

CS 2412 MID TERM EXAMINATION

October 17th, 2014

Duration: One and half hours

Name:

Student Number:

Note. There are total 7 problems in this exam set (6 pages). If the space is not enough for your answer, the back side of the pages can be used.

Problem 1. (10 marks)

Indicate which of the following statements are correct:

- (a) Stack is LILO (b) Stack is FILO (c) Binary tree is LILO
(d) Queue is LIFO (e) Queue is LILO (f) List is LOFI

The correct statements are:

Problem 2. (15 marks)

Decide if the following statements are correct or wrong. Give brief reasons.

1. An $O(c^n)$ program is more efficient than an $O(n^c)$ program.
2. A binary sequential search algorithm is an efficient search method for ordered data. But it may not be efficient for random data.
3. Since AVL tree is more efficient than BST, to make a program that outputs 10000 ordered integers after input 10000 random integers, we should use an AVL tree but not BST.

Problem 3. (10 marks)

Change the following infix expressions to postfix expressions and prefix expressions:

- $A * B - C * D$
- $(A + B) * C - D * F + C$

Problem 4. (10 marks)

The combination of n objects taken k at a time can be computed as $C(n, k) = \frac{n!}{k!(n-k)!}$.

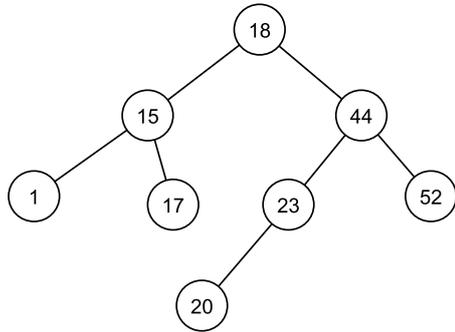
Note that $\frac{n!}{k!(n-k)!} = \frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!}$. Therefore

$$C(n, k) = \begin{cases} 1 & \text{if } k = 0 \text{ or } n = k, \\ C(n-1, k) + C(n-1, k-1) & \text{if } n > k > 0. \end{cases}$$

Use pseudo code to write a recursive algorithm to compute $C(n, k)$.

Problem 5. (15 marks)

Suppose we have an AVL tree as follows.



- Suppose we delete 15 from the tree. Draw the resulting AVL tree.
- Suppose we further delete 17 from the tree. Draw the resulting AVL tree.
- Finally, suppose we next delete 23 from the tree. Draw the resulting AVL tree.

Problem 6. (20 marks)

1. Give an application example which is suitable to use a multilinked list.
2. Use pseudo code to write an algorithm for inserting a node to the multilinked list for your example.

Problem 7. (20 marks)

Suppose a queue has been established which contains about one hundred integers. The ADT source codes are used. Part of the C source codes are as follows.

```
typedef struct node
{
    void*      dataPtr;
    struct node* next;
} QUEUE_NODE;
typedef struct
{
    QUEUE_NODE* front;
    QUEUE_NODE* rear;
    int        count;
} QUEUE;

bool dequeue (QUEUE* queue, void** itemPtr)
{
    QUEUE_NODE* deleteLoc;
    if (!queue->count)
        return false;
    *itemPtr = queue->front->dataPtr;
    deleteLoc = queue->front;
    if (queue->count == 1)
        queue->rear = queue->front = NULL;
    else
        queue->front = queue->front->next;
    (queue->count)--;
    free (deleteLoc);
    return true;
} // dequeue
```

Write a C function: `void printQueue(QUEUE* pQueue)` which prints the data in the queue and deletes the data in the queue using above `dequeue` function. So after call this function, the integers in the `pQueue` are printed and the queue became an empty queue.