Hyoin Lee (20102711)

Step 1 – Knowledge Question (20-50 words)

In your own words, describe what a Binary Search Tree (BST) is. In addition, describe two important properties of a BST: depth and height. How are they different?

A Binary Search Tree is one of data structures for organising and storing data in a sorted manner. Each node in a Binary Search Tree has at most two children, a left child and a right child, with the left child containing values less than the parent node and the right child containing values greater than the parent node.  
Depth of a node in BST is the number of edges on the longest path from the root node to a certain node, while height is the number of edges on the longest path from the root node to a leaf node which means the maximum depth of the tree.

Step 2 – Knowledge Question (50-80 words)

In your own words, describe how an algorithm to find an item in a Binary Search Tree works.

Algorithm starts at the root node of the BST. It keeps comparing the target item with the value of the current node. If the target item is equal to the value of the current node, the searching process is successful, and the algorithm return the node containing the target item. If the target item is less than the value of the current node, move to the left child of the current node. If greater, move to the right child of the current node. Algorithm keeps repeating this process until the target item is found or the current node is null.

Step 3 – Knowledge Question (20-60 words)

In your own words, describe what a balanced BST is.

Balanced BST represents a BST that has relatively even number of edges in both left and right subtrees, so it looks more like a proper tree. The difference in height between the left and right subtrees of any node is limited to at most one.

Step8 – Knowledge Question

With the newly balanced BST, how many steps does it take at most to find an existing item in the search tree?

With a balanced binary search tree, the maximum number of steps required to find an existing item in the search tree is logarithmic with respect to the number of nodes(n) in the tree. So, this maximum number of steps is typically said as 0(log n).

In binary search, the algorithm repeatedly divides the search space in half based on whether the target item is bigger or smaller than the current node value. As a result, with each iteration, the algorithm eliminates half of the remaining items to search through. This halving of the search space leads to a logarithmic growth in the number of iterations required to find the target item.