void ideaBank::query(string keyWord)

We are taking a snippet of code that defines the complexity of the algorithm which deals with the AVL tree itself. Any other subsidiary function contained within this code is additional information that adds to the complexity of the search function [and is unnecessary]. The reason why the value is bigO(k) or bigO(log2n) is related to the data structure contents that holds the ideas that are contained within the vector. If we analyze the search algorithm during failure, it represents something which is a suitable representation of the search complexity.   
  
Complexity: log2n + 9 + 4k OR log2n + 2

Code Snippet:

if (storage.size() == 1)

{

Index indexBuffer; //create buffer

if (iDAVLTree.AVL\_RetrieveV2(keyWord, indexBuffer) == true)

{

cout << "Key word: " << keyWord;

cout << endl;

cout << "Matches [ids] : ";

for (int i = 0; i < indexBuffer.idList.size(); i++)

{

cout << indexBuffer.idList[i] << " ";

}

cout << endl << endl;

}

else

{

cout << "No entry matching request..." << endl;

}

}