United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm   Year: 2018   Trimester: Spring
Course: CSI 217 Data Structures, Marks: 30, Time: 2 hours

Answer all the following questions.

1. a) Define data structure.   [1]
    b) Show the working mechanism of *Bubble Sort* by sorting the following numbers in *descending* order:
          14, 10, 2, 5, 3, 19, 12, 7   [2]
    c) Design the algorithm of '**partition'** function of *Quick Sort* algorithm for sorting numbers in *ascending* order (assuming the last element as pivot).   [2.5]
    d) Design the algorithm of *Selection Sort* for sorting numbers in *descending* order.   [2]

2. a) Apply '**Merge Sort'** on following dataset in order to sort them in descending order:
    14, 1, 12, 6, 10, 5, 7, 20   [3]
    b) Which searching algorithm do we use while searching for a word in a dictionary? Explain with an example.   [2.5]
    c) When will you prefer Linear search over Binary search despite the fact that Binary search is faster than Linear search?   [2]

3. a) Write a function that removes every alternate element of a singly linked list from the beginning. The before and after states of the linked list will be as follows.   [4]

Before	After
1->2->3->4->5->6->7->8->NULL	1->3->5->7->NULL
1->2->3->NULL	1->3->NULL
1->NULL	1->NULL

Assume that, the node is a structure that looks like as follows. Also assume that, there is data already inserted in the linked list and head is the pointer to the start of the linked list. You just have to write a function called **removeAlternateElements()** to do the task.

```
struct node{
    int data;
    struct node *next;
} ;
struct node* head;
void removeAlternateElements(){


}
```

b) A singly linked list has all nodes with values less than 10. Write a function that does the following: if the sum of two consecutive nodes is more than 10, then it inserts a node with data -10 between them. For example:

Before	After
1-> 2-> 3-> 8-> 9-> NULL	1-> 2-> 3-> -10-> 8-> -10-> 9->NULL
5-> 5-> 7-> 8-> 2-> 13->NULL	5-> 5-> 7-> -10-> 8-> 2-> -10->13->NULL
1-> 2-> 3->NULL	1-> 2-> 3-> NULL

Assume that, the node is a structure that looks like as follows. Also assume that, there is data already inserted in the linked list and head is the pointer to the start of the linked list. You just have to write a function called **insertMinusTen()** to do the task.   [3.5]

```
struct node{
    int data;
    struct node *next;
} ;
struct node* head;
void insertMinusTen(){
}
```

4. a) Suppose that you are playing a game called "guess the number" with your friend and you have to guess a number within a range of 1 to 100(inclusive). To have a hint, you can follow two methods:
   i.   Ask your friend "is the number 1"? And if s/he says no, then ask "is the number 2", and go on.
   ii. Ask your friend whether the number is greater or smaller than 50? If s/he says greater than 50. Then you ask whether the number is greater or smaller than 75? And go on…

   Which of the above method is more efficient if your friend guessed the number 100? With how many questions you can answer that s/he guessed 100 in both (i) and (ii)?

Write a pseudo-code for guessing the number with the most efficient way.
[4]

b) Draw a diagram for each of the statements given below. You need to draw a picture
for each of the instructions that has a line number at their left.   [3.5]

```
    typedef struct node{
       int data;
       struct node *next;
    };
    node* createNode(int data){
          node* x =(node*) malloc(sizeof(node));
          x->data =data;
          x->next =NULL;
          return x;
    }
    1.  node* a =createNode(1);
    2.  node* b =createNode(2);
    3.  node* c= createNode(3);
    4.  c->next=b;
    5.  b->next=a;
    6.  d= createNode(4);
    7.  node*temp=b->next;
    8.  b->next=d;
    9.  d->next=temp;
    10. node* head= c->next;
    11. c->next=NULL;
    12. free(c);
    13. node*temp=head;
    14. while(temp->next!=NULL){ //draw only a single picture for this while loop
    15.    temp->data++;
    16.    temp=temp->next;
    17. }
    18. temp->next=head;
    19. //at this line, draw the current state of the linked list with all the nodes and
          links between them
```