1. Recall the definition of a *binary search tree*: a binary tree is a binary search tree if:

- It is a nullptr
- It is not a nullptr, has a key, and:
  - The left-hand child is a binary search tree, all of whose keys are less than this node’s key.
  - The right-hand child is a binary search tree, all of whose keys are greater than this node’s key.

(a) There are five possible binary search trees (down to isomorphism) with three nodes. Draw them.

(b) There is only one possible binary search tree of size one (one vertex), there are two of size two, five of size three, and 14 of size four. Use this information to determine how many binary search trees of size five are possible. *An answer that does not use this information in a meaningful way to solve the problem will cause you to receive zero credit for the entire problem.*

2. Suppose we had a binary search tree where each node’s key is a letter of the alphabet, and the comparison between two letters is based on alphabetical order.

The relevant portion of alphabetical order, smallest to largest, is ABCDEFGHIJKLM. How many such binary search trees do we have that obey all of the following properties?

- The tree contains keys that consist of only the letters A through M, inclusive (thirteen letters total).
- Each of these letters appears exactly once as a key in the tree.
- The key of the root node of the tree is the letter ‘J’
- The key of the root’s left hand child is the letter ‘E’

In 1-3 sentences, explain how you got your answer. A response that does not show your work in a meaningful way will receive zero credit for the problem.

*Hint: Draw out, as generally as you can, every tree that meets these requirements.*