National University of Computer and Emerging Sciences, Lahore Campus

Course:	Applied Programming	Course Code:	CS-319
Program:	MS (Computer Science)	Semester:	Fall 2017
Duration:	60 Minutes	Total Marks:	50
Paper Date:	20-Sep-17	Weight	15 %
Section:	A and B	Page(s):	06
Exam:	Midterm 1	Reg. No.	

Instruction/Notes:

- Attempt all questions on this booklet
- If anything is unclear, make a reasonable assumption and mention it with the answer

For instructor's use only:

Question #	Maximum marks	Obtained marks
1	15	
2	05	
3	10	
4	20	
Total	**50**	

Q. 1. Fill in the following table with the complexity in big-oh notation for the worst case: [15 marks]

	Insert an element	Delete an element	Find an element	Find an element and move it to the beginning	Sort the elements in ascending order
Array	O(n)	O(n)	O(n)	O(n)	$O(n^2)$
Singly Linked List	O(n)	O(n)	O(n)	O(n)	$O(n^2)$
Doubly Linked List	O(n)	O(n)	O(n)	O(n)	$O(n^2)$

Q. 2. The worst case running time of algorithm A and B is O(n lg n) and $O(n^2)$, respectively. Is it possible that sometimes algorithm B runs faster than algorithm A? Provide reasoning. [5 marks]

Yes. The given values are for the worst cases. The worst case of the two algorithms may not be the same and for some input, algorithm B might run faster than algorithm A. Also, these are asymptotic bounds that hold for values of n greater than or equal to some n_0. For n less than n_0, the running time of algorithm A may be greater than that of algorithm B.

Q. 3. Prove the following using mathematical induction. [10 marks]

a) $\sum_{i=j}^{n} i = \frac{(n+1-j)(n+j)}{2}$

b) $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

a) $\sum_{i=j}^{n} i = \frac{(n+1-j)(n+j)}{2}$

Base case: $n = j$

L.H.S. $\sum_{i=j}^{j} i = j$ R.H.S. $\frac{(j+1-j)(j+j)}{2} = j$

Since L.H.S. = R.H.S, the hypothesis holds for $n=j$

Assume that

$\sum_{i=j}^{n} i = \frac{(n+1-j)(n+j)}{2}$ for some n

Induction step:

L.H.S. $\sum_{i=j}^{n} i + (n+1) = \frac{(n+1-j)(n+j)}{2} + n+1$

$= \frac{(n+1-j)(n+j)+2(n+1)}{2}$

$= \frac{n^2+3n+j-j^2+2}{2}$

R.H.S. Substitute n by $n+1$

$\frac{(n+1+1-j)(n+1+j)}{2} = \frac{(n+2-j)(n+1+j)}{2}$

$= \frac{n^2+n+\cancel{nj}+2n+2+2j-\cancel{nj}-j-j^2}{2} = \frac{n^2+3n+j-j^2+2}{2}$

Thus, proved that the hypotheses holds for all n.

b) Base case: $n = 1$

L.H.S. $\sum_{i=1}^{1} i^2 = 1$

R.H.S. $\frac{1(1+1)(2+1)}{6} = \frac{1 \times 2 \times 3}{6} = 1$

L.H.S = R.H.S.

Assume that the hypothesis holds for some n

Induction step:

L.H.S. $\sum_{i=1}^{n} i^2 + (n+1)^2 = \frac{n(n+1)(2n+1)}{6} + (n+1)^2$

$$= \frac{n(n+1)(2n+1) + 6(n+1)^2}{6} = \frac{(n+1)(n(2n+1) + 6(n+1))}{6}$$

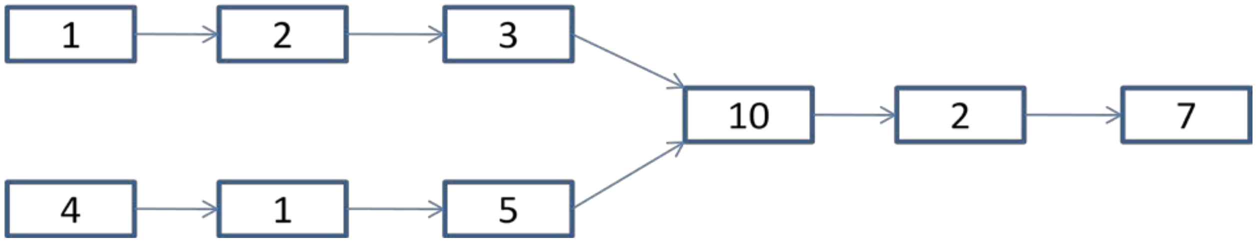
$$= \frac{(n+1)(2n^2 + n + 6n + 6)}{6} = \frac{(n+1)(2n^2 + 7n + 6)}{6}$$

$$= \frac{(n+1)(2n^2 + 4n + 3n + 6)}{6} = \frac{(n+1)(n+2)(2n+3)}{6}$$

which is the same if we replace n by $n+1$ in $\frac{n(n+1)(2n+1)}{6}$

Thus, proved.

Q. 4. You are given two linked lists with head pointers head1 and head2. The two linked lists merge at some point as shown in the following example. You need to write code that finds the merge point.



Complete the following starter code, adding functions as necessary so that the above requirements are met.

[20 marks]

```
struct Node{

        int key;

        struct Node* next;

};

struct Node *head1, *head2;

struct Node* FindMergePoint()
{
        struct Node* temp1 = head1, *temp2;
        while(temp1 != NULL)
        {
                temp2 = head2;
                while(temp2 != NULL)
                {
                        If (temp1 == temp2)
                                return temp1;
                        temp2 = temp2 -> next;
                }
                temp1 = temp1 -> next;
        }
        return NULL;
}
```