Objective:

- To look at an application (Data Compression) of Heap Structure.
- We shall not be covering the A-Z of data compression but the basic structures that are needed to do the data compression, more specifically we shall be looking at Huffman Tree.

Task

Let's suppose, we want to compress a data which is only consisting of two characters i.e. a and b only.

Suppose we have the following data, which will take 8 bytes (64 bits) for storage.

aaabbbab

We can compress it and can store this data in 1 byte.

The procedure for this job (compression) will be as follows:

We replace 'a' with '1' and 'b' with '0'.

So, the resultant data would become as follows:

11100010

This binary string will take only 1 byte storage. ☺ How this work?

Actually, after generating this binary string, we take its equivalent decimal value which in this case is 226.

So, the trick is that we will write a 'char' in the compressed file, which will take only 1 byte. Seven bytes saved☺.

Well for decompressing the string we can easily generate the binary pattern of the saved char in order to get the original compressed text.

But the life is not that much simple, it's not a practical scenario that we have picked up because practically there will be many possible characters in a given text file and what codes we going to assign to them will be a problem.

Let's take a look on the following scenario.

So if we apply the same technique then what should we choose to represent

abcdefgaab

If we assign some codes to these characters as shown below.

a=0

b=1

c = 10

d = 11

e = 100

f = 101

g = 110

So, the desired string will become

01**1**011100101110001

Now how can I identify that the 2nd 1 is with its previous one or next one (11 or 10)????

This ambiguity will occur if code of one character is prefix of another character's code.

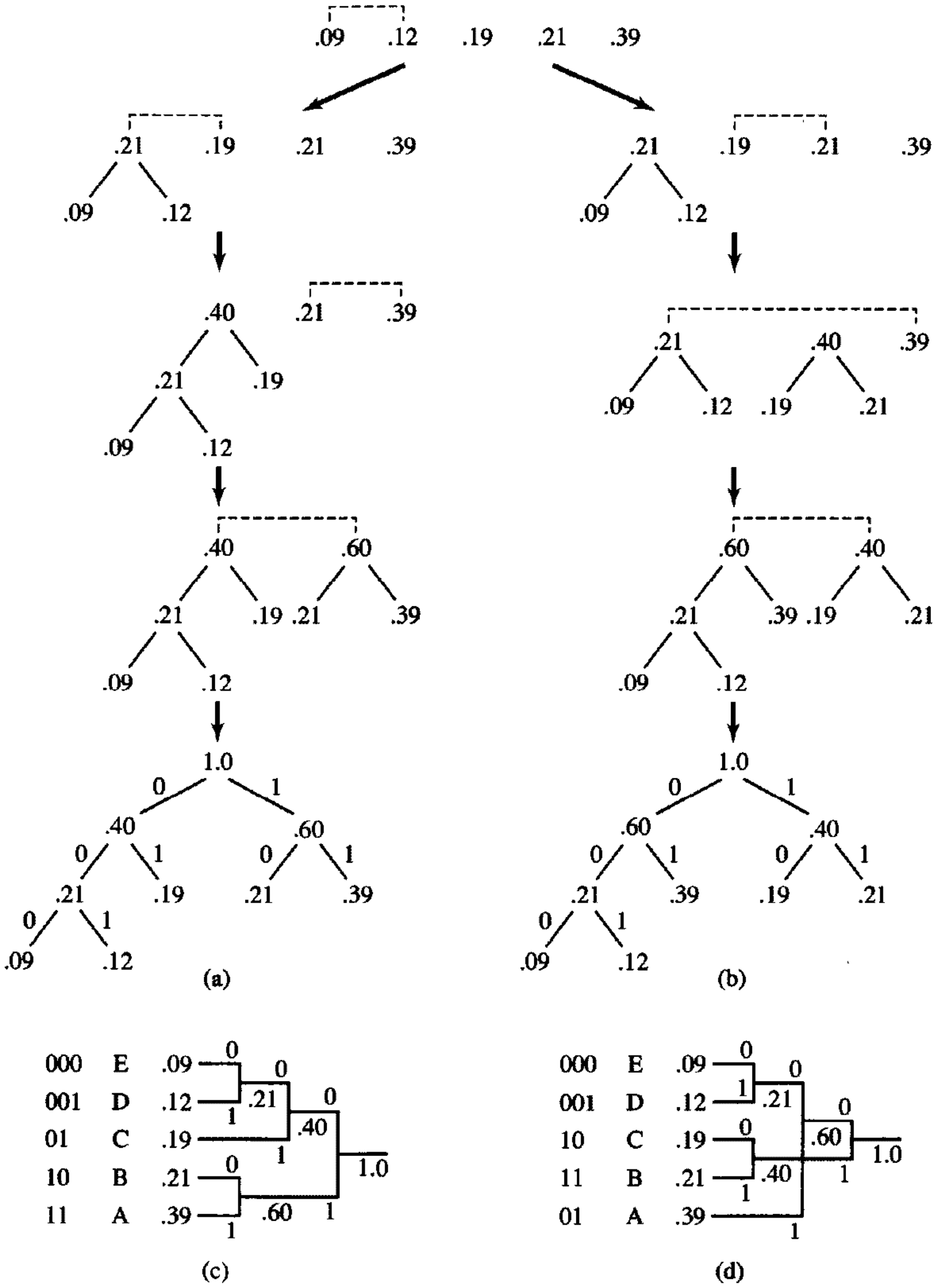
Solution to the above problems:

Build Huffman Tree

So the Huffman Tree/Coding will help us in following:

- It will help us to generate unique (code of one character is not prefix of another character's code) codes of the character used in the text.
- The characters code length for those whose frequency of occurrence is more than the low frequency characters, is less

Steps of making a Huffman Tree [image taken from Text Book – Adam Drozdek]



Your Task: **Assign codes to any given string according to Huffman tree.**