

Homework 5

Out: 10.24.16

Due: 11.2.16

1. [Selection 10 points]

An algorithm is in-place if it uses only a small amount of memory in addition to that needed to store the input. Is the linear time selection algorithm provided in class an in-place algorithm? If not, explain how you would modify it.

2. [Binary Search Tree 15 Points]

- Draw the binary search tree which results from inserting the following numbers, one after the other, from left to right: 3 12 2 8 1 1.5 6 0.5 7 20.
- Show the resulting binary search tree after deleting 8. Explain.
- Show the resulting binary search tree after deleting 8 and 12. Explain.

3. [Hashing, 25 points]

Insert the following keys into an initially empty hash table of size 10:
4371, 1323, 6173, 4199, 4344, 9679, 1989.

Show the resulting hash table, with the hash function $h(x) = x \bmod 10$.

- Using open hashing (i.e. chaining).
- Using closed hashing (open addressing) with linear probing.
- Using closed hashing (open addressing) with quadratic probing, $C_0=C_1=0, C_2=1$.
- Using double hashing with the secondary hash function $h_2(x) = 7 - (x \bmod 7)$.
- Using Cuckoo hashing with the above secondary hash function $h_2(x) = 7 - (x \bmod 7)$.

4. [Cuckoo Hashing, 50 points]

Implement a Cuckoo hash table in which two separate tables are maintained. Each table should have a fixed size of 13. Use the hash function $h_1(x) = x \bmod 13$ for the first table, and $h_2(x) = 11 - (x \bmod 11)$ for the second table.

Your program should read an input file, *input.txt*, consisting of one input per line, insert the input to the table in order, and print out the final content of the hash table in the provided format. Your implementation should utilize the provided *CuckooHashTable.h* file as is, without any edits. Note that the hash tables are implemented as a 2x13 matrix (vector of vectors), and that a print method declaration is provided, for you to implement.

Your program should be able to detect if an insert results in an infinite loop in the hash, and exit with an error message.

Sample *input.txt* file content:

```
1
4
6
27
```

123
14
17
195

Sample output for the above input:

Table 1:

195
14
-
-
17
-
123
-
-
-
-
-

Table 2:

-
-
-
-
-
6
27
4
-
-
1
-
-

Submit your solution, which should be contained in a single *CuckooHashTable.cpp* file, and should be compiled with the provided *CuckooHashTable.h* and *main.cpp* files. To compile these three files on the lab computers, use the provided *makefile*, and type the following two commands:

```
> module load gcc  
> make
```

Make sure to write your name in a comment at the top of the program, along with an explanation of your approach.