EECE.2160: ECE Application Programming

Spring 2017

Lecture 31-33: Key Questions
April 12-April 19, 2017

Note: This handout will be used for the next three lectures—if you get the handout during Lec. 31, please bring it to Lec. 32 and 33!

1. Explain the use of general data structures and pointer-based data structures in particular.

2. Describe the general design of a linked list.

3. Describe the structure used for each node in the list.

4. Explain the operation of the following function, which adds a node to the beginning of the list and returns a pointer to that node.

```
LLnode *addNode(LLnode *list, int v) {
  LLnode *newNode;
  // Allocate space for new node; exit if error
  newNode = (LLnode *)malloc(sizeof(LLnode));
  if (newNode == NULL) {
    fprintf(stderr,
              "Error: could not allocate new node\n");
    exit(0);
  }
  newNode->value = v;    // Copy value to new node
  newNode->next = list;  // next points to old list
  return newNode;
}
```

5. Write each of the following functions:

a. Finding item in list (Function should return pointer to node if found and return NULL otherwise)

```
LLnode *findNode(LLnode *list, int v) {









}
```

b. Write the following function used to remove a node from list:
   - Must deallocate space for deleted node
   - Function should return pointer to start of list after it has been modified
     - No modifications should be made if value v is not in list
     - Hint: you can use the `findNode()` function in this function, but you may not want to!
   - Note: removing first element in list is special case

```
LLnode *delNode(LLnode *list, int v) {













}
```

6. Describe how to maintain a sorted linked list.

7. Write each of the following functions:

c. Adding an item to a sorted linked list
   - Use **addNode()** as a starting point
   - Instead of adding node at beginning, find appropriate place in list and then add
   - Function should return pointer to start of list after it has been modified

```
LLnode *addSortedNode(LLnode *list, int v) {

}
```

d. Finding an item in a sorted linked list
   - Use **findNode()** as starting point—should perform same operation, but more efficiently
   - Function should return pointer to node if found
   - Return NULL otherwise

```
LLnode *findSortedNode(LLnode *list, int v) {
















}
```