PRINCETON UNIVERSITY

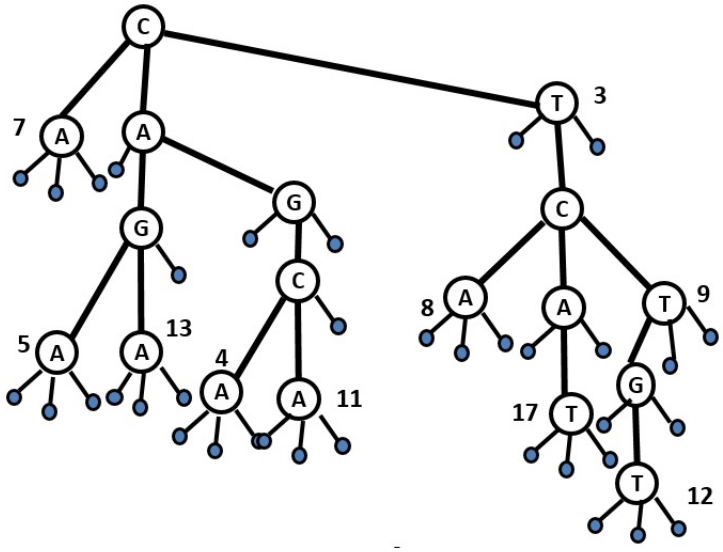
COS 226–Algorithms and Data Structures

Week 11: *Tries, KMP & Algorithm Design*
(Algs. §5.2,5.3 & videos §21.B, 22.C)

Version: December 8, 2017

Exercise 1 – Ternary Search Tries (TST)

Consider the following TST, where the values are shown next to the nodes of the corresponding string keys.



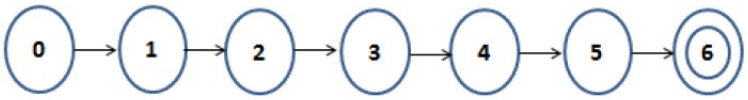
A. List the items (in alphabetical order) that were inserted into the TST.

B. Insert the two strings CGTT and TGA into the ternary search trie with the associated values 0 and 99, respectively; update the figure above to reflect the changes.

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

Exercise 2 – 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								

Exercise 3 – Finding all Patterns

An interesting problem in string searching is finding ”all” substrings that match a given substring pattern. For example, if we apply KMP-algorithm to find the pattern ”abab” in the text ”abababababab”, it only finds two patterns. However, closer examination of the text and pattern reveals that there are indeed 4 patterns that can be detected. Design an efficient algorithm to detect ”all” patterns in a text, given a text and a pattern, where patterns can be overlapping in the text.

A. Describe your algorithm in the space below (with figures if necessary).

B. What is the order of growth of the worst-case running time of your algorithm as a function of n, m and R, where n is the length of the text, m is the length of the pattern and R is the size of the alphabet?

