Andrew Plum
Prof. Beeston
CS 395

Assignment #2

11/4/2023

1) a) DFS:
(Letter$_{push,pop}$)

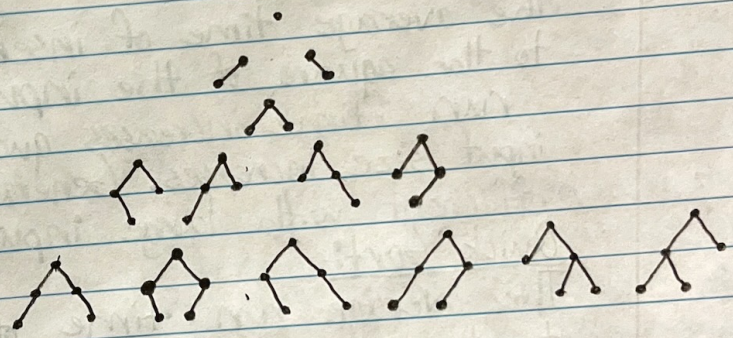
$E_{3,1}$ $F_{5,2}$
$B_{2,4}$ $G_{4,3}$
$A_{1,6}$ $C_{6,5}$ $D_{7,7}$

Source Removal: 1) D
2) A
3) B
4) C
5) G
6) E
7) F

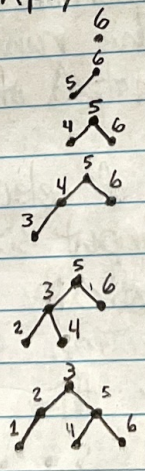
Order: ABEGFCD
b) Can't topologically sort because the digraph has a cycle

2)

n=1:
n=2:
n=3:
n=4:
n=5:



3) Start: empty tree
Next:
Next:
Next:
Next:
Next:
Next:



4) Back Side

4)

	Unordered Array		Ordered Array		Binary Search Tree		Balanced Search Tree		Hashing	
	Average-Case	Worst-Case	Average-Case	Worst-Case	Average-Case	Worst-Case	Average-Case	Worst-Case	Average-Case	Worst-Case
Search	O(n)	O(n)	O(log n)	O(log n)	O(log n)	O(n)	O(log n)	O(log n)	O(1)	O(n)
Insert	O(1)	O(1)	O(n)	O(n)	O(log n)	O(n)	O(log n)	O(log n)	O(1)	O(n)
Delete	O(n)	O(n)	O(n)	O(n)	O(log n)	O(n)	O(log n)	O(log n)	O(1)	O(n)

5) a) Big Theta Complexity
- Insertion Sort: $\Theta(n^2)$
- Quick Sort: $\Theta(n \log n)$
- Selection Sort: $\Theta(n^2)$

b) Insertion Sort:
The average time of insertion sort is proportional to the square of the input size meaning that the run time increases quadratically as the input size increases. Generally, insertion sort is not efficient with large input sizes.

Quick Sort:
The average run time of quick sort is proportional to n log n where n is the input size meaning quick sort is more efficient than insertion and selection sort for large input sizes because the run time grows logarithmically with the input size instead of quadratically.

Selection Sort:
The average run time of selection sort is proportional to the square of the input size meaning that the run time increases quadratically as the input size increases meaning generally selection sort is as slow as insertion sort and generally slower than quick sort.

6) Next Page

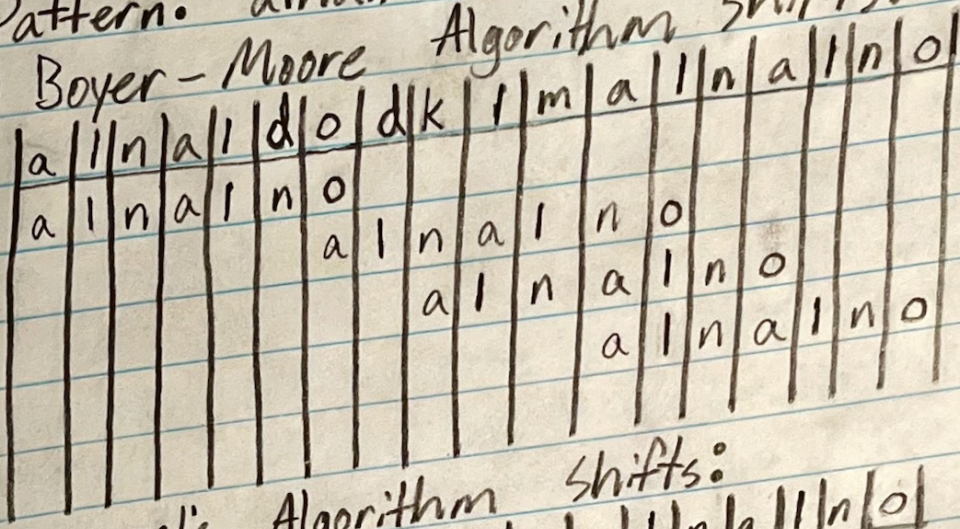
Andrew Plum
Prof. Beeston
CS 395
11/4/2023

Assignment #2 Cont.

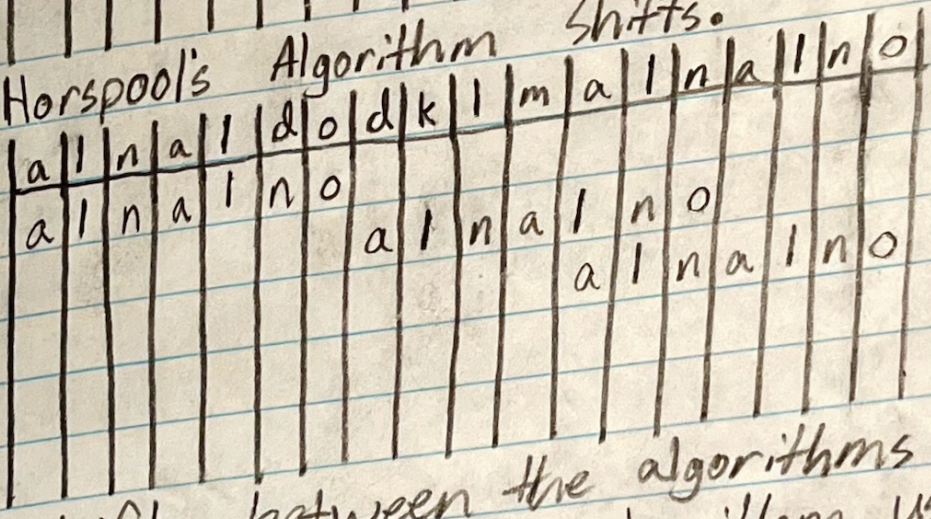
6) a) Input text: "aaaaaaaaaaaaaaa"
Pattern: "aaaaab"

6) i) Input text: "alnaldodklmalnalno"
Pattern: "alnalno"

ii) Boyer-Moore Algorithm shifts:



Horspool's Algorithm shifts:



The shifts between the algorithms are different because the Boyer-Moore algorithm uses the good suffix rule and the bad character rule whereas Horspool's algorithm only uses the bad character rule.