Introduction to Computer Science
HW #5
Due: 2019/05/22

Homework Rules:

Hand-written homework can be handed in **before lecture starts**. Otherwise, you may contact the TA in advance and then bring the hardcopy to the TA in MD-631 (please send e-mail in advance).

As for the programming part, you need to upload it to CEIBA before the deadline. The file you upload must be a **.zip** file that contains the following files:

README.txt

HW01_b07901XXX (a folder that contains all .cpp & .h as required),

If you program in C/C++:

1. Do not submit executable files (.exe) or objective files (.o, .obj). Files with names in wrong format will not be graded. You must **remove any system calls**, such as system ("pause"), in your code if any.
2. In README.txt, you need to describe which compiler you used in this homework and how to compile or execute it (if it is in a "project" form).
3. In your .cpp files, we suggest you write comments as detailed as you can. If your code does not work properly, code with comments earns you more partial credits.

If you program in Python:

1. Do not submit .pyc files. Files with names in wrong format will not be graded.
2. In README.txt, you need to describe which Python version you used in this homework (e.g. Python 3.5) and how to execute it.
3. In your .py files, we suggest you write comments as detailed as you can. If your code does not work properly, code with comments earns you more partial credits.

Introduction to Computer Science
HW #5
Due: 2019/05/22

Chapter 8 Review Problems (8% each):

23, 25, 28, 30, 34

M1 (5%): Draw a binary search tree (with keys only) which pre-order traversal yields 4, 3, 2, 1, 10, 9, 8, 5, 7, 6, 11, 12.

M2 (5%): What's the in-order traversal sequence for the BST in **M1**?

Programming Problem (50%):

Write a class **Heap** which is derived from **AbsHeap** (in "heap.h/ heap.py"). Implement the two virtual functions(pop() and push()) in AbsHeap. This is a binary min heap, so pop() always returns the element with the minimum key. You may assume **all keys are distinct**. You need to complete your code in "heap.h/heap.py".

For your test convenience, you may use testHeap to test your heap. To use them, rename "inputX.txt" to "input.txt", and check the corresponding "outputX.txt".

BONUS (5%)

Given a preprocessed article which is lowercase and non-punctuation. Your task is to find the k most frequent words. Your code reads inputs from a file "bonus.txt", where the 1st line is *k*, the 2nd line is article with each word separated by " ".

Your output should be in descending order of the most k frequent words. If two words have the same frequency, then the word with the lower alphabetical order comes first. Save your code in "bonus.cpp/ bonus.py".

Input:	2 **i love i2cs i love coding i am happy**
Output:	i love

Introduction to Computer Science
HW #5
Due: 2019/05/22

Please use the min heap to complete it. Your code needs to handle *n*=1000000 in reasonable time (less than 5 seconds). The following is the information you can refer to.

- C++ Map STL:
  https://mropengate.blogspot.com/2015/12/cc-map-stl.html
- Heap optimization:
  https://leetcode.com/problems/top-k-frequent-words/

How to submit:

Compress all your files into one single file and then submit electronically via Ceiba by the due date.