HashPipe Report

The whole structure is represented by a Node(the most top left element). The Node keeps the key/value pair, the level the node is on and 2 references for the right and down element respectively. This was chosen because it is easy to simulate the pointer structure, also having a logarithmic complexity.

The size() method runs in constant time, as it returns a variable incremented each time an item is inserted.

The put() method inserts a key on h levels, where h is the height of the key. It starts from the top of the pipe and goes right and down until it reaches the first place the item should be inserted(highest level). In order to insert from top to bottom, and pointing down to a node before it's created, I used a recursive function(insertNode). Because the distribution of hashing, the running time is in practice O(log N), but in the worst case, where the elements have the same heights, the complexity is O(N).

The get() and floor() methods use the same technique, going right/down until they find the required key, running in logarithmic time.

Due to the logarithmic complexity, all the tests have been passed on codeJudge.