COS 226 Data Structures and Algorithms
Computer Science Department
Princeton University
Fall 2015

Week 10 Activity

1. **Burrows-Wheeler Data Compression Algorithm**
   The final assignment is divided into 3 parts

   (a) **Burrows-Wheeler transform**. Given a typical English text file, transform it into a text file in which sequences of the same character occur near each other many times.

   (b) **Move-to-front encoding**. Given a text file in which sequences of the same character occur near each other many times, convert it into a text file in which certain characters appear more frequently than others.

   (c) **Huffman compression**. Given a text file in which certain characters appear more frequently than others, compress it by encoding frequently occurring characters with short codewords and rare ones with long codewords.

   It is important to get started early (Assignment Due 12/16). Start with MoveToFront code that can be done independent of Burrows-Wheeler transform and compression algorithms (to be discussed next week).

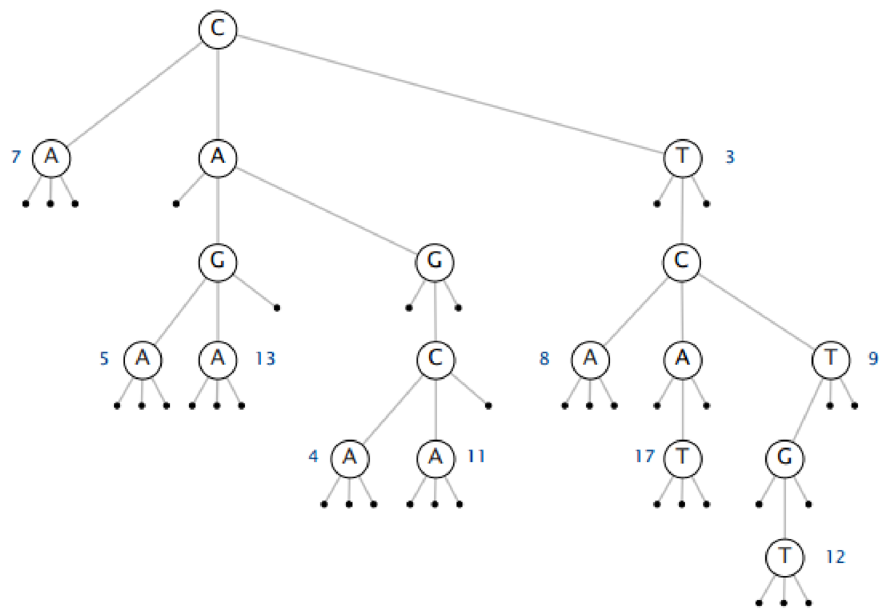
```
public class MoveToFront {
    // apply move-to-front encoding, reading from standard
input and writing to standard output
    public static void encode()

    // apply move-to-front decoding, reading from standard
input and writing to standard output
    public static void decode()

    // if args[0] is '-', apply move-to-front encoding
    // if args[0] is '+', apply move-to-front decoding
    public static void main(String[] args)
}
```

2. Tries

(a) **Ternary Search Trees (TST)**
Consider the following TST, where the values are shown next to the nodes of the corresponding string keys.



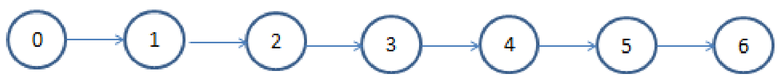
i. list the items (in alphabetical order) that were inserted into the TST.

ii. Insert the two strings **CGTT and TGA** into the TST with the associated values 0 and 99, respectively; update the figure above to reflect the changes.

(b) Under what circumstances would you use a R-way Trie instead of a TST? Discuss pros and cons of each approach.

3. **Substring search**

(a) Construct the Knuth-Morris-Pratt DFA for the string ABAABA over the alphabet $\{A, B\}$. Complete the transition diagram and the corresponding DFA table. State 6 is the accept state.



	0	1	2	3	4	5
A						
B						

(b) Below is a partially-completed Knuth-Morris-Pratt DFA for a string s of length 8 over the alphabet $\{A, B\}$. State 8 is the accept state. Reconstruct the DFA and s in the space below.

	0	1	2	3	4	5	6	7
A	0					6		8
B		1	3				5	
s								