Huffman Code

Huffman Coding Algorithm

Begin with the set S of symbols to be encoded as binary strings, together with their probabilities, and an empty decoding tree.

Repeat the following steps until only 1 symbol left in S:

- Remove the two members of S with lowest probability.
- Add a new symbol to S representing the combined symbols with probability equal to the sum of the probabilities of the two symbols.
- Create a new node of the decoding tree whose children (sub-nodes) are the symbols you've removed.
  - Label the left branch with a "0“
  - Label the right branch with a "1".

Example

Start with

1. Initial set of (symbols, probabilities)
S = {(B, 1/3), (Y, 1/2), (G, 1/12), (R, 1/12)}

2. Empty decoding tree

Example

S = {(Y, 1/3), (B, 1/2), (G, 1/12), (R, 1/12) }

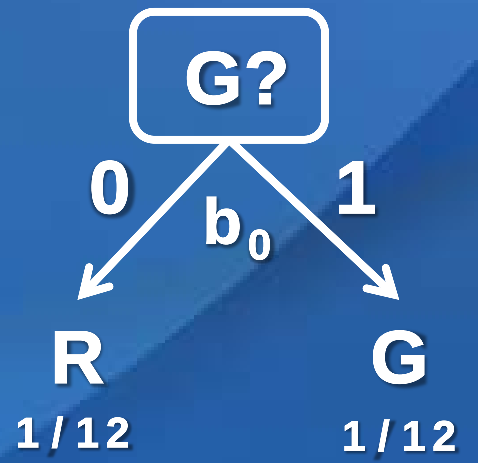
Step 1: Remove two symbols with lowest probabilities: R and G

S = {(Y, 1/3), (B, 1/2)}

Step 2: Add new symbol to S

S = { (Y, 1/3) (B, 1/2) (G?, 1/6) }

Step 3: Create new node, "G?"



Example

S = {(Y, 1/3), (B, 1/2), (G?, 1/6)}

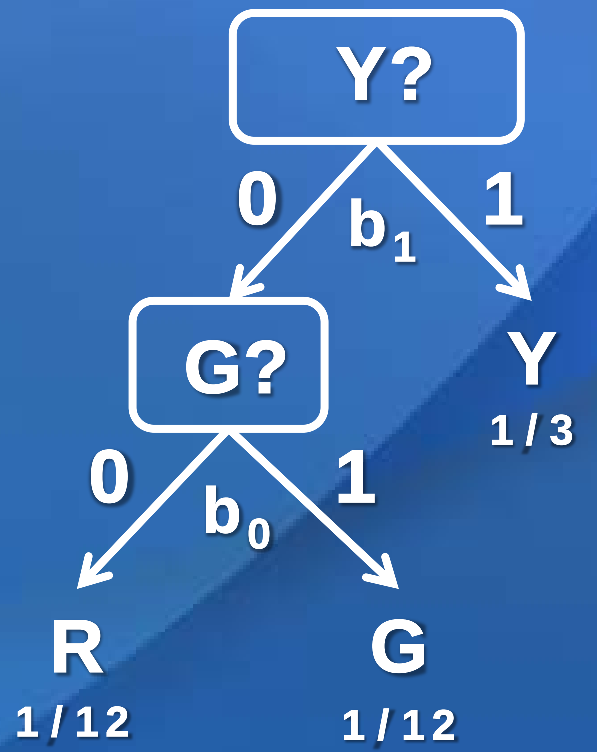
Step 1: Remove two symbols with lowest probabilities: Y and G?

S = {(B, 1/2)}

Step 2: Add new symbol to S

S = { (B, 1/2) (Y?, 1/2)}

Step 3: Create new node, “Y?”



Example

S = {(B, 1/2), (Y?, 1/2)}

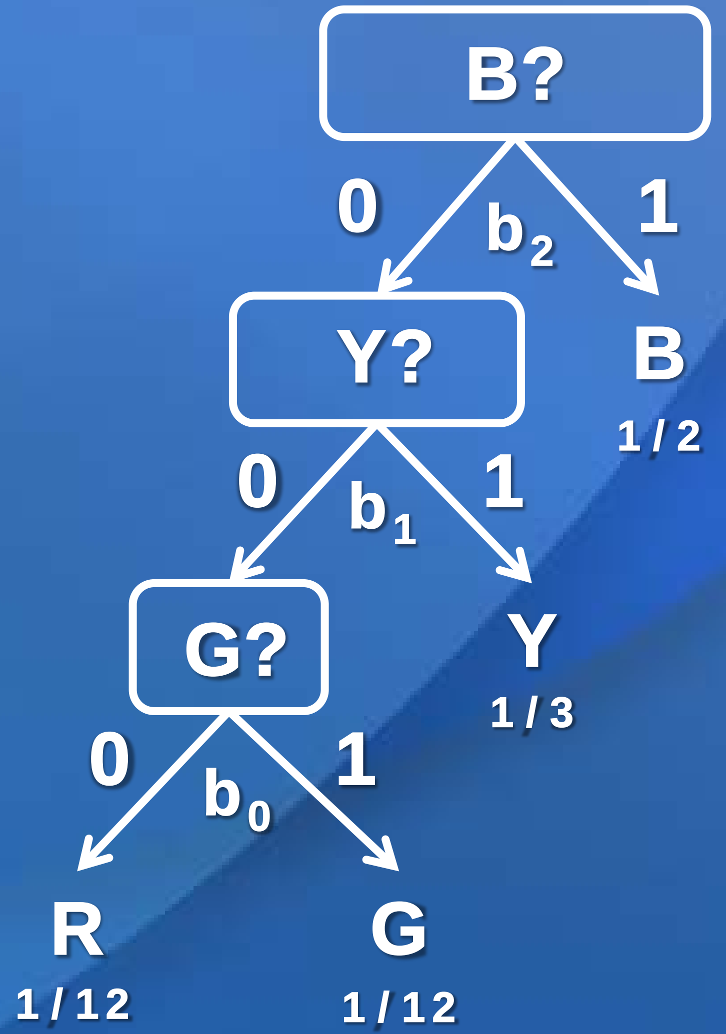
Step 1: Remove two symbols with lowest probabilities: B and Y?

S = { }

Step 2: Add new symbol to S

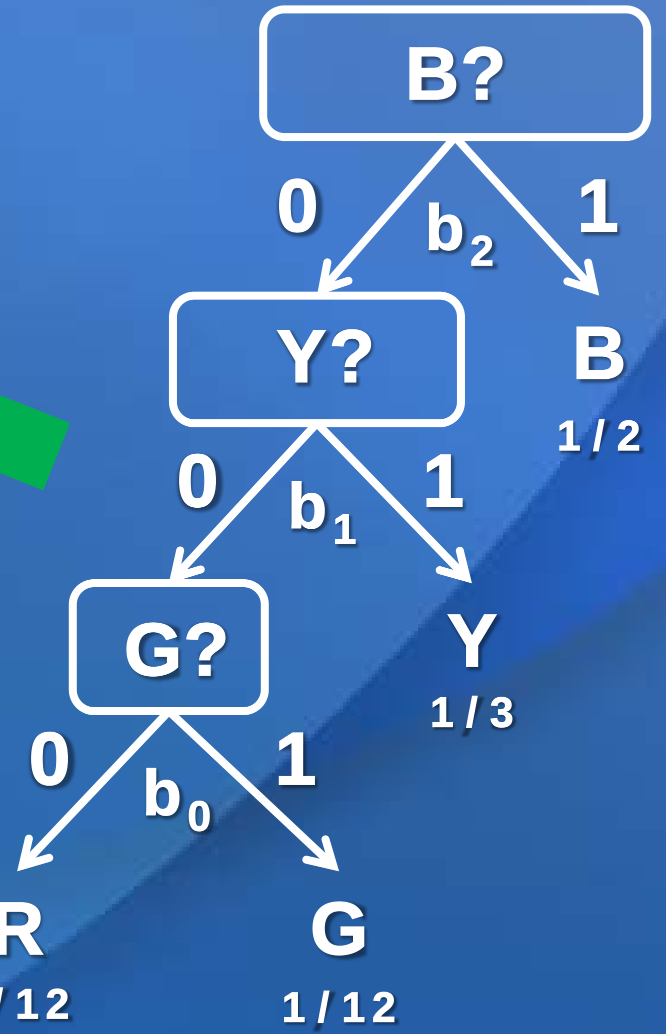
S = {(B?,1)}

Step 3: Create new node, “B?”



Encoding

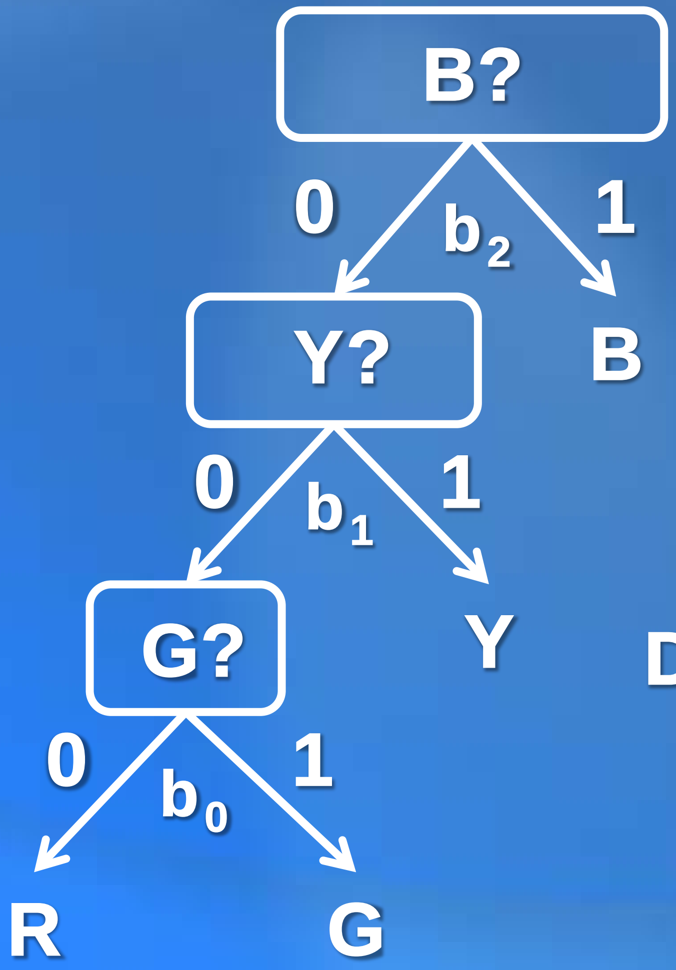
x_k	p_k	*encoding*
B	1/2	1
Y	1/3	01
G	1/12	001
R	1/12	000



Encoding of GRYBGBY:

001000011001101

Decoding



Decode: 001000011001101
G R YB GB Y

Average Code Length

$$H = -\sum_{k=0}^{K-1} p_k \log_2(p_k)$$

$$= -\tfrac{1}{2}\log_2(\tfrac{1}{2}) - \tfrac{1}{3}\log_2(\tfrac{1}{3}) - \tfrac{1}{12}\log_2(\tfrac{1}{12}) - \tfrac{1}{12}\log_2(\tfrac{1}{12})$$

$$\approx 1.626$$

$$\boxed{\overline{L} > H}$$

$$\overline{L} = \sum_{k=0}^{K-1} p_k\, l_k$$

$$= \tfrac{1}{2}\cdot 1 + \tfrac{1}{3}\cdot 2 + \tfrac{1}{12}\cdot 3 + \tfrac{1}{12}\cdot 3$$

$$\approx 1.667$$