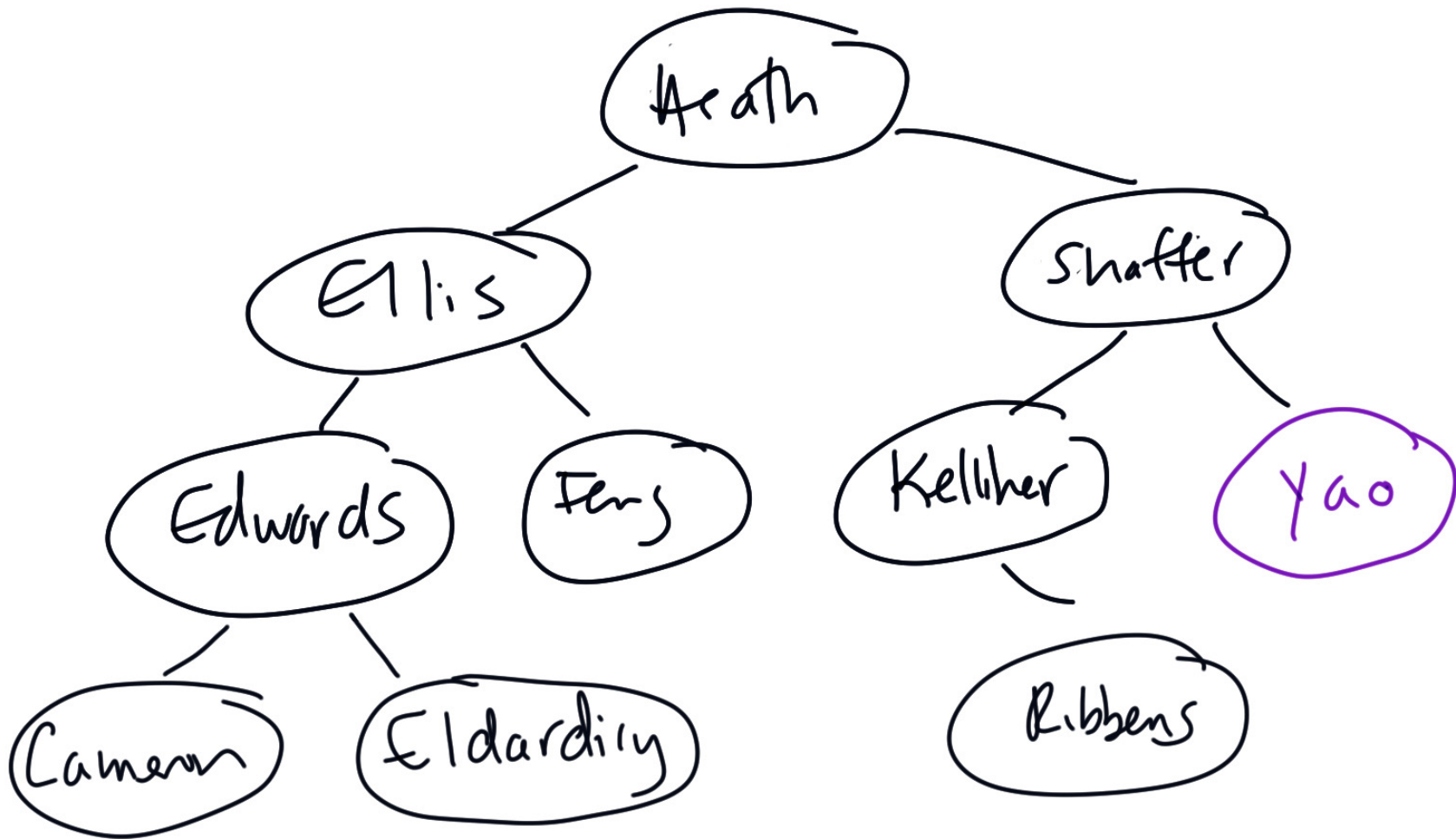
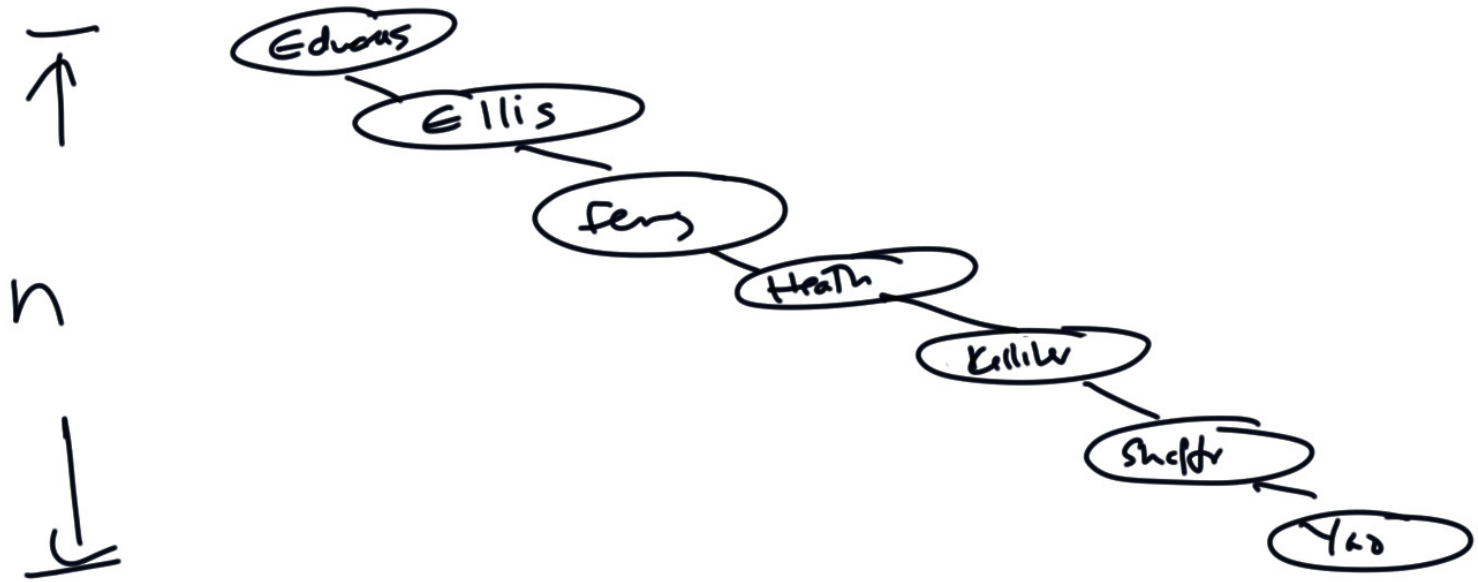
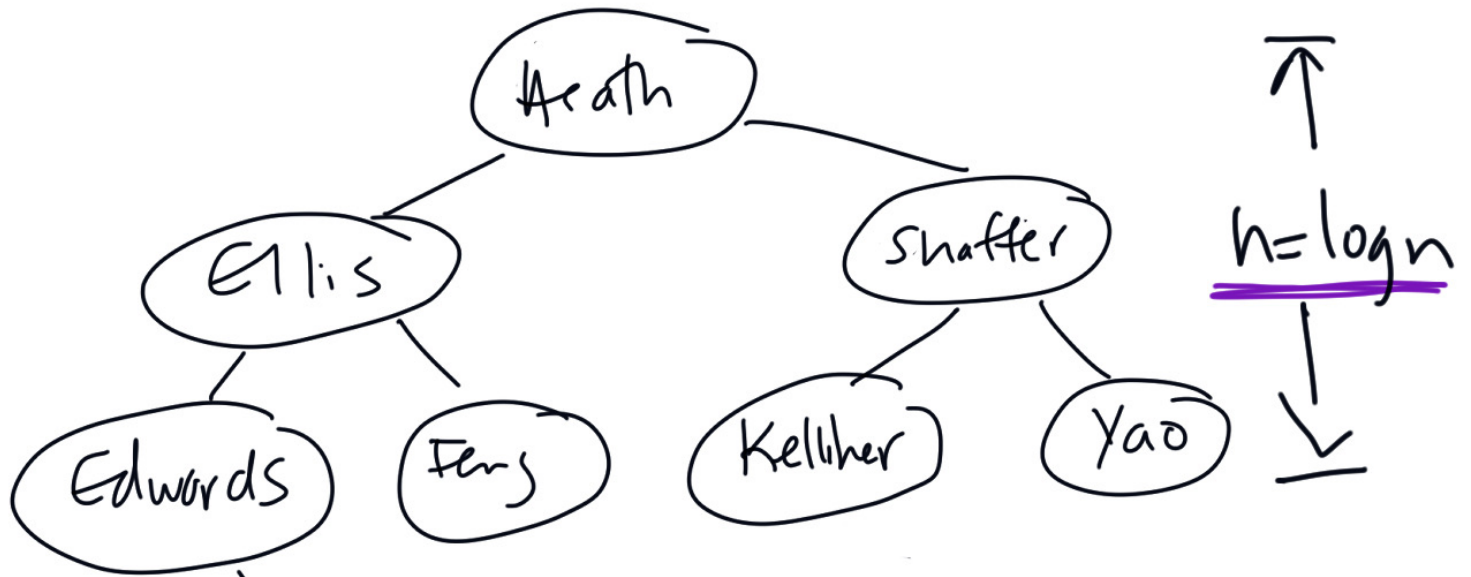


```
167Ⓞ /**
168     * Internal method to find an item in a subtree.
169     *
170     * @param data
171     *         is item to search for.
172     * @param node
173     *         the node that roots the tree.
174     * @return node containing the matched item.
175     */
176Ⓞ private BinaryNode<T> find(T data, BinaryNode<T> node) {
177     if (node == null) {
178         return null; // Not found
179     }
180     else if (data.compareTo(node.getData()) < 0) {
181         // Search in the left subtree
182         return find(data, node.getLeft());
183     }
184     else if (data.compareTo(node.getData()) > 0) {
185         // Search in the right subtree
186         return find(data, node.getRight());
187     }
188     else {
189         return node; // Match
190     }
191 }
```



data  
"Edwards"

"Edwards"	Edwards
"Edwards"	Ellis
"Edwards"	Heath



# Order in Which Nodes Are Added

- Order in which you add entries to a binary search tree affects the shape of the tree
- If you add entries into an initially empty binary search tree, do not add them in sorted order.

# Efficiency of Operations (Search)

