Data Structures
Homework #5

Due: Dec 16, 2025 (Before Class)

1. Given two binary search trees T_A and T_B, please make these two binary search tree united as one binary search tree. Your solution can not be simply deleting each node in one tree and inserting them into another one sequentially. Please also show the time and space complexity of your approach.

2. Consider an implementation of heap H by means of a linked structure.

   (a) Suppose that we use a reference to the last node in the structure. Please show how to update the reference to the last node after the operations `insert()` and `removeMin()` $O(\log n)$ time, where $H.size = n$ is the current number of nodes of H. Be sure and handle all possible cases.

   (b) Design an $O(\log n)$-time algorithm for finding the last node of H with $H.size = n$ nodes in the implementation of a heap by means of a linked structure that does not keep a reference to the last node.

3. Let S be a set of n points in the plane with distinct integer x- and y-coordinates. Let T be a complete binary tree storing the points from S at its external nodes, such that the points are ordered left-to-right by increasing x-coordinates. For each node v in T, let $S(v)$ denote the subset of S consisting of points stored in the subtree rooted at v. For the root r of T, define $top(r)$ to be the point in $S = S(r)$ with maximum y-coordinate. For every other node v, define $top(v)$ to be the point in S with highest y-coordinate in $S(v)$ that is not also the highest y-coordinate in $S(u)$, where u is the parent of v in T (if such a point exists). Such labeling turns T into a *priority search tree*. Describe a linear-time algorithm for turning T into a priority search tree.

4. (50 pts) (**Programming Exercise**)
   This exercise is about to implement a ***heap*** by means of a ***linked structure*** with Python, in stead of using an array with level-numbering. The sample definitions for the node and heap classes are included in the provided `HW5.jpynb` file. Each node has an entry with two attributes, `key` and `item`, where `key` is an integer representing the priority and `item` is a string. A smaller key has a higher priority. The methods in the heap include

   `removeMin()`: This is to remove the object with the minimum key from the heap;

   `Insert(v)`: This is to insert a heap node into the heap;

   `upwardHeapify(v)` This method performs upward adjustment from the current node v;

   `downwardHeapify(v)`: This method will adjust the heap downward from heap node v;

   `printHeapPreOrder(v)`: For management or verification, we print the heap in pre-order from node v with this method.

   The Python program starts with function `HeapwithEntriesInserted()` which reads the input file, `inFile.txt`. In the input file, each line contains only one entry: key and string

and as follows:

```
10 mary
25 john
35 mars
50 lowe
```

An example of input file is also provided on the course website. The execution should be as below and corresponding list operations is also shown in the given `HW5.jpynb` file.

```
>>> HeapwithEntriesInserted()
Heap size= 4 The highest priority is 10
pre-order traversal:
Node [ 10 mary ]
Node [ 25 john ]
Leaf [ 50 lowe ]
Leaf [ 35 mars ]
deleteMin
deleteMin
deleteMin
deleteMin
The heap is now empty
deleteMin
The heap is empty and no entry can be removed
insert 35, resume
insert 15, second
insert 20, fourth
Heap size= 3 The highest priority is 15
pre-order traversal:
Node [ 15 second ]
Leaf [ 35 resume ]
Leaf [ 20 fourth ]
```

About submitting this homework

1. For problem 1, 2 and 3, Please

   (1) **write by hand** all of your solutions on the papers of size A4,

   (2) leave you name and student ID on the first page, and

   (3) hand in your solutions for problem 1, 2, 3 and 4 to me in class

2. For problem 4, things to be submitted include:

   (1) please finish each problem right after the problem description in the `HW5.ipynb` file provided on the **i-school(Plus)** (`https://istudy.ntut.edu.tw/learn/index.php`) platform with sample given input file; and

   (2) please upload the completed `.ipynb` file with the filename as `HW5_studentID.ipynb` to **i-school(Plus)**

3. **Late work** is not acceptable. Remember, the **deadline** is the midnight of **December 16**, 2025.

4. Honest Policy: We encourage students to discuss their work with the peer. However, each student should write the program or the problem solutions on her/his own. Those who copy others work will get 0 on the homework grade.