

## CS 2321 - Quiz 2

Using the following methods, and any Sequence or Entry methods, write pseudocode for Huffman encoding and decoding. You may assume you have a table of entries,  $T$ , an Array Sequence of  $\text{Entry}\langle \text{char}, \text{Encoding} \rangle$ . The table is ordered by character.

### String

- $\text{Iterator}\langle \text{char} \rangle \text{ iterator}()$ ;
- $\text{char charAt}(int)$ ;
- $int \text{ length}()$ ;

### Encoding

- $\text{Iterator}\langle \text{boolean} \rangle \text{ iterator}()$ ;
- $int \text{ length}()$ ;
- $boolean \text{ getbit}(int)$ ;
- $Encoding \text{ getbits}(int, int)$ ;
- $boolean \text{ equals}(Encoding)$ ;

1. Write pseudo-code to encode a string. You may concatenate encodings with a “+”.

**Algorithm:**  $B$  Encoder( $S, T$ )

**Input** :  $S$  - a string of characters ( $n$  chars)

$T$  - ordered Array Seq. of  $\text{Entry}\langle \text{char}, \text{Encoding} \rangle$  ( $m$  entries)

**Output:**  $B$  - a binary encoding based on the table  $T$

```
for char  $c$  in  $S$  do
   $low \leftarrow 0$ 
   $high \leftarrow n$ 
   $i \leftarrow (low + high)/2$ 
  while ( $c \neq T.get(i).getKey()$ ) do
    if ( $c < T.get(i).getKey()$ ) then
       $high \leftarrow i$ 
    else
       $low \leftarrow i$ 
    end
     $i \leftarrow (low + high)/2$ 
  end
   $B \leftarrow B + T.get(i).getValue()$ 
end
```

2. In terms of  $n$  and  $m$ , what is the worst case time complexity of encoding a string?

$O(n \lg m)$

3. If  $T$  were not sorted, what would you change about the algorithm?

Do not use binary search, use sequential.

4. If  $T$  were a Linked Sequence instead of an Array, what would you change?

Do not use binary search, use iterator instead.

5. Write pseudo-code to decode a string.

**Algorithm:**  $S$  Decoder( $B, T$ )

**Input** :  $B$  - a binary encoding based on the table  $T$  ( $b$  total bits)  
 $T$  - ordered Array Seq. of Entry<char, Encoding> ( $m$  Entries)

**Output:**  $S$  - a string of characters

$i \leftarrow 0$

**while** ( $i < B.length()$ ) **do**

```
    for Entry  $ent$  in  $T$  do
         $len \leftarrow ent.getValue().length()$ 
        if  $ent.getValue() = B.getbits(i, i + len)$  then
             $S \leftarrow S + ent.getKey()$ 
             $i \leftarrow i + len$ 
            break
        end
    end
```

**end**

6. In terms of  $b$  and  $m$ , what is the worst case time complexity of decoding a binary sequence ( $T$  is ordered by character)?

$$O(bm \lg m)$$

7. If we were given the Huffman Tree instead of the Table of encodings, could the algorithm be improved? If so, how?

Yes. If given the Huffman Tree, we could start at the root and work down with each bit. When we reach a leaf node, we add the corresponding character, and restart at the root. This reduces the time to  $O(b)$