Prof. Dr. Alexander Bockmayr,
Prof. Dr. Oliver Serang,
Christopher Pockrandt

November 12, 2015

Algorithms - Programming Exercices

WS 2015/16

Exercise 1
Due date: 26.11.2015 until 11 pm.

Skiplists

1. Implement a class `skipList` that implements a skip list with the following public member functions:
   - `size_t numElem()`: returns the number of elements currently stored in the list.
   - `int find(int x)`: report the maximal element in the list that is at most equal to x.
   - `bool insert(int x)`: insert x into the list. On success return `true`. If x was already in the list return `false`.
   - `bool remove(int x)`: delete x from list. If deleted return `true`. If x was not in the list return `false`.
   - `skipList(const std::vector<int> & init)`: Constructor takes a `std::vector<int>` argument which contains the initial elements (not sorted) to be stored in the list.
2. The class shall be defined in a file `skipList.h` and member functions shall be defined in `skipList.cpp`.
3. On some generated datasets with different size compare the runtime of insertions and removal and also evaluate the height and size of your lists.
4. There will be a skeleton file exercise3.cpp in the `data` directory, which can be used as minimal sanity check. It requires that the source `skipList.cpp` and `skipList.h` exist.
5. Together with your program you have to provide a short application note (1-2 DIN-A4 pages; pdf) describing the implementation and an evaluation of the runtimes and discussing the remarks above.
6. The due date is Thursday 26^{th} of November 2015 until 11 pm at the latest. All versions submitted later than this time stamp won't be assest.

7. The program shall run on a linux pool machine (see the wiki for additional information `http://www.mi.fu-berlin.de/w/AgMathLife/ProgrammingExercisesWS15`).

8. The compiler flags shall be set to `-pedantic -Wall -ansi -O3`.

You can score 4 pts.. 3 pts. for the program and 1 pt. for the application note.
3 pts. = The program compiles and runs successfully with no errors.
2 pts. = The program compiles and contains minor errors.
1 pt. = The program compiles and contains some critical errors.
0 pts. = The program doesn't compile or does not meet the requirements.