Reading 3

Weiss 3.2

Simple Linked Lists

- not stored contiguously
- consists of a series of nodes that are not necessarily adjacent in memory
- each node containes an element and a link to a node containing the next node
- the last cell’s link points to null
- **printlist()** or **find(x)** would be linear-time, traversing through next links
- **findkth()** operation is not as efficient as an array
- **remove** method can be executed in one **next** pointer change
- **insert** method requires obtaining a new node from the system by usign a new cell and executing two next pointer maneuvers
- inserting and deleting in a linked list does not require moving a lot of items; only a constant number of changes to node links
- adding to the front or removing the first item is constant time, as well as adding at the end of the list
- doubly linked lists also exists

Shaffer 4.1,4.2