Huffman Coding

Greedy Technique

Huffman Coding

General Info

- Encode a message composed of a string of characters
- Codes used by computer systems
    - ASCII
        - uses 8 bits per character
        - can encode 256 characters
    - Unicode
        - 16 bits per character
        - can encode 65536 characters
        - includes all characters encoded by ASCII
- ASCII and Unicode are fixed-length codes
    - all characters represented by same number of bits

Problem

- Is It possible to send the information in a compressed way?
- Is there any other technique to send the same information with minimal bits?
- In other words, if I have an information and that needs to be transferred over some medium where there is a limitation in sending the original message as such.
- If I change the information, how can I recover it back?
- Also, if I change the information, how can I track, how the information is changed? [Decoding]

Sample Solution

Straight forward approach

- Assume that the message to be transferred is “**DEAACDAAAABAE** ”

Symbol	ASCII	Binary representation
A	65	01000001
B	66	01000010
C	67	01000011
D	68	01000100
E	69	01000101

Symbol	Frequency /Count	Intermediate	Total bits
A	7	7 x 8	56
B	1	1 x 8	8
C	1	1 x 8	8
D	2	2 x 8	16
E	2	2 x 8	16
		Total Bits	104

Simplified Version

Reduction

- Is there any simplified version of the same process, but with different bit size ? YES, lets see.

Symbol	ASCII	Binary representation	Mapping Code
A	65	01000001	000
B	66	01000010	001
C	67	01000011	010
D	68	01000100	011
E	69	01000101	100

Symbol	Frequency /Count	Intermediate	Total bits
A	7	7 x 3	21
B	1	1 x 3	3
C	1	1 x 3	3
D	2	2 x 3	6
E	2	2 x 3	6
		Total Bits	39

We cannot directly use the 3 bit information, we need its mapped ACII value also to decode.

So, the total number of bits required becomes,

$$Total = (5 * 8) + (39)$$

$$Total = 79$$

Is there any possibility to further reduce?

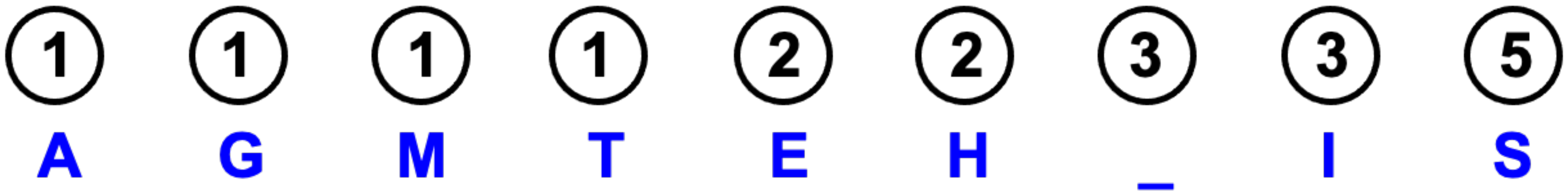
Huffman Technique

- Assume the message what we need to transfer is going to be **“This is his message”**

Character Frequency

A	G	M	T	E	H	_	I	S
1	1	1	1	2	2	3	3	5

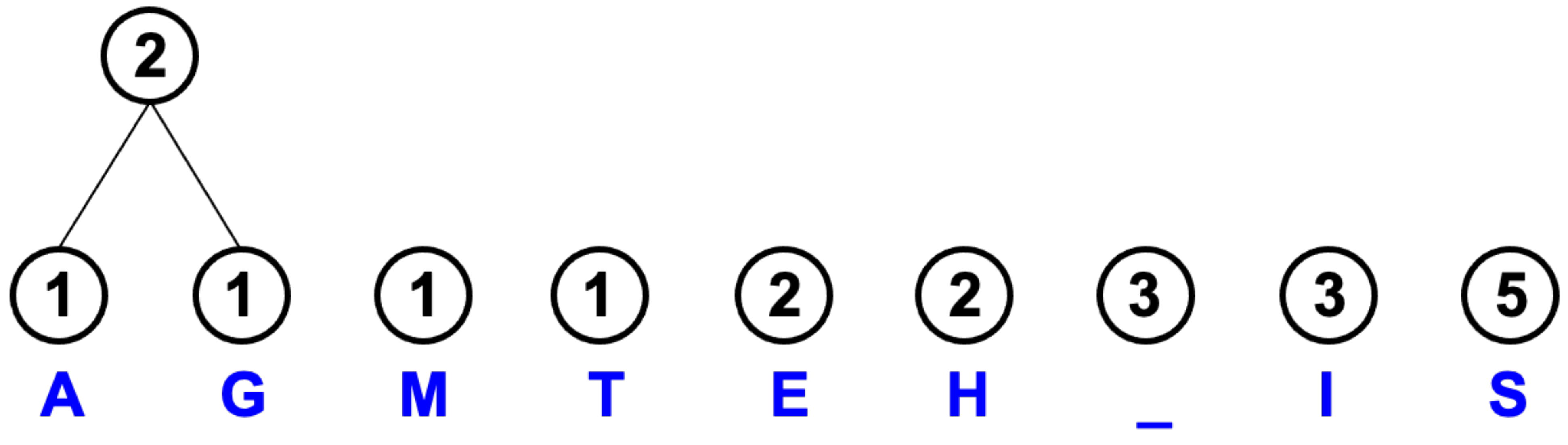
- Representation of every alphabet as a individual tree



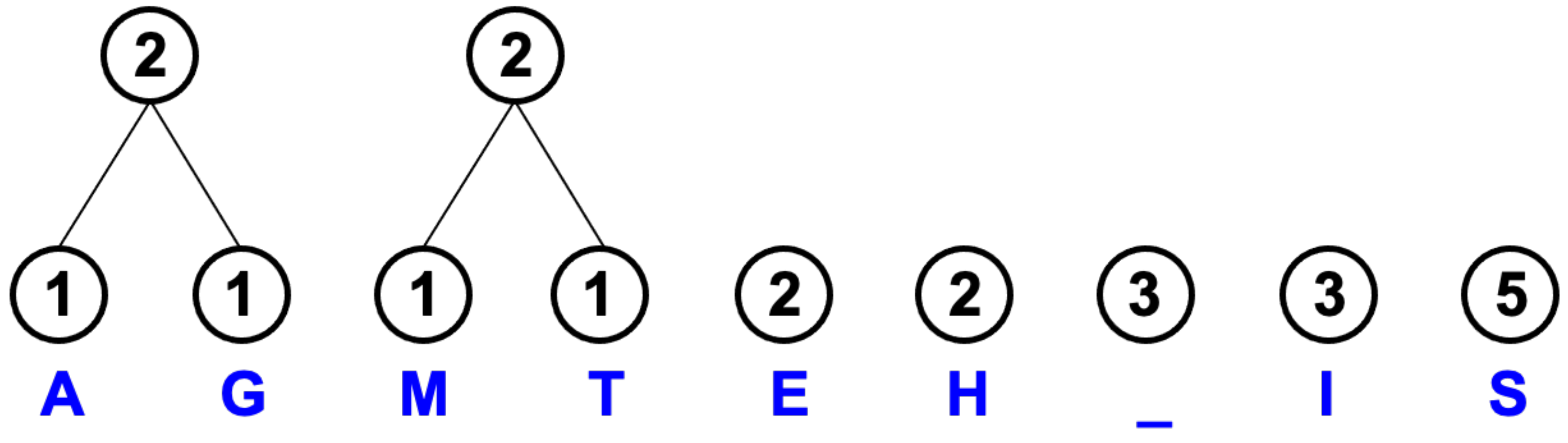
Building the tree

Step by Step

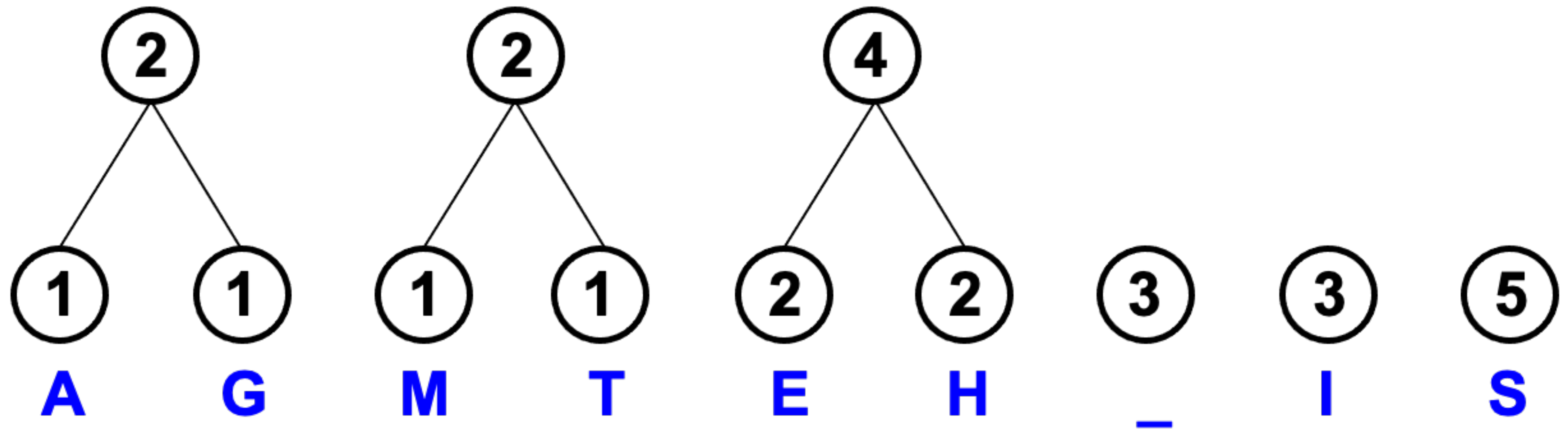
Step 1:



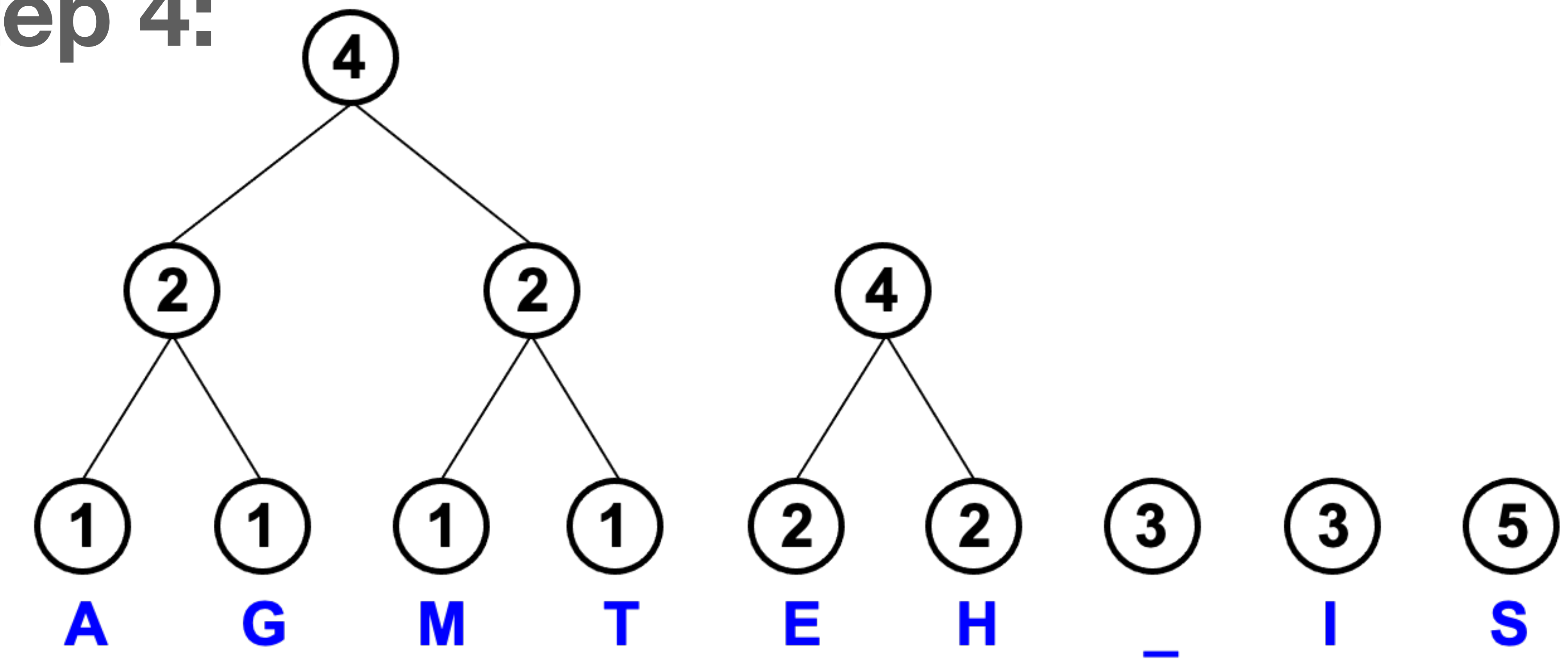
Step 2:



Step 3:

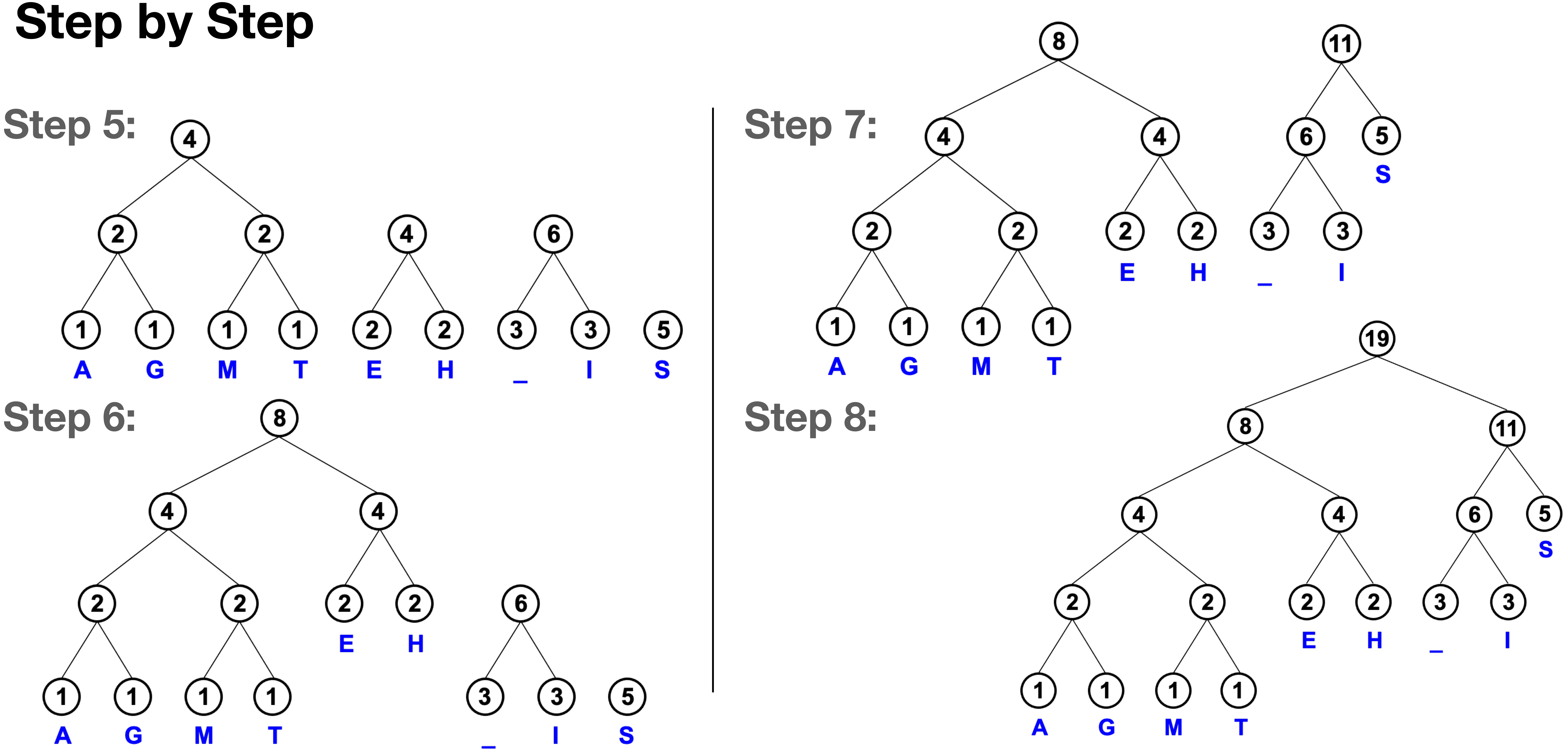


Step 4:



Building the tree

Step by Step



Step 5:

Step 6:

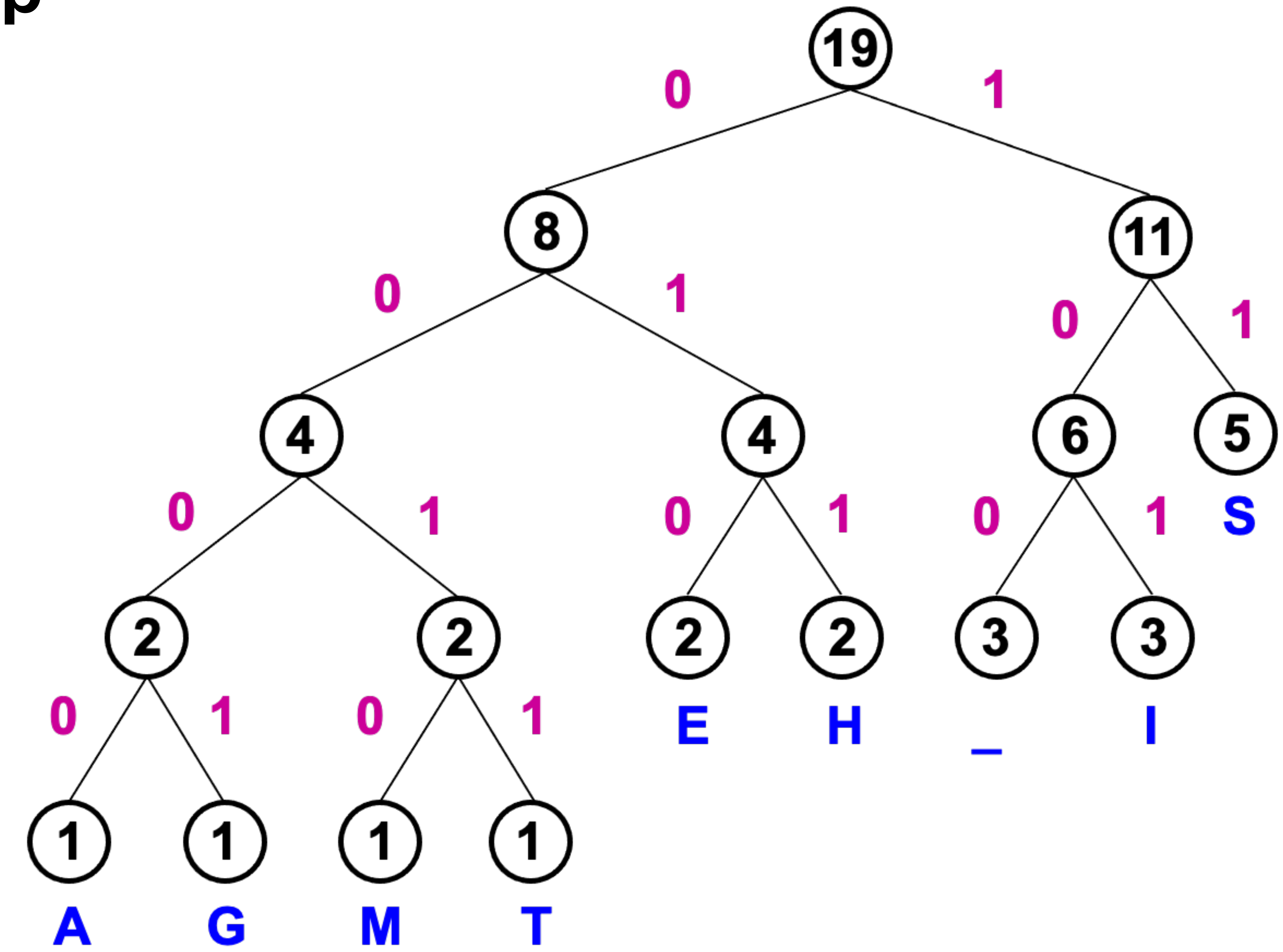
Step 7:

Step 8:

Building the tree

Step by Step

Final Step



S	11
E	010
H	011
_	100
I	101
A	0000
G	0001
M	0010
T	0011

What happen to the Message?

Process

S	11
E	010
H	011
_	100
I	101
A	0000
G	0001
M	0010
T	0011

Final Message become,

00110111011110010111100011101111000010010111100000001010

Prefix Property

- A code has the prefix property if no character code is the prefix (start of the code) for another character
- Example

Symbol	Code
A	000
B	11
C	01
D	001
E	10

Code: 01001101100010

Result is : C D E B A E

Extra Reading

Do at your convenience

- Visualisation - Click Here
- Huffman tree Generator - Click Here
- Source Code & Explanation - Click here
- Worth reading - Click Here