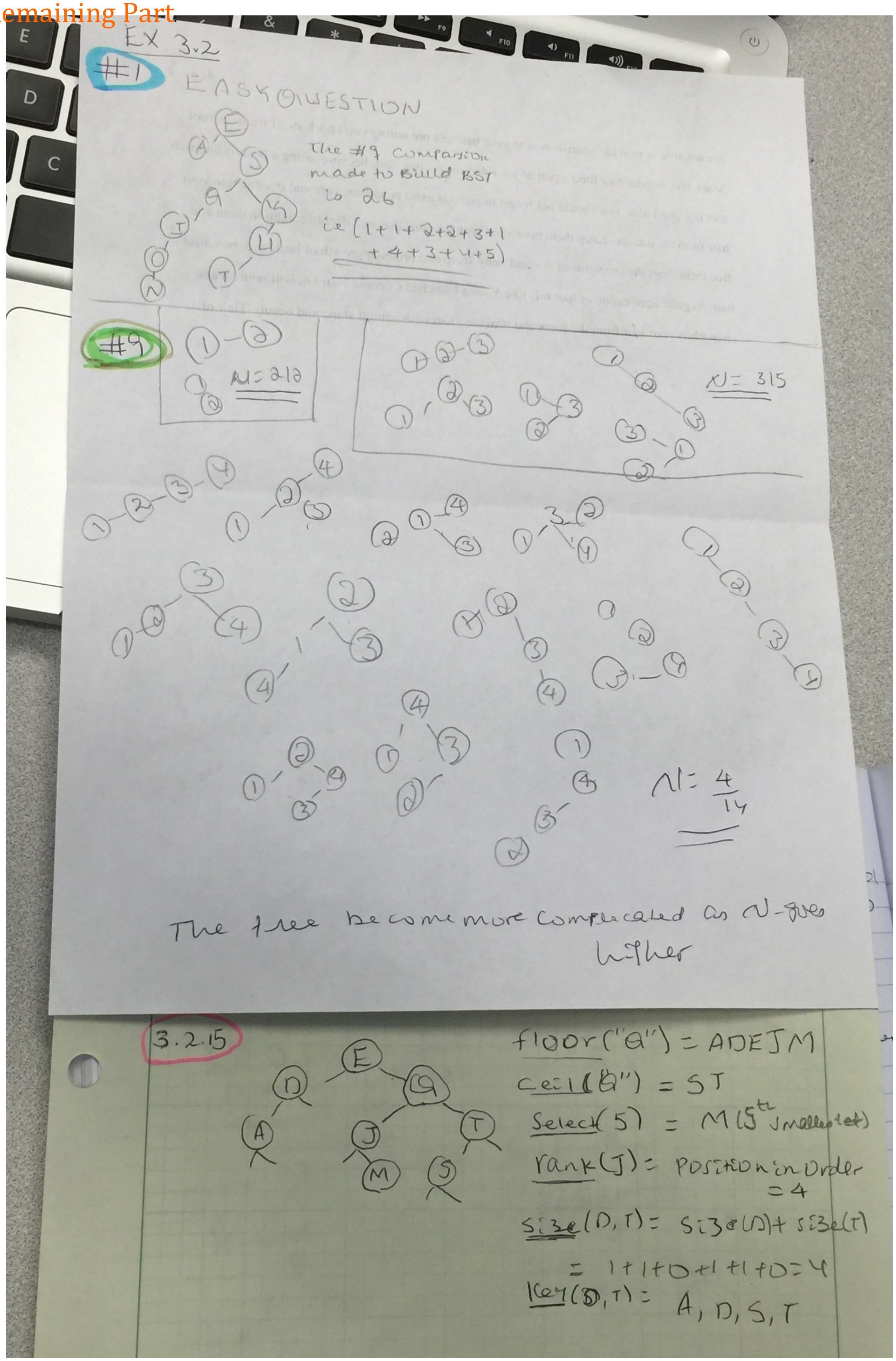
Name Abinet Kenore
CS 2050
HW #3
Remaining Part



(4) The maximum no of nodes in a BT of depth N is $2^N - 1$, $N \geq 1$

(5) The Minimum no of Nodes in a BT of depth N is One, b/c

$2^N - 1$, $N \geq 1$

$2^1 - 1 = 2 - 1 = 1$

i.e at least there will be root without any leaf/children

The following note is something, which I used for Citation for the above diagrams. (GOOGLE)

The number of binary search trees can be seen as a recursive solution. i.e., Number of binary search trees = (Number of **Left** binary search sub-trees) * (Number of **Right** binary search sub-trees) * (Ways to choose the root)

In a BST, only the relative ordering between the elements matter. So, without any loss on generality, we can assume the distinct elements in the tree are *1, 2, 3, 4, ...., n*. Also, let the number of BST be represented by *f(n) for n elements*.

Now we have the multiple cases for choosing the root.

1. choose 1 as root, *no* element can be inserted on the left sub-tree. *n-1* elements will be inserted on the right sub-tree.
2. Choose 2 as root, *1* element can be inserted on the left sub-tree. *n-2* elements can be inserted on the right sub-tree.
3. Choose 3 as root, *2* element can be inserted on the left sub-tree. *n-3* elements can be inserted on the right sub-tree.