Computer Science Foundation Exam

December , 2005

Computer Science

Section 1A

No Calculators!

KEY

Solutions and Grading Criteria

Name:____

SSN:____

Score: /50

In this section of the exam, there are four (4) problems. You must do all of them.

Partial credit cannot be given unless all work is shown and is readable.

Be complete, yet concise, and above all be neat.

1. [**12 pts**] Transform the following infix expression into its equivalent postfix expression using a stack. Show the contents of the stack at the indicated points 1, 2 and 3 in the infix expressions.

1 2 3

P / (B + S – T) + M * N / C – A + F

+		
(	*	/
/	+	+
1	2	3

Resulting Postfix Expression :

P	B	S	+	T	-	/	M	N	*	C	/	+	A	-	F	+

3 points each correct stack
Points taken off for wrong entries
3 points for final expression

2. [11 x 2 pts] Indicate the time complexity for each of the following operations in terms of Big-O notation, assuming that efficient implementations are used. Give the *worst case* complexities. Following notations are being used:

AINC is an array containing n integers arranged in increasing order.
AD is an array containing n integers arranged in decreasing order.
AR is an array containing n integers in random order.
Q is a queue implemented as a linked list and containing **p** elements.
LINK is a linked list containing n nodes.
CIRC is a circular linked list containing n elements, where C points to the last element.
T is a binary search tree containing **n** nodes.

2 points each correct answer
0 otherwise

a) Searching for an element in AINC using linear search. ___ O(n) _____

b) Deleting the 10th node of linked list LINK. ___O(1) _____

c) Calling a function which uses Q, and calls *dequeue* **m** times. ___ O(m) ______

d) Inserting an element at the end of the list CIRC. ___ O(1) _____

e) Deleting the last element of CIRC. ___ O(n) _____

f) Finding the largest element of T. ____O(n) ____

g) Doubling the value stored in root node of T. ___ O(1) _____

h) Making the call selectionsort (AINC, n). ___ $O(n^2)$ _____

i) Making the call bubblesort(AINC, n). ____ O(n) ____

j) Making two calls one after another. The first call is mergesort(AD,n), followed by the call quicksort(AD,n). ____ $O(n^2)$ ____

k) Converting a decimal integer *num* into its binary equivalent. ___ O(log num) _______

3. **[8 pts**] Write a function which accepts a linear linked list J and converts it into a circular linked list. The function should return a pointer to the last element.
The node structure is as follows:

```
struct node{
     int data;
     struct node *next; };
```

Partial credits may be awarded if the function is not correct

```
struct node * convert ( struct node * J)
{
    struct node * temp = J;
    while ( temp -> next != NULL)
        temp = temp->next;
    temp->next = J;
    return temp;
}
```

4. **[8 pts]** Write the recurrence relation for the following function which takes as input the first n elements of the array ARRAYS holding integers. Solve it to work out the total number of operations, and the time complexity of the algorithm.

```
int modify( int ARRAYS[ ], int n){
     int maximum;
     if (n==1) return ARRAYS[0];
     else
     {
          maximum = findmax(ARRAYS, n);
          ARRRAYS[n-1] = maximum;
          return modify(ARRAYS, n-1);
     }
}
```

The function *findmax* returns the largest value in the array ARRAYS of size n.

```
int findmax(int A[ ], n) {
     max = A[0];
     for (i=1, i<=n, i++)
          if (A[i] > max) max=A[i];
     return max;
}
```

The findmax function has a time complexity of O(n)
So it can simply be taken as n

2

Recurrence relation:
T(n) = T(n-1) + n
T(1) = 1

2
points

T(n-1) = T(n-2) + n-1
Substituting back T(n) = T(n-2) + n + n-1
Again T(n-2) = T(n-3) + n-2
Substituting back T(n) = T(n-3) + n + n-1 + n-2

In general
T(n) = T(n-k) + n + n-1+. . . . +(n-k+1)

Since T(1)=1, Let n-k = 1
This gives k = n-1
Thus T(n) = T(1) + n + n-1 + . . . + 3 + 2

= n + n-1 + . . . + 3 + 2 + 1
= n(n +1)/2

3
points

1 point

`Time complexity = O(n`^{2}` )`