Homework 2
Com S 311
Due: Monday, September 23, 11:59 pm

5% bonus for submission by Sunday, September 22, 11:59 pm

10% penalty for late submission by Tuesday, September 24, 11:59 pm

No submissions accepted after Tuesday, September 24

You may assume that `add, delete, search` operations take $O(1)$ time with Hash Tables. For all algorithm/data structure design problems, part of the grade depends on efficiency.

1. (25 pts) Let $S = \{44, 21, 10, 22, 12, 3, 19, 1, 4\}$.

   - Draw the Binary Search Tree obtained by inserting the elements of S in the order they appear above.
   - Draw the hash table (using chain hashing) obtained by adding elements of S into a hash table of size 5. Use $(7x + 2)\%5$ as the hash function.

2. (30 pts)

   (a) Given any set S of unique integers, and an integer $x \in S$, $succ(x)$ is defined to be the smallest element in S that is larger than x. If there is no such element, then $succ(x)$ returns a null value. Suppose that S is the set of keys of a hash table H that is implemented using chaining. Give an algorithm that, given H and x as inputs, returns $succ(x)$. Report the run-time of your algorithm.

   (b) Using the *succ* function above, here is a strategy for constructing an ordered array of keys

```
n = number of elements in H
arr = new array of length n
x = -infinity
i = 0
while i < n
  x = succ(x)
  arr[i++] = x
```

   What is the runtime of this algorithm? What would be a better way to produce an ordered array of keys?

3. (25 pts) Let G be a graph with vertices identified by integers $\{1, 2 \cdots, n\}$. The edges of G can be represented as a set E of ordered pairs $\langle i, j\rangle$, representing that there is a directed edge from i to j. We define a graph G to be *undirected* if $\langle i, j\rangle \in E$ always implies that $\langle j, i\rangle$ is also in E. Suppose you are given G as an array of pairs representing all edges in E. Give an algorithm to detect whether the graph is undirected or not. Report the run-time of your algorithm.

4. (50 pts) This problem is about the array-based binary heap implementation of a priority queue. Let `H` denote the backing array, I.e H is a binary Min-heap Array. You should assume that the universe of possible keys is extremely large relative to the size of the heap (i.e., it is not practical to use a `Position` array as described on p. 65 of the text).

   (a) Give an algorithm for a new operation `decreaseKey(index, delta)` that subtracts `delta` from the key located at the given index in `H`, and restores the heap ordering. Report the time taken by the algorithm.

   (b) Suppose we also want an operation `findKey(v)` that returns the index of a key `k` in H (if `k` appears more than once, then it does not matter which index is returned). Describe a modified implementation of the heap in which your `findKey(v)` operation is $O(1)$, finding the minimum is still $O(1)$, and the operations to insert a key or remove the minimum are still $O(\log n)$.

GUIDE LINES:

- Please write your recitation number, time and TA name.

- It is important to know whether your really know! For each problem, if you write the statement "I do not know how to solve this problem" (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere between 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem and you must not submit any attempted solution.

- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.

- When proofs are required, you should make them clear and rigorous.

- Any concerns about grading should be made within one week of returning the homework.

- **Please submit your HW via Canvas. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solutions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. Name your file *YourNetID-HW2.pdf*. For example, if your net id is `bondj`, then the file name should be *bondj-HW2.pdf*. HW's submitted in incorrect format (non pdf, incorrect file name etc) may incur a penalty of 20%**

- If you hand writing is not legible or the quality of the scan is poor, your homework will not be graded.