

## Data Structure Homework E : BST and Heaps

- Objective
  - Implement a binary search tree with linked lists
  - Implement a heap with arrays and queues
  - Provide following functions: insert, delete, search and print for the binary search tree and insert, delete max and print for the heap.
- Description
  - You can combine the BST and heap to one program(take a look at the sample.exe), or spilt it into two programs.
  - The functions you need to provide for the binary search tree are listed as below:
    - ◆ insert() – let a user enter a number and insert it into the tree.
    - ◆ delete()– let a user enter a number and delete it from the tree.  
(if no such a number in the tree, print the error message)
    - ◆ search() – let a user enter a number, search it, and print the success/fail message to the user.
    - ◆ printInfixOrder() ,printLevelOrder()
      - in binary search tree, the output will be in increaing order in infix order.
  - The functions you need to provide for the heap:
    - ◆ insert() - let a user enter a number and insert it into the tree.
    - ◆ deleteMax() – delete the max number in the heap and print the result.
    - ◆ printHeap() – print the queue which is used to store data in the heap.
  - Other functions you may use:
    - findMin() – you may need it when doing deletion in BST
    - push(), pop() – you may need it when using the queue
- Grade policies
  - 5% - Source code can be compiled without any error
  - 40% - BST : insert 10%, delete 10%, search 10% , print 10%
  - 35% - Heap: insert 10%, delete max 15%, print 10%
  - 20% - readme file, code style, and comments in source code
- Turn in
  - System
    - ◆ Turn in files to the workstation : csie0.cs.ccu.edu.tw
    - ◆ Command: turnin ds.hwE [files...]
    - ◆ This source code will be compiled and tested on the workstation
  - Source code

- ◆ Source code with appropriate comments
- Report
  - ◆ A document named “readme.txt” or “readme.doc” or “readme.pdf”.  
you should describe the details of your project in your readme file
- Output example:
  - ◆ After insert 10, 18, 15, 7, 4, 9 , 8, 6, 2 into binary tree:

```

<I>nser t a number.
<D>elete a number.
<S>earch a number.
<P>rint in infix&level order.
The tree in infix order : 2 4 6 7 8 9 10 15 18
The tree in level order : 10 7 18 4 9 15 2 6 8
請按任意鍵繼續 - - -

```

- ◆ After insert 5, 8, 2, 10, 13, 20, 26, 19, 33 into the heap:

```

<I>nser t a number.
<D>elete the max number.
<P>rint the heap queue.
The data in the heap queue : 33 26 20 19 8 2 13 5 10
請按任意鍵繼續 - - - ■

```

- ◆ After delete the max of the above heap:

```

<I>nser t a number.
<D>elete the max number.
<P>rint the heap queue.
After delete the max : 26 19 20 10 8 2 13 5
請按任意鍵繼續 - - -

```