CS 170 2.25 LECTURE 11

DP: solve large problems by breaking it down into smaller subproblems.

SHORTEST PATHS IN DAG

Find shortest path from $s \in V$ to $v \in V$.
Subproblems: shortest path to v' for v' closer to s than v
Recombining: $dist(v) = \min(dist(u) + w(u,v))$ for $u : (u,v) \in E$
Base Case: $dist(s) = 0$, or $dist(w) = \infty$ if w is a source
Order to Solve: topologically sort vertices starting at s

What if we wanted to find the longest path? Just change min to max!

LONGEST INCREASING SUBSEQUENCE

Subproblems: $f(i) = LIS$ in $x_1, \ldots, x_i$ X $L(i) =$ length of LIS in $x_1, \ldots, x_i$
Recombining: $L(i) = \max(1, \max\limits_{\substack{j<i \\ x_i < x_j}} (L(j)+1))$

Base Case: $L(1) = 1$
Order to Solve: increasing i Cost $= \sum O(i) = O(n^2)$

For $i<j$, there is an edge $x_i \to x_j$. This is longest path in DAG!

EDIT DISTANCE

How many additions, deletions, or substitutions needed to change x to y?
Subproblems: for $1 \le i \le len(x)$, $1 \le j \le len(y)$: $f(i,j) = ED(x[:i], y[:j])$
Recombination: look at last char $\to$ $(x_i, \_)$ $(\_, y_j)$ (x_i, y_j)
Cost: $f(i-1,j)+1$ $f(i,j-1)+1$ $f(i-1,j-1) + int(x_i \neq x_j)$
$f(i,j) = \min(\text{the above } 3)$
Base Case: $f(i,0) = i$ (delete) or $f(0,j) = j$ (insert)
Order to Solve: for (i,j) need the above 3 cases; think of a matrix

KNAPSACK PROBLEM

Your knapsack can carry w lbs, and you can choose among n jewels with values $v_1, v_2, \ldots, v_n$ and weights $w_1, w_2, \ldots, w_n$. How can you maximize value within the weight limit?
Subproblems: $f(i,u) =$ max value packing subset of $1, \ldots, i$; $\sum weight \le u$
$= \text{if } (w_i > u): f(i-1, u)$
$\text{else}: \max(f(i-1,u), v_i + f(i-1, u-w_i))$
Base Case: $f(0,u) = 0$ and $f(i,0) = 0$
Order to Solve: for $i=1$ to $i=n$: for $u=1$ to $u=w$: $f(i,u) =$ subproblem