

# CSI 33 LECTURE NOTES (Ojakian)

## Topic 4: Linked Lists

---

### OUTLINE

(References: Ch 4, 11)

1. Linked Lists
  2. Efficiency issues
- 

#### 1. Linked List Introduction

- (a) Empirical Timing of Programs on Insertion

**PROBLEM 1.** *Experiment with the CPP program: array vs vector vs list. Why the difference in run times, especially for the list class?*

- (b) Concept of Linked List

- i. Define the data structure

**PROBLEM 2.** *Show a diagram for the linked list with the following data:*

3, 4, 3, 1

- ii. Search for an element: by data or index.

**PROBLEM 3.**

A. *Show the steps for finding the 1.*

B. *Show the steps for finding the item at position 2.*

- iii. Insert an element (compare to array insertion)

**PROBLEM 4.**

A. *Use a diagram to show the steps for inserting a 2 after the 4.*

B. *Use a diagram to show the steps for inserting a 5 at the beginning of the linked list, and at the end of the linked list.*

- iv. Delete an element (compare to array deletion)

**PROBLEM 5.**

A. *Use a diagram to show the steps for deleting the 4.*

B. *Use a diagram to show the steps for deleting the first and last elements.*

#### 2. Programming a Linked List in C++

**PROBLEM 6.** *In C++, write just a List Node class and use it to build a class and delete some elements, following example program.*

**PROBLEM 7.** *Write the code for a Linked List Class. Then repeat the above operations from the prior problems. Put in the following methods: insert (at position), find (at position), remove (an element).*

### 3. Theta Analysis: Linked List versus Array

**PROBLEM 8.** *Do the theta analysis of **searching** in arrays versus linked lists: by data value and by index.*

**PROBLEM 9.**

- (a) *Do the theta analysis of **insertion** in arrays versus linked lists: consider where the insertion happens.*
- (b) *Do the theta analysis of **deletion** in arrays versus linked lists: consider where the deletion happens.*

### 4. Other Linked Structures

There are many other kinds of “Linked Structures”

- (a) Doubly Linked Lists. For example C++ list.

**PROBLEM 10.** *Make a diagram of the list 3, 4, 3, 1 as a doubly linked list. Insertions and deletions?*

- (b) Circularly Linked Lists: Cat has its tail ...
- (c) Trees can be: We'll see later ...