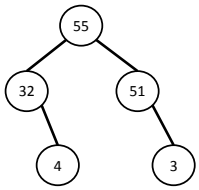
COMP 235102 Data Structures
Spring 2013, CS Dept., NTHU
Midterm Exam
Date: 15 April, Mon (10:10 am – 12:00 pm)

In this paper, there are **105 marks** in total, in which **5 marks** are *bonus marks*. This means that if you answer all parts in all questions, you will get 5 bonus marks for free. Finally, we note that you can describe your answers in English/Chinese sentences.

1. (16%) In the following, please answer (True or False) to each question.
   (a) (1%) $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$.
   (b) (1%) $n! = O(n^n)$.
   (c) (1%) $f(n) = O(g(n))$ *iff* there exist positive constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for all n, $n \geq n_0$.
   (d) (1%) The following is a complete binary tree.



   (e) (1%) Every node in a tree must have one parent.
   (f) (1%) The time complexity of inserting an element into a linked list is $O(1)$ if we already have a pointer points to its previous element.
   (g) (2%) The result of this prefix expression * - 15 * 5 7 + + 30 8 * 11 16 is -4280.
   (h) (2%) We cannot find a binary tree whose preorder and postorder traversals give the same result.
   (i) (2%) We can find a binary tree whose preorder and inorder traversals give the same result.
   (j) (2%) We can use several stacks to implement a queue.
   (k) (2%) There is a Hanoi Tower with 4 disks stay on peg A at the beginning. If we want to move all disks from peg A to peg C, we need to move the disks at least 16 times.

2. (10%) Please answer the following short questions.
   (a) (3%) What is a circular queue?
   (b) (3%) What is a stack?
   (c) (4%) What criteria do algorithms have to satisfy?

3. (8%) Use stack to evaluate the following expression by using the method taught in Chapter 3. For the following two subquetions, please write down the type of notation, *prefix*, *infix*, or *postfix*, the final computed result, and the maximum number of elements stored in the stack

at any time moment during the computation process. (*Note that you are advised to show the intermediate steps during the computation for partial credits in case that you make errors during your computation process.*)

(a) (4%) + - + 3 * - * 5 4 2 1 7 6

(b) (4%) 4 2 5 * 3 - 6 * + 7 - 1 +

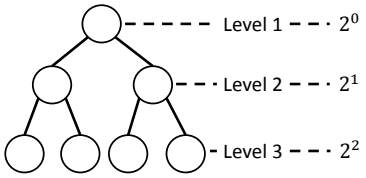
4. (10%) Rank the following functions by order of growth; that is, give an ordered list of the function $g_1, g_2, \ldots, g_{10}$ satisfying $g_1 = O(g_2)$, $g_2 = O(g_3)$, and so on.

$$n,\ \sqrt{n},\ (\frac{3}{2})^n,\ 4^{\log_2 n},\ n\log^2 n,\ n\log n^2,\ n^2\log n,\ n\log\log n,\ n^3,\ 1000.$$

5. (12%) This question is about the binary trees.

(a) (2%) What is the maximum number of nodes at level i?

(b) (4%) What is the maximum number of nodes in a binary tree of depth k?



(c) (6%) Prove that for any binary tree if n_0 = number of leaf nodes and n_2 = number of nodes of degree 2 then $n_0 = n_2 + 1$.

6. (17%) The different order traversal results of binary tree T can help to obtain the original T.

(a) (8%) Suppose that a binary tree T has an inorder traversal result $dbaegchfj$ and a postorder traversal result $dbgehjfca$. Please draw out the tree T.

(b) (9%) Discuss why inorder and preorder or inorder and postorder can decide a unique tree, but preorder and postorder together cannot.

7. (12%) *abcde* is a preorder traversal result of a binary tree T. The following are inorder traversal results of the same binary tree T. Please either answer True and draw out the tree T if the inorder traversal result is possible; or answer False and give the explanation why the inorder traversal result is impossible.

(a) (4%) *bdcea*

(b) (4%) *cdaeb*

(c) (4%) *bcaed*

8. (10%) Following are two operations, *insert* and *delete*, of a circular doubly list. The data type of each node is *struct DATA*. Please complete (a) to (j) in the following two C procedures to *insert* or *delete* one element.

```
struct DATA{
    int value;
    struct DATA *previous; /*Point to the node before this.*/
```

```
    struct DATA *next; /*Point to the node after this.*/
};
/*insert the node pointed by ptrB after the node pointed by ptrA*/
void insert(struct DATA *ptrA,struct DATA *ptrB){
    ptrB-> (a) (1%) = ptrA;
    ptrB-> (b) (1%) = ptrA-> (c) (1%) ;
    ptrA-> (d) (1%) = ptrB;
    ptrA-> (e) (1%) = ptrB;
}
/*delete the node pointed by ptrB*/
void delete(struct DATA *ptrB){
    ptrB-> (f) (1%) = ptrB-> (g) (1%) ;
    ptrB-> (h) (1%) = ptrB-> (i) (1%) ;
    free( (j) (1%) );
}
```

9. (10%) Complete (a) to (j) in the following C program to produce the permutations of 1,2,3 in the following order: 231, 321, 312, 132, 213, 123.

```
#include <stdio.h>
#include <string.h>
/*swap the values pointed by x and y*/
void swap(char *x, char *y){
    char tmp;
    (a) (1%);
    (b) (1%);
    (c) (1%);
}
/*generate permutatioms recursively*/
void permutation(char *start, char *end){
    char *cp;
    if( (d) (1%) )
        ptrinf( (e) (1%) );
    else
        for( (f) (1%) ; (g) (1%) ; cp++ ){
            swap( (h) (1%) );
            permutation( (i) (1%) );
            swap( (j) (1%) );
        }
}
int main(void){
    char str[] = {'1', '2', '3', '\0'};
    permutation(str, str+strlen(str)-1);
    return 0;
}
```

——— END OF PAPER ———