Trevor Smith

CS 321 Data Structures (Fall 2020)

Final Exam (100 points), 12/16/2020

Q1(12 points): Running time and asymptotic notations

a) (3 points) (Multiple choice) If an algorithm's running time can be expressed as a function $f(n) = \Theta(n) + O(n^3)$, then which one of the following running time asymptotic notations is not possible for the algorithm?

1. $\Theta(1)$
2. $\Theta(n)$
3. $\Theta(n^2)$
4. $\Theta(n^3)$

b) (3 points) (Multiple choice) Which one of the following sorting algorithms has the WORST running time if the input array is already sorted.

1. Merge sort
2. Insertion sort
3. quick sort
4. heap sort

c) (6 points) Given the following functions, please re-order them based on their growth rate (from the least to the greatest). In addition, please underline those functions having the same growth rate asymptotically (if any).

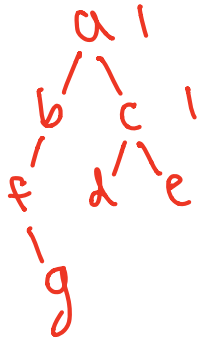
$log_2(n!)$, $log_2 n^{1000}$, $\sqrt{2}^n$, $(log_2 n)^n$, $n!$, $log_e n^n$, 2^n, $\sqrt{n}$

$\sqrt{n}$, $log_2 n^{1000}$, $log_2(n!)$, $\sqrt{2}^n$, 2^n, $log_e n^n$, $(log_2 n)^n$, $n!$

Q2(36 points): Trees

a) (6 points) Given a binary tree, please write a pseudocode to calculate the smallest depth among all leaf nodes. That is, to calculate the shortest path length among all paths from the root to all leaves.

```
ShortPath(r)        // r is the root of the tree;  Return the length of shortest path
{
    if (r == null)    return ____0____;
    llength = ShortPath (r.left);
    rlength = ShortPath (r.right);
    if (llength <= rlength)
        return ____llength + 1____;
    else
        return ____rlength + 1____;
}
```



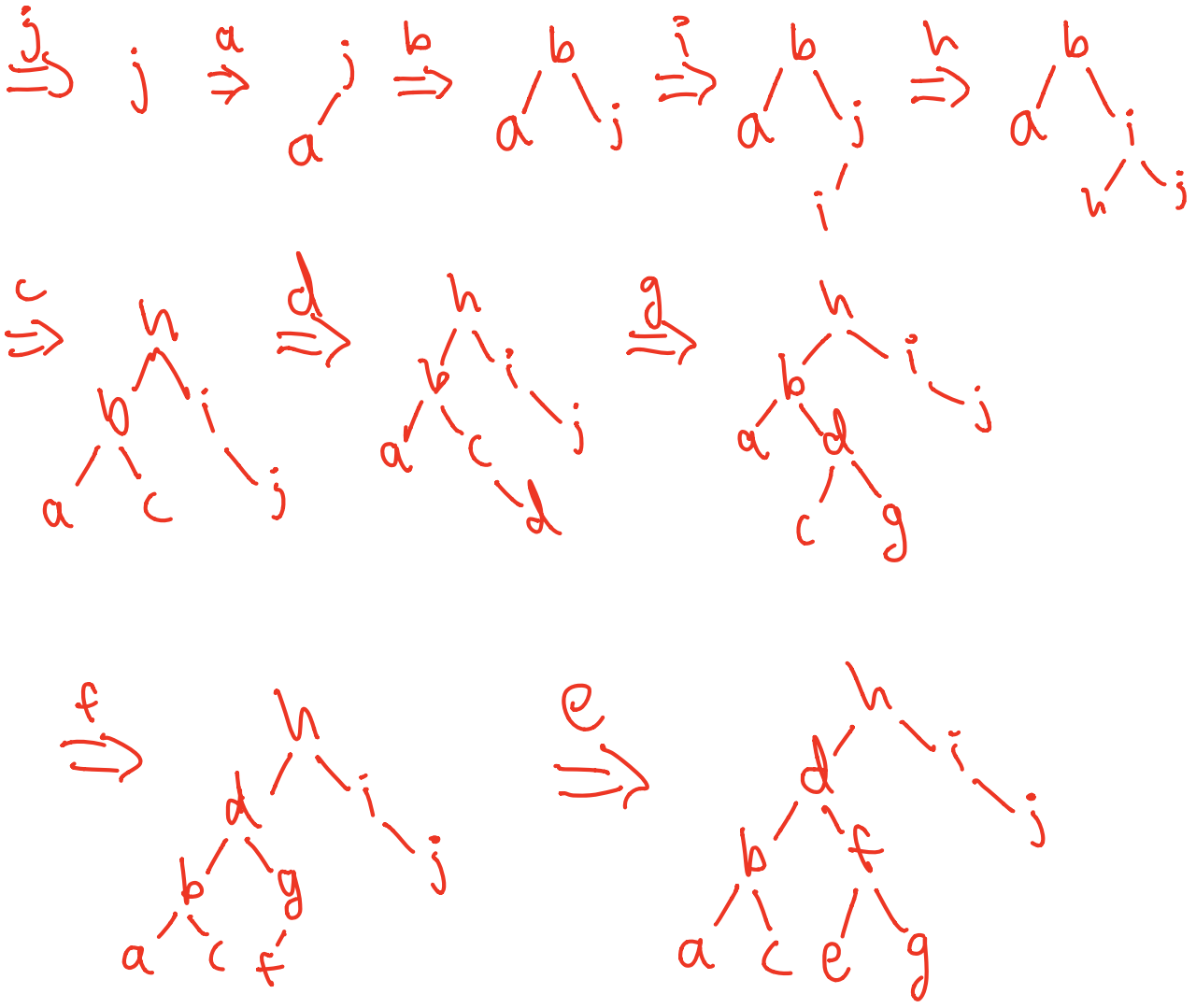
b) (10 points) For a binary tree rooted at *r*, please write a non-recursive procedure to print the nodes in a pre-order traversal.

PreOrderPrint(r) // r: root of a binary tree
{

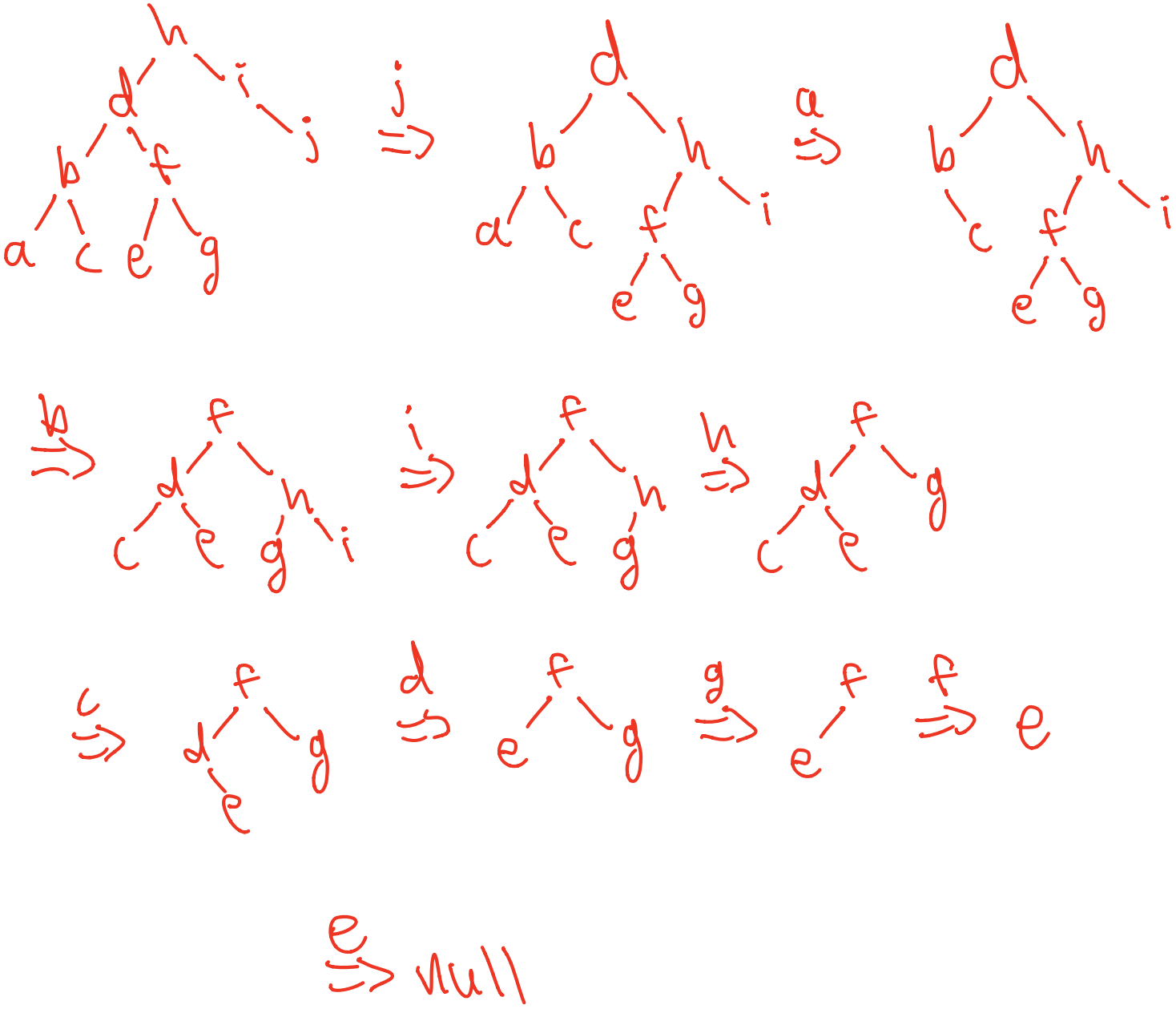
```
1. if(r is not null){
    A. Initialize empty stack s
    B. S.push(r);
    C. While(s is not empty){
        a. n = s.pop;
        b. Print n;
        c. If(n.right is not null){
            1. S.push(n.right);
        d. }
        e. If(n.left is not null){
            1. S.push(n.left);
        f. }
    D. }
```

}

c) (10 points) What are the sequence of AVL tress after inserting each character in the list < j, a, b, i, h, c, d, g, f, e > to an initially empty AVL tree? Note: please draw only one tree after each insertion.

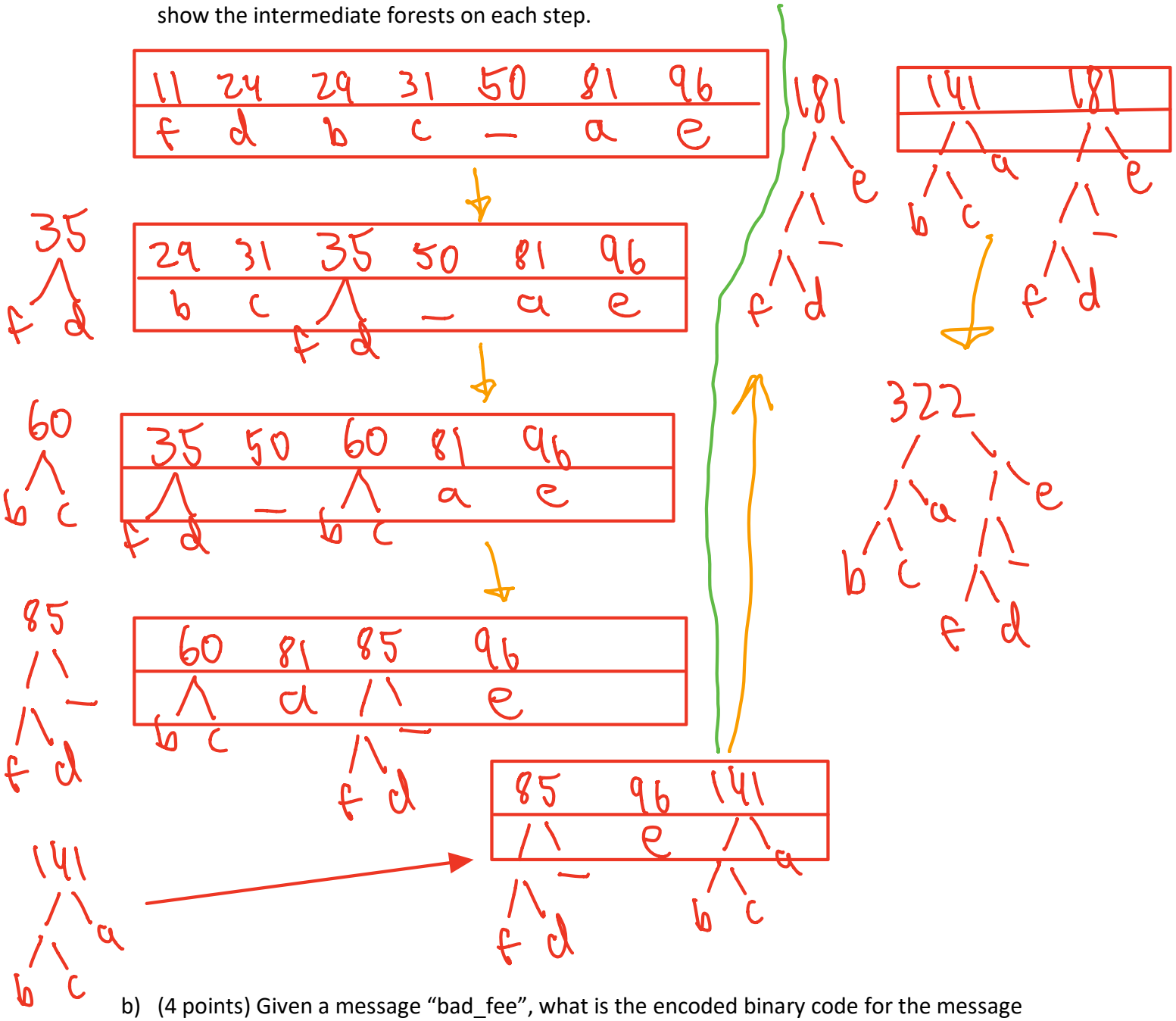


d) (10 points) From the AVL tree you have built in part (b), what are the sequence of trees after deleting each character in the list (i.e., delete j, delete a, ...)? Note: please draw only one tree after each deletion.



Q3(14 points) More Trees – Huffman Tress

a) (10 points) Given a text file with only six different characters {a, b, c, d, e, f, _}, where the char '_' is a white space. The frequencies of these characters in the file are {(a: 81), (b: 29), (c: 31), (d: 24), (e: 96), (f: 11), (_: 50)}. Based on these frequencies, please construct a Huffman tree with only seven leaf nodes (one for each of these seven characters). Please show the intermediate forests on each step.



b) (4 points) Given a message "bad_fee", what is the encoded binary code for the message based on the Huffman tree you have just built in part (a)?

a 01
b 000
c 001
d 1001
e 11
f 1000
_ 101

000 01 1001 101 11 11 11

Q4(19 points) Data Structures

a) (3 points) (Multiple choice) To search for a stored element, which one of the following data structures has the BEST average running time?

1. binary maxheap
2. hash table
3. balanced binary search tree
4. AVL tree

b) (3 points) (Multiple choice) The following operations are performed frequently on a collection: adding items, searching items, removing items, and listing items in order. Which of the following data structures would be the best choice for storing this collection?

1. maxheap
2. hash table
3. BTree
4. AVL tree

c) (3 points) (Multiple choice) We can either use an array or a singly circular linked list toimplement a queue. Which one of the following statements is true? (Assume there is only one entry point to the linked list - Tail).

1. Both enQueue and deQueue run in constant time for the worst-case in the array implementation, but not in the linked list implementation.
2. Both enQueue and deQueue run in constant time for the worst-case in the linked list implementation, but not in the array implementation.
3. Both enQueue and deQueue run in constant time for the worst-case in both implementations.
4. None of the enQueue and deQueue run in constant time for the worst case in both implementations.

d) (10 points) Suppose we would like to insert a sequence of numbers < 48, 20, 6, 77, 13 > into a hash table with table size 7 using the three open addressing methods, with the primary hash function $h_1(k) = k \ mod \ 7$, the secondary hash function

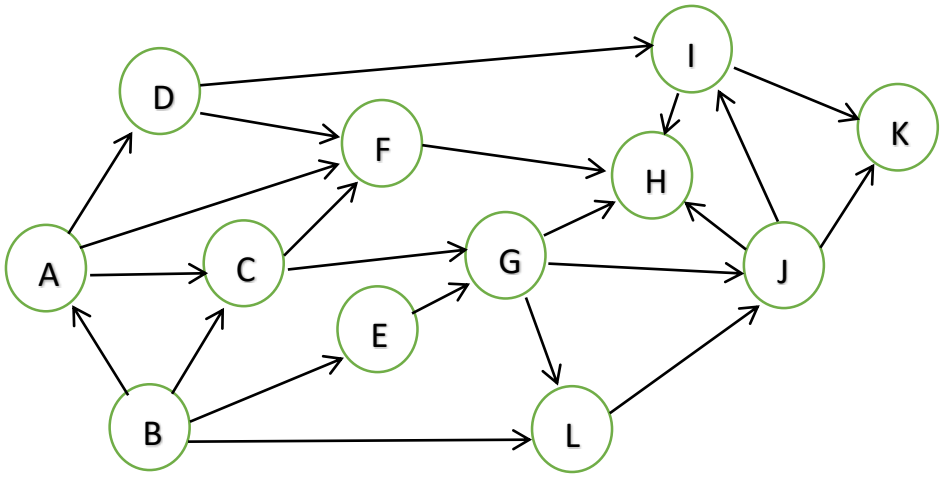
$$h_2(k) = \begin{cases} 2 & if \ k \ mod \ 6 \ is \ even \\ 4 & if \ k \ mod \ 6 \ is \ odd \end{cases}$$

and the constants $c_1 = c_2 = 1/2$ (in quadratic probing). Please insert numbers to the tables below.

Index	Linear	Quadratic	Double
0	20	20	77
1	6	77	20
2	77	6	
3	13		6
4			13
5		13	
6	48	48	48

Q5(19 points) Graphs

A weighted and directed graph is given below.



a) (4 points) Let vertex *B* be the source vertex, please find a (any) discovering sequence of vertices in a BFS search.

B, ACEL, DFGJ, IHK

b) (4 points) Please find the discovering sequence of vertices in the DFS search, assuming that during the search if there are multiple vertices can be discovered next, please discover vertices based on their alphabetical order.

BACFHGJIKLDE

c) (4 points) Based on the DAG above, please find the topological sequence of vertices, assuming that during the DFS search if there are multiple vertices can be discovered next, please discover vertices based on their alphabetical order.

BECADGLJIKFH

d) (7 points) Please find a (any) minimum spanning tree of the graph below