• **DISCLAIMER:** These notes are not necessarily an accurate representation of what I said during the class. They are mostly what I intend to say, and have not been carefully edited.

• Main topics:
  – Dynamic programming for knapsack. State is weight of things kept so far, transition depends on whether repetitions are allowed.

• Textbook:
  – Chapter 6.4.

• Last time
  – Memoization view of Dynamic Programming
  – All-pairs shortest path

**Problem:** Given set of items \{1, ..., n\}, each with weight and value \((w_i, v_i)\) such that \(w_i\) is an integer. For a given \(W\) choose a set of items with total weight \(\leq W\) such that the total value is maximized.

**Can choose same item multiple times**

1. Let \(K[w]\) be the optimal solution of knapsack with \(W = w\).

2. Base case: \(K[0] = 0\).

3. States: \(K[w]\) with \(w \leq W\)

4. Transition: \(K[w] = \max_i \{K[w-w_i] + v_i : w-w_i \geq 0\}\)

   **Knapsack with repetition**
   1. \(K[0] = 0\)
   2. For each \(w = 1\) to \(W\)
      
      (a) \(K[w] = \max_i \{K[w-w_i] + v_i : w-w_i \geq 0\}\).
   3. Return \(K[W]\).
1. Each transition takes $O(n)$ time
2. The for loop iterates $O(W)$ times
3. The full runtime is then $O(nW)$ time.

**Can only choose each item once**

1. Let $K[w, \{1, \ldots, i\}]$ be the optimal solution of knapsack with $W = w$, and the set of items $\{1, \ldots, i\}$
2. Base cases: $K[w, \{1\}] = v_1$ if $w - w_1 \geq 0$, and 0 otherwise
3. States: $K[w, \{1, \ldots, i\}]$ with $w \leq W$ and $i \leq n$
4. Transition: $K[w, \{1, \ldots, i\}] = \max\{K[w - w_i, \{1, \ldots, i - 1\}] + v_i, K[w, \{1, \ldots, i - 1\}]\}$

**Knapsack without repetition**

1. $K[w, \{1\}] = v_1$ if $w - w_1 \geq 0$, 0 otherwise
2. For $i = 1$ to $n$
   (a) For each $w = 1$ to $W$
      i. $K[w, \{1, \ldots, i\}] = \max\{K[w - w_i, \{1, \ldots, i - 1\}] + v_i, K[w, \{1, \ldots, i - 1\}]\}$
3. Return $K[W, \{1, \ldots, n\}]$.

1. Each transition takes $O(1)$ time
2. The first for loop iterates $O(n)$ times
3. The second for loop iterates $O(W)$ times
4. The full runtime is then $O(nW)$ time.