Dijkstra's algorithm

CS 312

Greedy Algorithms

function Greedy (C: set) : set
S := emptySet
while nonEmpty (C) and not solution (S) do
    x := select (C)
    C := C \ {x}
if feasible (S union {x}) then S := S union {x}
if solution (S) return S else noSolution

Knapsack problem I

Given n objects.
Each object has a value, v_i, and a weight, w_i.
Knapsack can only carry weight W.
Maximize the value of the stuff in the knapsack.
CAN split objects into pieces and carry x_i of object i.

maximize \( \sum_{i=1}^{n} x_i v_i \)
but keep \( \sum_{i=1}^{n} x_i w_i \leq W \)

Knapsack

function knapsack (v[1...n], w[1...n], W) : array [1...n]
for i = 1 to n do x[i] := 0
weight := 0
while weight < W do
    i := the best remaining object
    if weight + w[i] <= W then
        x[i] := weight + w[i]
    else x[i] := (W - weight) / w[i]
    weight := W
return x