Huffman Decoding

Shikha Nair

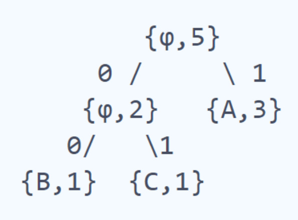
October 23, 2017

Question

Huffman coding assigns variable length codewords to fixed length input characters based on their frequencies. More frequent characters are assigned shorter codewords and less frequent characters are assigned longer codewords. A Huffman tree is made for the input string and characters are decoded based on their position in the tree. We add a '0' to the codeword when we move left in the binary tree and a '1' when we move right in the binary tree. We assign codes to the leaf nodes which represent the input characters.

Input characters are only present on the leaves. Internal nodes have a character value of '\0'.

For example, the following Huffman tree has codewords A=1, B=00, and C=01.



```
        {φ,5}
     0 /     \ 1
    {φ,2}   {A,3}
   0/   \1
{B,1}  {C,1}
```

Encoded String "1001011" represents the string "ABACA".

Given the root node of a Huffman tree, write a method which takes a Huffman-encoded string and prints the decoded string.

Explanation and Algorithm

Your algorithm will iterate through the given encoded string S and, for each character, move through the Huffman tree. there are two cases for the current character in S, 0 and 1. You change your position in the tree based on this character, either moving to the left or right node, respectively. Once you have changed positions, check if the current node is a leaf by checking if it has a character value. If it doesn't, move on to the next character in S. If it does, print the character, reset your position to the root of the tree, and move on to the next character in S.

Hints

1. For this method you want to iterate through the given encoded string. For each character, what should you check?
2. What are the possible cases for each character you read from the encoded string? What should you do in the Huffman tree in each case?
3. You should move through the Huffman tree depending on the current character in S. After you move in the tree, what should you check in the current node?
4. Once you reach a leaf node and print the decoded letter, how should you reset your position in the Huffman tree?

Code

```java
void decode(String S , Node root)
    {
        Node curr = root;
        for(int i=0; i<S.length(); i++){
            if(S.charAt(i)==’0’){
                curr = curr.left;
                if(curr.data != ’\0’){
                    System.out.print(curr.data);
                    curr=root;
                }
            }else{
                curr = curr.right;
                if(curr.data != ’\0’){
                    System.out.print(curr.data);
                    curr=root;
                }
            }
        }


    }
```

Big O Analysis

The run-time of this method is O(n), where n is the length of the encoded string.

Sources

Question, answer and other material taken from HackerRank.com and GeekforGeeks.org.