⋮

Proof

Assume w.l.o.g. that

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \frac{v_3}{w_3} \geq \cdots \geq \frac{v_n}{w_n}$$

let $x = (x_1, x_2, \ldots, x_n)$ be the solution returned by Knapsack.

If all $x_i = 1$, then x is certainly optimal. Assume this not the case.

let j be the 1^{st} index for which $x_j < 1$, so

$$x_i = 1 \qquad \text{for } 1 \le i < j$$

$$x_j < 1$$

$$x_i = 0 \qquad \text{for } j < i \le n$$

Also

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

let $V(x) = \sum_{i=1}^{n} x_i v_i$, the value of x. let $y = (y_1, y_2, \dots, y_n)$ be any other feasible solution and $V(y) = \sum_{i=1}^{n} y_i v_i$, its value.

must show: $V(x) \geq V(y)$.

since y is feasible:

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

so

$$\sum_{i=1}^{n} (x_i - \overset{y_i}{\cancel{x_i}}) w_i = \sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i$$

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

$$\geq 0.$$

observe

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

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

and

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

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

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

$$j < i \Rightarrow x_i = 0 \Rightarrow x_i - y_i \leq 0 \quad \& \quad \frac{v_i}{w_i} \leq \frac{v_j}{w_j}$$

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

In all cases $(1 \le i \le n)$

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

so

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

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

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

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

$$\ge 0$$

$\therefore \quad V(x) \geq V(y).$

since y was an arbitrary feasible soln, x is optimal. $\blacksquare$

<u>Runtime</u>:

- sort items: $\Theta(n \log n)$
- run loop: $\Theta(n)$

total cost $\Theta(n \log n)$

Recall: Dyn. Prog. Cost $= n(W+1)$

$$= \Theta(nW)$$

Exercise

adapt greedy algorithm to the discrete Knapsack. (break when can't take all of an object.)

show by examples that this algorithm sometimes produces an opt. solution, sometimes does not.

Coin Changing again

Denominations: $d = (d_1, d_2, \ldots, d_n)$

Amount : N units

Greedy strategy:
from among all denominations whose addition would not cause sum to exceed N, choose largest. Stop when sum is N.

Exercise (hard, doable)

show for $d = (1, 5, 10, 25, 100)$ that greedy strategy yields an opt. soln for all $N \geq 0$.

Exercise (easy)

show for $d = (1, 10, 25, 100)$ greedy strategy does not yield an opt. soln. for some N.

Properties to look for when designing a greedy algorithm

- optimal substructure.
- Greedy choice Property: Some greedy strategy works.

Minimum weight spanning trees (MWST)

Defn

Let G be a graph. A sub-graph T of G is a spanning tree iff

(1) T is a tree

and (2) $V(T) = V(G)$.

Thm

G contains a spanning tree iff G is connected.

Suppose G has a weight function on edes

$$w : E(G) \to \mathbb{R}.$$

<u>MWST Problem</u>

Given a weighted connected Graph G, find a <u>MWST</u> in G. (min total edge weight)

As usual:

- find value of opt. soln., i.e. weight of MWST
- find opt. soln. itself, edge set of a MWST.

Theorem

If the weights of edges are distinct, then G (connected) contains a unique MWST.

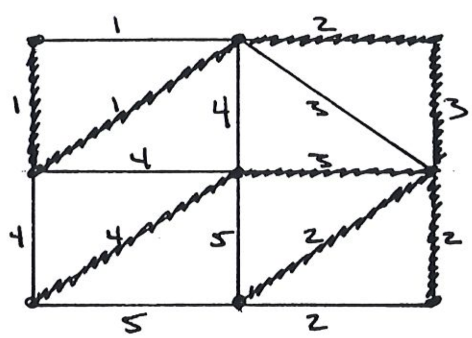
Algorithms:

- Kruskal (23.2)
- Prim (23.2)

PRIM'S ALGORITHM (23.2)

- CHOOSE AN INITIAL VERTEX (WHICH IS A TREE)
- AMONGST ALL EDGES INCIDENT WITH THE CURRENT TREE WHOSE ADDITION WOULD NOT CREATE A CYCLE, CHOOSE ONE OF MINIMUM WEIGHT.
- STOP WHEN $n-1$ EDGES HAVE BEEN SELECTED.

EX.



$W(T) = 18$

OBSERVE THAT AT EACH STAGE OF EXECUTION, PRIM'S ALGORITHM MAINTAINS A TREE SINCE NO CYCLES ARE CREATED AND ONLY INCIDENT EDGES ARE ADDED.

WHEN THIS TREE CONTAINS $n-1$ EDGES IT MUST HAVE n VERTICES (BY PREVIOUS THEOREM), HENCE IT IS A SPANNING TREE.

THEOREM

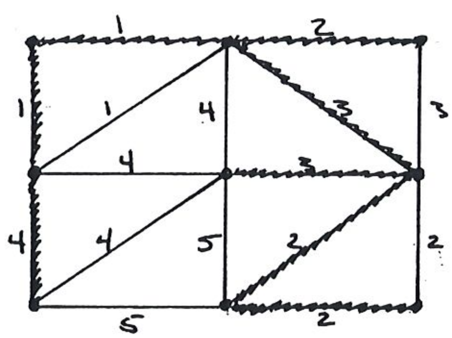
THIS SPANNING TREE HAS MINIMUM POSSIBLE WEIGHT.

(SEE BOOK OR TAKE CE 177 FOR PROOF.)

KRUSKAL'S ALGORITHM (23.2)

- CHOOSE AN EDGE OF MINIMUM WEIGHT
- AMONGST ALL EDGES WHICH DO NOT CREATE A CYCLE WITH PREVIOUSLY SELECTED EDGES, CHOOSE ONE OF MINIMUM WEIGHT.
- STOP WHEN $n-1$ EDGES HAVE BEEN SELECTED.

EX.



$w(T) = 18$

OBSERVE THAT AT EACH STAGE OF EXECUTION KRUSKAL'S ALGORITHM HAS CREATED A FOREST (UNION OF DISJOINT SUBTREES) SINCE NO CYCLES ARE CREATED.

WHEN THIS FOREST CONTAINS $n-1$ EDGES IT MUST ALSO HAVE n VERTICES. (ANY FOREST ~~GRAPH~~ WITH $n-1$ EDGES HAS AT LEAST n VERTICES. THIS FOREST CAN CONTAIN NO MORE THAN n VERTICES SINCE IT IS A SUBGRAPH OF G.)

THUS THE RESULTING FOREST IS CONNECTED (BY PREVIOUS THEOREM) AND IS A SPANNING TREE IN G.

<u>Theorem</u>

The sp. tree produced by Kruskal is a MWST in G.

<u>Proof</u>

let T be the sp. tree created by Kruskal, let S be any other sp. tree in G. we must show

$$w(T) \leq w(S)$$

Assume $S \neq T$.

Let $\{e_1, e_2, e_3, \ldots, e_{n-1}\} = E(T)$

in order added by Kruskal.

Since $S \neq T$ there is a first edge e_k not in S

i.e.

$$\{e_1, \ldots, e_{k-1}\} \subseteq E(S)$$

$$e_k \notin E(S)$$

Let H be subgraph

$$H = S + e_k$$

obtained by adding e_k to $E(S)$.

By t-ennes Thm, H contains a unique cycle, call it C.

Note C contains an edge e of S that is <u>not</u> in T (otherwise $E(C) \subseteq E(T)$, impossible since T is acyclic.)

Now remove e from H; call result R

$$R = H - e = S + e_k - e$$

Since R is connected and has $n-1$ edges, & no cycles, R is another sp. tree in G.

Kruskals algorithm guarantees that

$$w(e_k) \le w(e)$$

<u>claim</u> e does not form a cycle with $\{e_1, \dots, e_{k-1}\}$ since $\{e_1, \dots, e_{k-1}, e\} \subseteq E(S)$.

so R is a sp. tree in G with (at least) one more edge in common with T than S.

Also $w(R) \le w(S)$, since $w(e_k) \le w(e)$. If $R = T$, done. If $R \ne T$, repeat with R in place of S.

$$w(S) \ge \underset{\substack{\parallel \\ R_0}}{w(R)} \ge w(R_1) \ge \ldots \ge w(T)$$

eventually, some $R_q = T$.

$\therefore \quad w(S) \ge w(T)$. ∎