# 16.216: ECE Application Programming

Fall 2015

Lecture 35: Key Questions
December 4, 2015

1. Review the following:

a. General design of a linked list built using the following structure:

```
typedef struct node {
   int value;              // Data
   struct node *next;      // Pointer to next node
} LLnode;
```

b. The function below, 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;
}
```

c. The function below, which attempts to find a node containing value v, returning the address of that node if it exists and NULL otherwise:

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

   n = list;          // Start with first node

   while (n != NULL) {          // Search until after last node
      if (n->value == v)        // Data found--return n
         return n;
      n = n->next;
   }
   return NULL;       // If you get here, data wasn't found
}
```

2. 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) {

}
```

3. Describe how to maintain a sorted linked list.

4. Write each of the following functions:

a. 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) {

}
```

b. 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) {




}
```

5. Describe the key components of Program 10.