資料結構期中考 2017.4.24

滿分 110 分

(1) (20%, 4% each) True of False. Please provide brief explanation. Otherwise, no points will be given.

(1a) Linked list is an efficient data structure for random access.

(1b) An algorithm with time complexity $\Theta(n)$ runs faster than another algorithm which has time complexity $\Theta(n^2)$.

(1c) An algorithm with time complexity $O(n)$ also has time complexity $O(n\log n)$.

(1d) Given a max heap with **n** nodes. A *Push* operation takes $O(\log n)$ time, and a Pop operation takes $O(1)$ time.

(1e) Inserting one node into an AVL tree will need at most 1 rotation operation to re-balance.

(2) (8%) Consider the AVL tree resulting from sequentially inserting the following numbers into an empty AVL tree: 40 60 55 15 20 2 25 30.

(2a) **(4%)** What is the sum of the numbers in the leaf nodes in the AVL tree?

(A) 90 (B) 130 (C) 155 (D) 160 (E) 170

(2b) **(4%)** Please draw this AVL tree after inserting 99 into the above AVL tree.

(3) (6%) Please compare and sort the time complexity:

$O(n^2)$、$O(n^3)$、$O(2^n)$、$O(1000^{3000})$、$O(\log n)$、$O(n\log n)$

(4) (6%) Consider the following program. What is the result of data[3]?

```
#include <stdio.h>
int data[] = {8, 26, 5, 77, 1, 61, 11, 60, 15, 49, 19};
 int n = 6;
void a(int r)
{
        int e, j;
        e=data[r];
        for(j=2*r;    j<=n; j*=2) {
                if (j<n&&data[j] >data[j+1]) j++;
                if (e>=data[j]) break;
                data[j/2] = data[j];
        }
        data[j/2] = e;
}

int main()
{
        int i;
        for(i = n/2; i>=1; i--)
            a(i);
}
```

(5) **(6%)** Assume there are ***n*** numbers to be searched. (No proof needed)

(5a) **(3%)** What is the worst-case time complexity of **binary search** ?

(5b) **(3%)** What is the worst-case time complexity of search in a **binary search tree**?

(6) **(5%)** Which of the following statement is **correct** ? (多選) No explanation is needed.

(A) A max(min) heap is a complete binary tree.

(B) Recursive algorithm **should not** return a value or stop at certain condition.

(C) In a priority queue, the element to be processed/deleted is the one with highest (or lowest) priority.

(D) Singly linked list is suitable for random access.

(E) Sparse Matrix has many zero elements.

(7) **(12%)** Read the following sentences and answer the questions. Note that explanation is necessary.

The following program is a count-controlled loop going from 1 to 5. At each iteration, the loop counter is either printed or put on a stack depending on the result of Boolean function `RandomFunc ()` (Here the behavior of `RandomFunc ()` is immaterial (無關緊要的)). At the end of the loop, the items on the stack are popped and printed. Due to the logical properties of a stack, this program cannot print certain sequences of the values of the loop counter. You will be given an output and asked to determine if the program could generate the output.

```
For ( count =1 ; count <=5 ; count ++ )
   if ( RandomFunc() )
      cout << count ;
   else
      stack.push(count);
while (!stack.empty())
{
   stack.pop(number);
   cout << number;
}
```

(7a) **(6%)** Is the following output possible with a stack : 1 3 5 2 4 ?

Please provide your explanation.

(7b) **(6%)** If the stack is replaced with a queue, i.e.,

```
For ( count =1 ; count <=5 ; count ++ )
   if ( RandomFunc() )
      cout << count ;
   else
      queue.enqueue(count);
while (!queue.empty())
{
   queue.dequeue(number);
   cout << number;
}
```

Is the following output possible with a queue : 1 3 5 2 4 ? Please provide your explanation.

(8) (8%) Please answer the following questions regarding prefix/infix/postfix expressions.

(8a) **(5%)** Consider a postfix expression "***24+6*8-***". Please transform it to infix and explain the steps with the stack.

(8b) **(3%)** Suppose that we use a stack to transform an infix with ***m*** operands and ***n*** operators to its corresponding postfix expression (Note: operators include +-*/). In the following, which one describes the size of the stack most appropriately?

(A) $O(n/m)$ (B) $O(n)$ (C) $O(mn)$ (D) $O(m+n)$

(9) (12%) Consider a tree T whose postorder and inorder sequences are as follows.

Postorder: A, C, E, D, B, H, I, G, F.

Inorder: A, B, C, D, E, F, G, H, I.

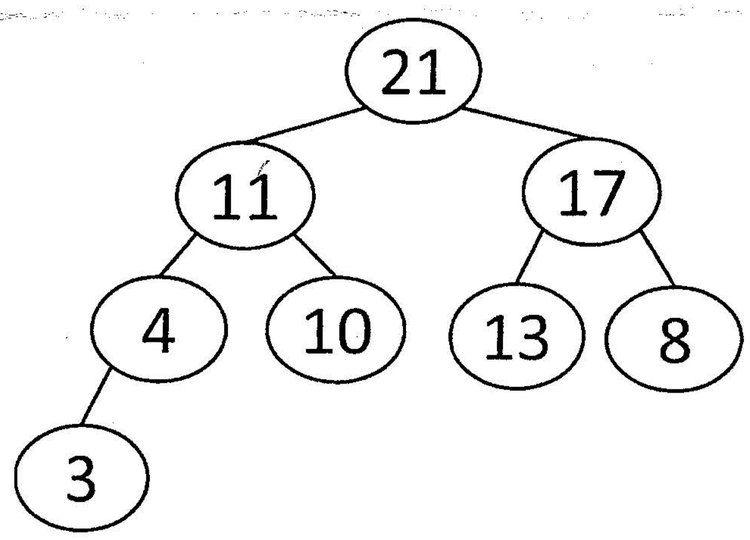
~~(8a)~~ **(6%)** Is the preorder sequence unique?

9a If yes, please show the preorder sequence of T. If no, please present an example.

~~(8b)~~ **(6%)** Explain why postorder and preorder together cannot decide a unique tree structure by

9b example.

(10) (5%) A max heap is given below. The numbers at the nodes are their keys. Now after one insertion operation (the key inserted is 14) and then one deletion operation (recall that a deletion operation in max heap always removes the node with the largest key), <u>which of the following statements is **FALSE**</u> regarding the resulting heap? Here `node(k)` represents the node with key `k`.



(A) The parent of `node(4)` is `node(11)`.

(B) `node(13)` has two children.

(C) The position of `node(10)` in the tree does not change.

(D) `node(17)` is the root.

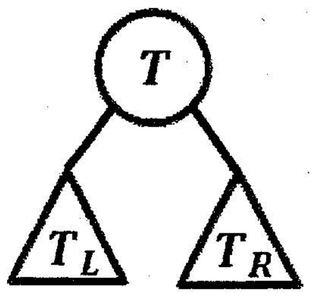
(E) The number of leaf nodes does not change.

(11) (10%) Consider a binary tree and answer the following questions.

11a ~~(10a)~~ **(4%)** Prove that if the height of the binary tree is ***k***, the maximum number of nodes is $2^k - 1$.

11b ~~(10b)~~ **(6%)** Prove that if $n_0 =$ number of leaf nodes, and $n_2 =$ number of degree-2 nodes, then $n_0 = n_2 + 1$.

(12)(10%) Given binary tree T as shown below. Please answer the following questions.



(11a) **(3%)** *Balance factor* of T, i.e., $\mathrm{bf(T)}$, is defined as $\mathrm{bf(T)} = \mathrm{height(T_L)} - \mathrm{height(T_R)}$. Then, what are the two conditions that we can say T is *height balanced*?

(11b) **(7%)** Given an AVL tree T_A with height ***h***, and let ***N(h)*** denote the minimum number of nodes in T_A. Please prove that $\mathrm{h} = \mathrm{O(logN(h))}$.

(13) **(2%)** Please give some comments and the advices to this course.