CS 162, Lecture n-2: Data Structures

4 June 2018
Data Structures

• the conceptual shape or arrangement of the data

• Arrays

  Stored in contiguous memory
  Random access to data elements
  Same data

• Lists

  Not stored contiguously
  No random access
  Any data of the same type
Abstract Data Types

• Stack: entries are only inserted and removed at the head
  • Last in, First out (LIFO)
  • Push: add to the top/front
  • Pop: remove from the top/front
  • Ideal for storing items that must be retrieved in the reverse order from which they are stored
• Examples:
  - Stack of plates
  - Recursion
  - Parking in Cowell's War (Card game)
  - Food containers

order of undos
order of tasks

```python
def sum(list):
    if len == 0:
        return val
    return sum(list-1) + val
```
Abstract Data Type

• Queue: entries only removed at the front, entries only added to the back
  • FIFO
  • Push: add to the back
  • Pop: remove from the front
  • Examples

  Printing queue
  Nerf gun
  Line of people
  Customer service call
  Waiting at a stop light

  Updates
  Server
  Buffer
Trees

• A collection whose entries have a hierarchical organization
• Each position is called a node
• Node at the top is the root node
• Nodes at the end are called terminal nodes (leaves)
• Binary tree: a tree in which each parent has no more then two children
• Examples

  Ancestry
  Employee hierarchy

Solving puzzles (chores)
How do we store a binary tree? What functions do we need to work with a tree?

```
struct node {
    data;
    node *left;
    node *right;
}
```

![Binary tree diagram]

- Insert
- Delete
- Balance
- Clear
- Remove

Example tree:
```
root

/     |
2a    2b   3a   3b   3c   3d
```

Find parent:
```
int node index = 2
```