THEOREM

IF WE MAXIMIZE v_i/w_i ON LINE (6) THEN KNAPSACK RETURNS AN OPTIMAL SOLUTION.

PROOF:

WITHOUT LOSS OF GENERALITY WE MAY ASSUME THE OBJECTS ARE INDEXED BY DECREASING v_i/w_i:

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \cdots \geq \frac{v_n}{w_n}$$

LET $x = (x_1, \ldots, x_n)$ BE THE SOLUTION RETURNED BY KNAPSACK.

IF ALL $x_i = 1$, THEN x IS CLEARLY OPTIMAL OTHERWISE LET j BE THE FIRST INDEX SUCH THAT $x_j < 1$. INSPECTING THE ALGORITHM ITS CLEAR THAT

$$x_i = 1 \quad \text{FOR } 1 \leq i < j$$
$$x_j < 1$$
$$x_i = 0 \quad \text{FOR } j < i \leq n,$$

AND THAT

$$\sum_{i=1}^{n} x_i w_i = W$$

LET $V(x) = \sum_{i=1}^{n} x_i v_i$, THE VALUE OF x.
LET $y = (y_1, \ldots, y_n)$ BE ANY OTHER FEASIBLE SOLUTION, AND $V(y) = \sum_{i=1}^{n} y_i v_i$ ITS VALUE.

WE MUST SHOW $V(x) \geq V(y)$, AND HENCE x IS OPTIMAL.

SINCE y IS FEASIBLE $\sum_{i=1}^{n} y_i w_i \leq W$, AND THEREFORE

$$\sum_{i=1}^{n} (x_i - y_i) w_i = W - \sum_{i=1}^{n} y_i w_i \geq 0.$$

NOW

$$V(x) - V(y) = \sum_{i=1}^{n} (x_i - y_i) v_i$$

$$= \sum_{i=1}^{n} (x_i - y_i) w_i \left(\frac{v_i}{w_i}\right).$$

OBSERVE THAT

$$i < j \Rightarrow x_i = 1 \Rightarrow x_i - y_i \geq 0 \text{ AND } \frac{v_i}{w_i} \geq \frac{v_j}{w_j}$$

$$\therefore (x_i - y_i)\left(\frac{v_i}{w_i}\right) \geq (x_i - y_i)\left(\frac{v_j}{w_j}\right)$$

$$i > j \Rightarrow x_i = 0 \Rightarrow x_i - y_i \leq 0 \text{ AND } \frac{v_i}{w_i} \leq \frac{v_j}{w_j}$$

$$\therefore (x_i - y_i)\left(\frac{v_i}{w_i}\right) \geq (x_i - y_i)\left(\frac{v_j}{w_j}\right)$$

$$i = j \Rightarrow (x_i - y_i)\left(\frac{v_i}{w_i}\right) = (x_i - y_i)\left(\frac{v_j}{w_j}\right)$$

THUS $(x_i - y_i)\left(\frac{v_i}{w_i}\right) \geq (x_i - y_i)\left(\frac{v_j}{w_j}\right)$ FOR $1 \leq i \leq n$.

$$\therefore V(x) - V(y) \geq \sum_{i=1}^{n} (x_i - y_i) w_i \left(\frac{v_j}{w_j}\right)$$

$$= \left(\frac{v_j}{w_j}\right) \sum_{i=1}^{n} (x_i - y_i) w_i$$

$$\geq 0 .$$

$\therefore X = (x_1, \ldots, x_n)$ is OPTIMAL, AS REQUIRED. ///.

EXAMINING LOOPS 2-3 AND 5-13, ONE SEES THAT KNAPSACK RUNS IN TIME $\Theta(n)$.

OUR DYNAMIC PROGRAMMING SOLUTION TO THE 0-1 KNAPSACK PROBLEM RAN IN TIME $\Theta(nW)$, SINCE THE TABLE WAS OF SIZE $n \times (W+1)$

PERHAPS THE 0-1 KNAPSACK PROBLEM CAN BE SOLVED MORE EFFICIENTLY USING A GREEDY STRATEGY.

Ex. SAME AS BEFORE, BUT NOW 0-1 KNAPSACK.

$n = 5$, $W = 10$

V	2	3	6.6	4	6
W	1	2	3	4	5
V/W	2	1.5	2.2	1	1.2

GREEDY SOLUTION:

$X = (1 \quad 1 \quad 1 \quad 1 \quad 0)$ $\begin{cases} \text{VALUE} = 15.6 \\ \text{WEIGHT} = 10 \end{cases}$

EXERCISE

CHECK THAT DYNAMIC PROGRAM YIELDS THE VERY SAME SOLUTION.

Ex. $n = 5$, $W = 11$

V	1	6	18	22	28
W	1	2	5	6	7
V/W	1	3	3.6	3.67	4

GREEDY SOLUTION!

$X = (1 \quad 1 \quad 0 \quad 0 \quad 1)$ $\begin{cases} \text{VALUE} = 35 \\ \text{WEIGHT} = 10 \end{cases}$

EXERCISE

CHECK THAT DYNAMIC PROGRAM YIELDS THE SOLUTION

$Y = (0 \quad 0 \quad 1 \quad 1 \quad 0)$ $\begin{cases} \text{VALUE} = 40 \\ \text{WEIGHT} = 11 \end{cases}$

THUS THE GREEDY STRATEGY DOES NOT ALWAYS YIELD THE OPTIMAL SOLUTION TO 0-1 KNAPSACK.

COIN CHANGING PROBLEM

AS BEFORE DENOMINATIONS $d = (d_1, \ldots, d_n)$ AND AN AMOUNT N TO BE DISBURSED WITH THE FEWEST COINS. (ASSUME AN UNLIMITED SUPPLY OF COINS IN EACH DENOMINATION.)

THE GREEDY STRATEGY TO COIN CHANGING IS AS FOLLOWS:

- FROM AMONGST ALL THE DENOMINATIONS WHOSE ADDITION WOULD NOT CAUSE THE SUM TO EXCEED N, CHOOSE THE LARGEST.
- STOP WHEN SUM IS N.

EXCERCISE (HARD)

SHOW THAT FOR $d = (1, 5, 10, 25, 100)$ THE GREEDY STRATEGY YIELDS AN OPTIMAL SOLUTION FOR ANY $N \geq 0$.

EXCERCISE (EASY)

SHOW THAT FOR $d = (1, 10, 25, 100)$ THE GREEDY STRATEGY DOES NOT YIELD AN OPTIMAL SOLUTION FOR SOME N. (e.g. $N = 30$)

EXCERCISE (HARD)

CHARACTERIZE ALL DENOMINATION SETS $d=(d_1,..,d_n)$ SUCH THAT THE GREEDY STRATEGY YIELDS AN OPTIMAL SOLUTION FOR ALL $N \geq 0$.

HOW CAN WE TELL IF A GREEDY ALGORITHM WILL SOLVE A PARTICULAR OPTIMIZATION PROBLEM? IN GENERAL THIS IS A DIFFICULT QUESTION. THE KEY INGREDIENTS TO LOOK FOR ARE

- OPTIMAL SUBSTRUCTURE : OPTIMAL SOLUTIONS CONTAIN OPTIMAL SUBPROBLEM SOLUTIONS.

- GREEDY CHOICE PROPERTY : A GLOBALLY OPTIMAL SOLUTION CAN BE OBTAINED BY MAKING LOCALLY OPTIMAL (GREEDY) CHOICES (WITH RESPECT TO SOME SELECTION FUNCTION.)

MUCH OF THE HARD WORK IN DESIGNING A GREEDY ALGORITHM IS IN PROVING THAT THE GREEDY CHOICE PROPERTY IS SATISFIED.

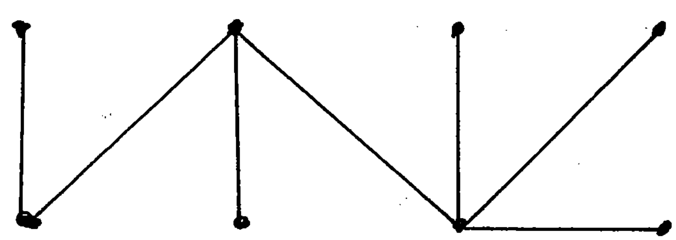
THE SITUATION IS COMPLICATED BY THE FACT THAT THE SELECTION FUNCTION MAY NOT COINCIDE WITH THE OBJECTIVE FUNCTION.

MINIMUM WEIGHT SPANNING TREES

WE RECALL SEVERAL DEFINITIONS RELATED TO AN (UNDIRECTED) GRAPH $G=(V,E)$.

- G IS CALLED CONNECTED IF G CONTAINS A u-v PATH FOR ALL $u,v \in V$.
- A CYCLE IN G IS A CLOSED PATH IN G, i.e. A PATH WHOSE INITIAL AND TERMINAL VERTICES ARE IDENTICAL
- G IS CALLED ACYCLIC IF IT CONTAINS NO CYCLES
- A GRAPH WHICH IS CONNECTED AND ACYCLIC IS CALLED A TREE.

Ex.



$|V|=8$, $|E|=7$

THEOREM

THE FOLLOWING ARE EQUIVALENT.

(1) G is A TREE
(2) G is CONNECTED AND $|E| = |V| - 1$
(3) G is ACYCLIC AND $|E| = |V| - 1$
(4) G is ACYCLIC, BUT IF A NEW EDGE is ADDED, A UNIQUE CYCLE is CREATED.
(5) G is CONNECTED, BUT THE REMOVAL OF ANY EDGE DISCONNECTS G.
(6) THERE is A UNIQUE u-v PATH IN G FOR ALL $u, v \in V$.

FOR A PROOF SEE APPENDIX B P. 1085, OR TAKE CE 177.

- A SUBGRAPH OF G is CALLED A <u>SPANNING TREE</u> IF IT IS A TREE, AND IT INCLUDES ALL VERTICES OF G

EX.