Lecture 14

More on Recursion

Ex.

- There are n items.
- Your bag can hold total weight of W.
- If item i is worth v_i, and weighs w_i, then how would you decide which items to select?

Maximize total value of items you select.

Cannot select fractions of items.

Solution: Given an array of values and weights
For each item, can either pick it or not.
Creates sub problems – if we pick the first item subtract the weight of it from the capacity of the bag and move to next item.
Base condition: reach the end of the list.

More efficient iterative solution:

```
public class Knapsack {
    private List<Integer> weights;
    private List<Integer> values;
    private int w;
    public int recursiveCalls = 0;

    ...

    public int solve_recursive() {

    }

    public int solve_iterative() {
        // add later
    }
```

Recursive Datatypes

Given an unbalanced bst (ie. all values are to the right of each other), how would you balance it?
Maybe PPT question

Solution: Root should be middle value of sorted elements. (Maybe represent as linked list). Remove the middle element and then just add the new middle element to the tree.

Ex. BigInteger

- Large values cannot be represented with regular integers
- If you reference an integer as int = zero + prev, you can represent very large numbers.

Ex. Boolean Statements

- Each statement is true or false
- Certain operations to be defined: S = S1 or S2, S = S1 and S2, etc.