|  |  |  |
| --- | --- | --- |
| Tree |  |  |
| Root | Node |  |
| numOfLeaf | Int |  |
|  |  |  |
| **Node** |  |  |
| parent | Node |  |
| rChild | Node |  |
| mChild | Node |  |
| lChild | Node |  |
| key | <K> | Internal node->key = max key of descendants  Leaf node->key |
| value | <L> | Internal node->value = sum of descendants  Leaf node -> value |
| numOfDesc | Int | Number of descendants |
| numOfChild | Int | Number of children |

Extreme Cases:

sumValuesInterval

* If key 1 > key 2 SOLUTION: if statement
* If there are no keys found in the given range. Needs to be outputted null. Solution: Save last highest key that was checked. If they are the same, return null.