

ITU Computer Engineering Department
BLG 223E Data Structures, Summer 2021
Homework #9
Due August 17, 2021 23:59

Definition

You are asked to implement a basic **non-duplicated sorted** integer skip list that can perform operations provided in a file. Please pay attention to the following details in your implementations:

- Your skip list nodes should follow the following definition. You may add additional private methods/fields when necessary.

```
class MultiNode{
    private:
        int height;
        int data;
        vector<MultiNode*> next;

    public:
        //Other methods
};
```

- Since skip lists are probabilistic data structures, the seed that you should use in your implementations will be given as an input. Please use the following function **once** at the beginning of your program with the provided seed value:

```
srand(seed);
```

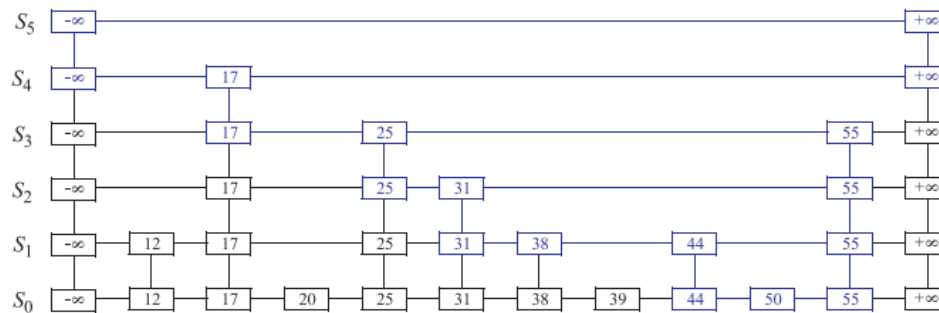
Afterwards, you can use rand() function to generate node heights.

- Your implementations should contain initial and terminal sentinels.
- Your skip list should hold integers (including negative numbers and zero) in an incrementally sorted way. Your list **should not contain** duplicate values.
- You may influence from the algorithms provided in Section 9.4. But please pay attention that the skip list structure in your book is a little different from the homework definition.
- Efficiency is important in this homework. It should be okay if you implement the efficient search functionality as described in the lecture.

Input-Output

Your program should accept a filename as a command line parameter. In the accepted file:

- First line is going to contain a single number that gives the seed for your random number generation.
- Second line is going to contain the height of your skip list followed by the LIST keyword. This number determines the maximum number of next links you may have in your node.
- Following lines may include one of the following commands:
 - ADD keyword followed by **at least** one integer that should be added to your skip list.
 - DEL keyword followed by **at least** one integer that should be removed from your skip list.
 - CONTAINS keyword followed by **at least** one integer that should be queried to be contained by your skip list. As the query result each element should be output separately with either an " IN " or an " OUT " string concatenated with the number. See the provided test cases for examples.
 - PRINT command prints the contents of the list. In the printed content each level is going to be printed with using the number of the level it belongs to as a prefix. Lowest level would be numbered as 0. An example is given below:



Output when printed:

L0: 12 17 20 25 31 38 39 44 50 55 L1: 12 17 25 31 38 44 55 L2: 17 25 31 55 L3: 17 25 55 L4: 17 L5:

See the provided test cases for more examples.

- CLEAR command clears the contents of the list.

Input will not contain syntactical errors. However it may contain removal of non-existent elements. See the provided test cases for examples.

Deliver

Please zip and deliver the directory structure defined below:

- HW9: Topmost folder, that will contain all the folders in your submission. No other files should be present under this folder in your submission.
- HW9/src: Contains all the *.cpp files
- HW9/src/main.cpp: Contains your main function.
- HW9/src/MultiNode.cpp: Contains your skip list's node class definitions.
- HW9/src/SkipList.cpp: Contains your skip list class' definitions.
- HW9/include: Contains all the header files you use
- HW9/bin: An empty directory that will contain objective files when your project is compiled

Please check the calico test file in the homework definition to see how your files will be compiled and tested.

Restrictions and Guidelines

- Compilation environment: Only the code that can be compiled by the environment of the container definition provided in ninova will be accepted.
- Testing of your program will be performed using Calico (<https://calico.readthedocs.io/en/latest/>). Test cases that will be used to test your homework is provided as an attachment in ninova.
- **STL usage is not allowed. Only exception is the next vector provided in the MultiNode definition.**
- **This homework is for individual submissions.** Any kind of code sharing or code adaptation from an external source is strictly forbidden. Submitted code will undergo a plagiarism check process and plagiarsim penalties may be given if the submitted code's similarity is approved by the instuctor.
- Make sure you write your name and number in all of the files of your project, in the following format:
/* @Author
Student Name:<studentname>
Student ID :<studentid>
Date:<date>*/
- Only electronic submissions through Ninova will be accepted no later than deadline
- Use comments wherever necessary in your code to explain what you did.