Data Structure HW#5

Part I Paper Work

1 (Exercise 2 of Chapter 5.2) What is the maximum no. of nodes in a k-ary tree of height h? Prove your answer.

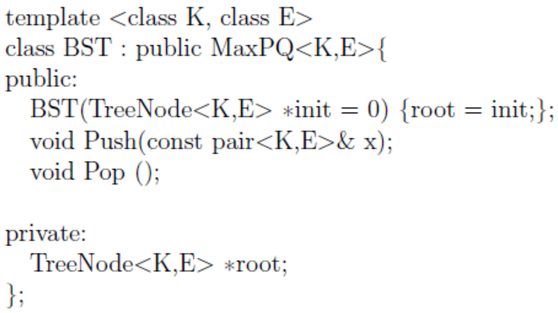
2 (Exercise 4 of Chapter 5.2) Extend the array representation of a complete binary tree to the case of complete trees whose degree is $d, d > 1$. Develop formulas for the parent and children of the node stored in position i of the array.

3 (Exercise 6 of Chapter 5.3) Write a non-recursive version of function $Preorder$ (Program 5.2).

4 (Exercise 2 of Chapter 5.5) Write a C++ forward iterator, $PreorderIterator$, for threaded binary trees. You should sequence through the nodes in inorder.

5 (Exercise 3 of Chapter 5.7) Suppose that each node in a binary search tree also has the field $leftSize$ as described in the text. Write a function to insert a pair into such a binary search tree. The complexity of your function should be $O(h)$, where h is the height of the search tree. Show that this is the case.

6 (Exercise 9(b) of Chapter 5.7) Notice that a binary search tree can be used to implement a priority queue. The C++ class definition for a max priority queue that represent the priority queue as a binary each tree is as below:



```
template <class K, class E>
class BST : public MaxPQ<K,E>{
public:
   BST(TreeNode<K,E> *init = 0) {root = init;};
   void Push(const pair<K,E>& x);
   void Pop ();

private:
   TreeNode<K,E> *root;
};
```

It is a publicly derived class of $MaxPQ$. Write a C++ function for the virtual functions, Pop and $Push$ of $MaxPQ$.

7 (Exercise 2 of Chapter 5.8) Write a function to construct a winner tree for k records. Assume that k is a power of 2. Each node at which a tournament is played should store only a pointer to the winner. Show that the time complexity is $O(k)$.

8 (Exercise 5.2 of Chapter 5.10) Write a function Splitting Find that uses **path splitting** for the find operation instead of path collapsing. The is defined below:
Definition [***Path Splitting***]: In the path splitting, the parent pointer in each node (except the root and its child) on the path from I to the root is changed to point to the node's grandparent.

Note 1: You can reuse one-side used papers but must in A4 size. Please hand in your assignments to the TAs (R721, Applied Science &Technology Building) by the deadline.

Note 2: Please compress your code as well as the snapshot your execution results into a zip file and upload it to **iLearning**.