it yields an optimal answer.
   Answer:

$$LIS(X[1..i]) = \Big\{ max[LIS(X[1..j) + 1 \quad \text{where } 0 < j < i, X[j] < X[i] \tag{2}$$

   (b) Write down a pseudo code for a top-down recursive algorithm using the formula developed in (a).
   Pseudo code:

```
function LIS(X[1..i])
  for m = 1 to i do
    length[m] = 1
  for j = m+1 to i do
    if X[j] > X[m] then
        length[j] = max[length[j], length[m]+1]
  return max_{1≤j≤n} length[j]
```

(c) Describe a bottom-up dynamic programming iterative algorithm for the problem. Write down the pseudo code and establish its running time.
Description: We would have one array which is used to keep track of the length of longest increasing sub-sequence for each element in the original array. First, we would set the length as 1 for every number in the array. Starting from the first number, X_1, because X_1 is the first number, length for X_1 would not be changed. Then we look at the second number. As long as there is a number before X_2 which is less then X_2, the length for X_2 would be changed to the length for the number we just found and plus one. Then we repeat the process till the end of array. At last, we would return the greatest length among all the number.

The run-time for this algorithm is $\theta(n^2)$

(d) Use your bottom-up algorithm to solve the problem for the input $X = \langle 13, 36, 40, 41, 19, 43, 37, 10 \rangle$. Show the final tabulation and write down the final answer.

(72).png (72).png

Number	13	36	40	41	19	43	37	10
Length	1	2	3	4	1	5	3	1

The final answer is 5