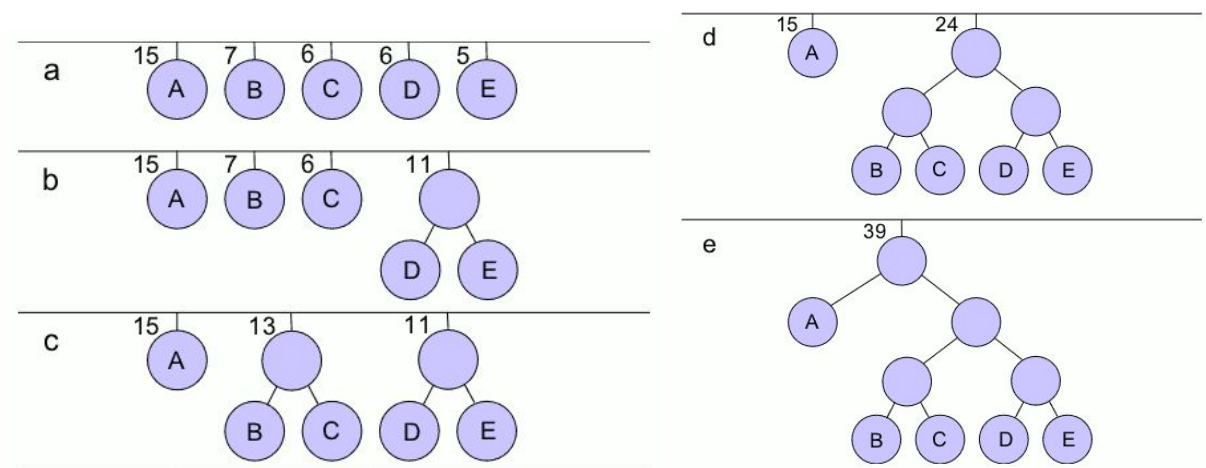
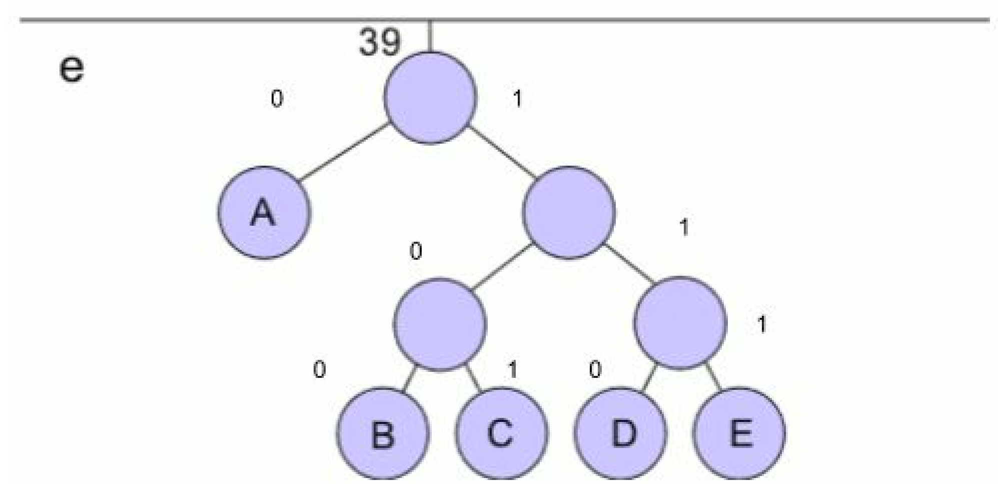
Question 1: Huffman Encoding Proof

Prove:
For two symbols A and B with probabilities P(A) >= P(B), then in the resultant representation sequence according to Huffman encoding procedure, the length of symbol A is no longer than that of symbol B.

Assume:
$H(A_i, P_i)$: *is an optimal Huffman coding tree*
$P(A) = \{p_1, p_2, \ldots, p_i\}$ *Probability/weight of a symbol in the Alphabet occurring in a word for a given tree.*
$A = \{a_1, a_2, \ldots, a_i\}$



$A = \{A, B, C, D, E\}$, $P(A) = \{15, 7, 6, 6, 5\}$



Optimal Tree H

The binary representation of each given symbol is determined by the path that must be traversed to get to the leaf node.

A	B	C	D	E
0	100	101	110	111

The length of a symbols binary representation is determined by the depth at which the leaf nodes lies in the optimal tree. In example by examination of tree H we can see.

$P(A) >= P(B) \; : \; 15 \geq 7 \; \textit{and the corresponding lengths are} \; 1 \leq 3$

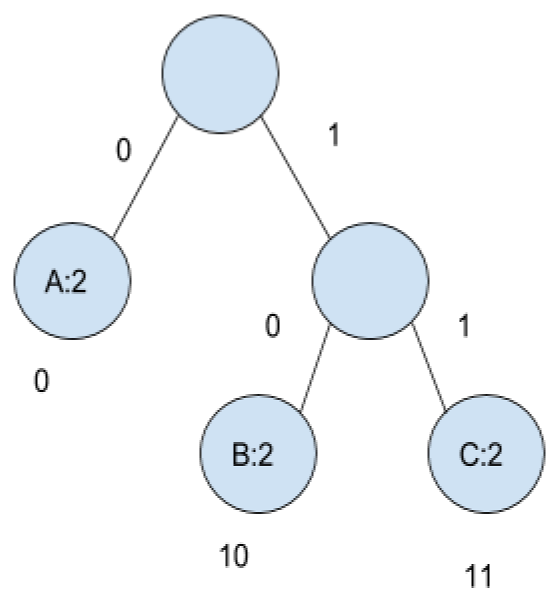
Let's assume S is the set of all symbols and S' is the set of all symbols with probability less than symbol A

$S = \{A, B, C, D, E\}$

$S' = \{\forall i \in S \mid P(i) <= P(A)\} \quad \textit{therefore} \quad S' \in \{B, C, D, E\}$

We can generalize this statement to $P(A) >= P(S'_i)$. For tree H we can take any symbol on the right hand side of the tree and it's binary code length will always be larger than P(A). As Huffman coding states, that the symbols with the lowest probabilities must be pair first.

Case N = 1: where H has only two symbols we can see that P(a) >= P(b) let's assume that a and b have equal probability the length of A and B will be both 1, with their paths equal to 1 respectively 0,1.
Case N = 3 : To follow take the case all probabilities are equal, P(A,B,..,N) = 1/N and N is odd. we can look at the case where 3. We can see that P(A)>=P(B)>=P(C) the lengths respectively are 1<=2<=2.



Argument:
N is the number of symbols in tree T
Hypothesis: The result is true for less than N symbols where N >3.

Contrary argument : T is optimal for N symbols, T contains A,B such that P(A) > P(B) and L(A) > L(B). $L(N_i)$ is the length of the binary code for symbol N_i , $L(N_i)$ is determined by the depth of the symbol.

R is the root of tree T, in which A and B binary code differ by their first bit A = {0….}, B = {1…}. We pick a node J_i and K_i which are parent nodes of A,B respectively, by the Huffman coding definition the total frequency of $P(J_i) > P(K_i)$. By the contrary argument $L(J_i) > L(K_i)$, therefore J_i is deeper than K_i. If we repeat this claim finitely to the Nth case, we will reach nodes J_n and K_n where K_n is the root R and J_n is not. This proves to be a contradiction as the root must contain the largest frequency/probability, since $P(J_n) > P(K_n)$ there is no way for $P(K_n)$ to be the root with a subtree with a probability/frequency. Therefore J_n must be equal to R Proving that the depth of A must less or equal to B, in turn we get L(A) <= L(B).