

# Exercise 5

---

Implement the skip list data structure according to the interface given in the file `SkipList.h`. This interface mimics the interface of the STL container `std::map`.

1. The interface given in the file `SkipList.h` represents an associative container, i.e. a container which stores pairs (key,value). A key is utilized to find the corresponding value.
2. The class `Compare` contains the method  
`bool operator() (const Key&, const Key&),`  
which is used to compare two keys and defines a strict weak ordering of keys.
3. The operator  
`bool operator==(const Key&, const Key&)`  
might not exist. Keys `k1` and `k2` are considered equivalent if  
`!Compare() (k1,k2) && !Compare() (k2,k1).`
4. The data structure cannot contain two elements with equivalent keys.
5. As in the class `std::map`, the method `insert()` returns a pair. Its first element is the pointer to the newly inserted item or to the already existing item with an equivalent key. The second element of the pair is `true` if a new item was inserted and `false` otherwise.
6. Iterators should be implemented according to the interfaces given in the file `SkipList.h`.

Reference:

[http://en.wikipedia.org/wiki/Skip\\_list](http://en.wikipedia.org/wiki/Skip_list)