2.4 CONNECTOR PROBLEMS

DEFN.
A WEIGHTED GRAPH IS A GRAPH $G=(V,E)$ EQUIPED WITH A REAL VALUED WEIGHT FUNCTION

$$w : E \to \mathbb{R}$$

THE WEIGHT OF A SUBGRAPH $H \subseteq G$ IS

$$w(H) = \sum_{e \in E(H)} w(e)$$

A MINIMUM WEIGHT SPANNING TREE T SATISFIES

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

FOR ALL SPANNING TREES S IN G.

RECALL OUR ALGORITHM FOR CONSTRUCTING A SPANNING TREE WORKED FROM THE "TOP DOWN". i.e. WE SYSTEMATICALLY REMOVED CYCLE EDGES FROM G UNTIL NONE WERE LEFT.

ANOTHER APPROACH WOULD BE TO WORK FROM THE "BOTTOM UP", i.e. BUILD A SPANNING TREE BY ADDING EDGES SO AS NOT TO CREATE CYCLES.

LET $G=(V,E)$ BE CONNECTED, AND SET $E' = E - \{\text{loops}\}$.

<u>ALGORITHM FOR SPANNING TREE</u>

1.) $F \leftarrow \emptyset$
2.) while $|F| < n-1$
3.) CHOOSE $e \in E' - F$ SUCH THAT $(V, F \cup \{e\})$ is ACYCLIC.
4.) $F \leftarrow F \cup \{e\}$.

OBSERVE THAT STEP (3) CAN BE PERFORMED AS LONG AS $|F| < n-1$, SINCE WE CAN CHOOSE ANY e JOINING TWO COMPONENTS OF (V,F). e is THEN A BRIDGE IN $(V, F \cup \{e\})$ AND DOES NOT CREATE A CYCLE. IF NO SUCH e EXISTS, THEN (V,F) is CONNECTED, AND BEING ACYCLIC is A TREE. $\therefore$ $|F| = n-1$, CONTRADICTING $|F| < n-1$.

<u>THEOREM</u>

WHEN THE ALGORITHM is COMPLETE $T=(V,F)$ is A SPANNING TREE IN G.

PROOF.

THE ALGORITHM STOPS WHEN $|F| = n-1$. $T = (V,F)$ is ACYCLIC BY ITS VERY CONSTRUCTION, SO IS A SPANNING TREE. ///.

WE CAN SEE THAT SPANNING TREES ARE SPANNING SUBGRAPHS WITH DUAL CHARACTERIZATIONS :

(i) MINIMAL WITH RESPECT TO THE CONNECTED PROPERTY

(ii) MAXIMAL WITH RESPECT TO THE ACYCLIC PROPERTY.

A SLIGHT ALTERATION OF THE ABOVE ALGORITHM FINDS A MINIMUM WEIGHT SPANNING TREE IN A WEIGHTED GRAPH. LET $G = (V,E)$ BE CONNECTED, $w: E \to \mathbb{R}$, AND $E' = E - \{\text{loops}\}$.

KRUSKAL

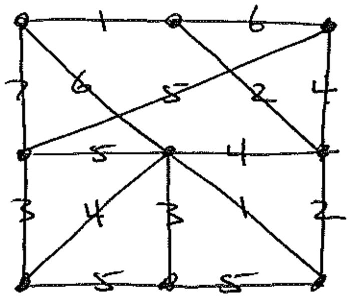
1.) $F \leftarrow \emptyset$

2.) while $|F| < n-1$

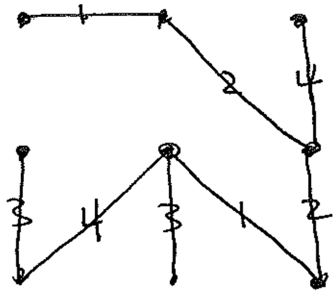
3.) CHOOSE $e \in E' - F$ SUCH THAT
   (i) $(V, F \cup \{e\})$ is ACYCLIC.
   (ii) $w(e)$ is MINIMUM SUBJECT TO (i)

4.) $F \leftarrow F \cup \{e\}$.

EX.



G



$W(T) = 20$.

THEOREM.

WHEN KRUSKAL IS COMPLETE, $T = (V, F)$ IS A MINIMUM WEIGHT SPANNING TREE.

PROOF.

WE'VE ALREADY SEEN THAT T IS A SPANNING TREE IN G. SUPPOSE S IS A SPANNING TREE OTHER THAN T. WE MUST SHOW

$$W(T) \leq W(S).$$

LET $e_1, e_2, \ldots, e_{n-1}$ BE THE EDGES OF T IN THE ORDER PRODUCED BY KRUSKAL'S ALGORITHM. LET e_K BE THE FIRST EDGE IN T WHICH IS NOT IN S.

i.e. $\{e_1, \ldots, e_{K-1}\} \subseteq E(S)$ BUT $e_K \notin E(S)$.

LET $H = S + e_K$. BY THE "TREENESS" THEOREM, e_K BELONGS TO A UNIQUE CYCLE C IN H.

NOW C MUST CONTAIN AN EDGE e WHICH IS NOT IN T (OTHERWISE T CONTAINS THE CYCLE C.)

THUS $R = H - e = (S + e_K) - e$ IS CONNECTED, AND SINCE $|E(R)| = n-1$, R IS ANOTHER SPANNING TREE IN G.

NOTE THAT $w(e_K) \leq w(e)$, FOR OTHERWISE KRUSKAL WOULD HAVE CHOSEN e ON THE K^{TH} STEP INSTEAD OF e_K.

(OBSERVE THAT $(V, \{e_1, .., e_{K-1}, e\})$ IS ACYCLIC SINCE $\{e_1, \ldots, e_{K-1}, e\} \subseteq E(S)$.)

THUS $w(R) \leq w(S)$ AND R CONTAINS ONE MORE EDGE OF T THAN S DOES. i.e. $\{e_1, \ldots, e_{K-1}, e_K\} \subseteq E(R)$.

NOW REPEAT THIS WHOLE PROCESS WITH R IN PLACE OF S TO OBTAIN A SPANNING TREE WITH ONE MORE EDGE IN COMMON WITH T, AND HAVING WEIGHT AT MOST $w(R)$.

CONTINUING IN THIS MANNER WE EVENTUALLY REACH T. THEREFORE

$$w(T) \le \dots \le w(R) \le w(S) \quad .$$

///.

NOTE THAT KRUSKAL REQUIRES THAT WE ARE ABLE TO DETECT CYCLES, WHICH CAN REQUIRE MORE WORK. (BUT SEE CORMAN)

PRIM'S ALGORITHM WORKS BY ADDING LEAVES TO A GROWING TREE, THEREBY AVOIDING CYCLES WITHOUT HAVING TO DETECT THEM.

AS BEFORE LET $G=(V,E)$ BE CONNECTED, AND SET $E' = E - \{loops\}$.

<u>PRIM</u>.

1.) CHOOSE $v \in V$

2.) CHOOSE $e \in E'$ SUCH THAT $w(e)$ is MINIMUM AMONG ALL EDGES INCIDENT WITH v.

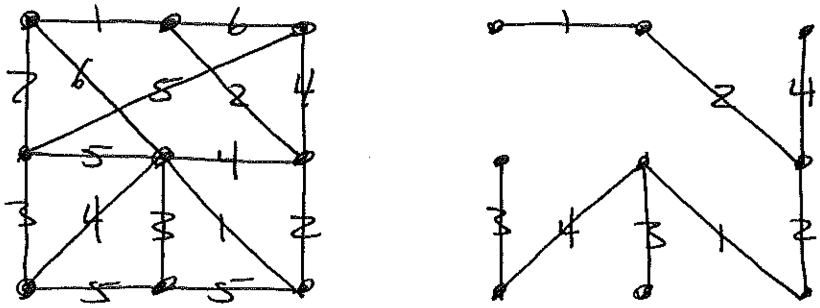
3.) $F \leftarrow \{e\}$

4.) while $|F| < n-1$

5.) CHOOSE $e \in E' - F$ SATISFYING
(i) e HAS EXACTLY ONE END IN $G[F]$
(ii) $w(e)$ is MINIMUM SUBJECT TO (i)

6.) $F \leftarrow F \cup \{e\}$.

Ex.



G W(T) = 20.

REMARKS

(1.) OBSERVE THAT ON EACH ITERATION OF LOOP 4-6 THE GRAPH G[F] IS CONNECTED SINCE WE ALWAYS ADD AN EDGE ADJACENT TO A PREVIOUS ONE.

(2.) ALSO G[F] IS ACYCLIC SINCE WE ALWAYS CREATE A NEW LEAF, SO THE NEW EDGE IS NOT PART OF A CYCLE. THUS G[F] IS A TREE ON EVERY ITERATION OF LOOP 4-6.

(3.) NOTE THAT STEP (5) IS ALWAYS POSSIBLE SINCE G[F] CANNOT INCLUDE ALL VERTICES UNTIL $|F| = n-1$. THEREFORE AT LEAST ONE EDGE SATISFYING (5i) EXISTS AS LONG AS $|F| < n-1$.

(4) WHEN PRIM IS COMPLETE, $|F| = n-1$ AND $T = G[F]$ IS A SPANNING TREE IN G.

(5) OBSERVE THAT PRIM WORKS BY GROWING A SINGLE TREE UNTIL A SPANNING TREE IS REACHED, WHILE KRUSKAL MAY GROW SEVERAL TREES SIMULTANEOUSLY.

THEOREM

WHEN PRIM IS COMPLETE $T = G[F]$ IS A MINIMUM WEIGHT SPANNING TREE IN G.

PROOF.

THE ABOVE REMARKS SHOW THAT T IS A SPANNING TREE. LET S BE ANY OTHER SPANNING TREE. WE MUST SHOW $w(T) \leq w(S)$.

LET $e_1, e_2, \ldots, e_{n-1}$ BE THE EDGES OF T IN THE ORDER PRODUCED BY PRIM'S ALGORITHM. LET T_i BE THE SUBTREE CREATED BY THE ADDITION OF e_i. i.e.

$$T_i = G[\{e_1, \ldots, e_i\}] \qquad 1 \leq i \leq n-1.$$

LET e_K BE FIRST EDGE IN T WHICH IS NOT IN S. i.e.

$$\{e_1, \ldots, e_{K-1}\} \subseteq E(S) \text{ BUT } e_K \notin E(S).$$

SUPPOSE e_K HAS ENDS u, v. LET P BE THE UNIQUE PATH IN S FROM u TO v. ($\therefore$ P DOES NOT INCLUDE e_K).

BY CONSTRUCTION ONE END OF e_K IS IN T_{K-1}, WHILE THE OTHER IS NOT. SAY u BELONGS TO T_{K-1} AND v DOES NOT.

SINCE P IS A u-v PATH IT MUST CONTAIN SOME EDGE e WITH ONE END IN T_{K-1} AND THE OTHER NOT IN T_{K-1}.

NECESSARILY $w(e_K) \le w(e)$ SINCE OTHERWISE PRIM WOULD HAVE CHOSEN e AT THE K^{TH} ITERATION RATHER THAN e_K.

NOW $P + e_K$ IS A CYCLE SO IF WE SET

$$R = (S + e_K) - e,$$

THEN R IS A CONNECTED SUBGRAPH ON $n-1$ EDGES, n VERTICES, HENCE IS ANOTHER SPANNING TREE IN G.

BUT $w(R) \leq w(S)$ SINCE $w(e_k) \leq w(e)$, AND R HAS ONE MORE EDGE IN COMMON WITH T THAN S DOES. *

REPEAT THIS WHOLE PROCESS WITH R IN PLACE OF S TO OBTAIN YET ANOTHER SPANNING TREE WITH MORE IN COMMON WITH T, AND OF WEIGHT AT MOST $w(R)$.

CONTINUING IN THIS MANNER WE EVENTUALLY REACH T ITSELF AND A CHAIN OF INEQUALITIES

$$w(T) \leq \ldots \leq w(R) \leq w(S) \quad ,$$

///.

* ACTUALLY R & S MAY HAVE THE SAME NUMBER OF EDGES IN COMMON WITH T SINCE e MAY BELONG TO T. BUT e CANNOT BE AMONGST THE EDGES $\{e_1, .., e_{k-1}\}$ SINCE OTHERWISE IT WOULD HAVE BOTH ENDS IN T_{k-1}. THUS $\{e_1,..,e_{k-1}\} \subseteq E(S)$ AND $e_k \notin E(S)$ WHILE $\{e_1, .., e_k\} \subseteq E(R)$. THEREFORE BY REPEATING THE PROCESS WE EVENTUALLY REACH T, AS REQUIRED.

CORRECTION TO PROOF OF PRIM'S

THE STATEMENT "R HAS ONE MORE EDGE IN COMMON WITH T THAN S DOES" IS INCORRECT, AS EXAMPLE SHOWS. HOWEVER THE FOLLOWING *IS* TRUE:

IF e_l IS THE FIRST EDGE IN T WHICH IS NOT IN R, THEN $l > k$. THUS, BY REPEATING THIS PROCESS, WE DO EVENTUALLY REACH T

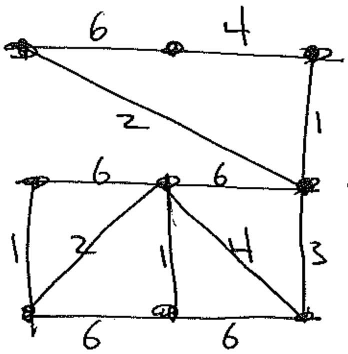
$\therefore\ w(T) \le w(S)$

AS REQUIRED.

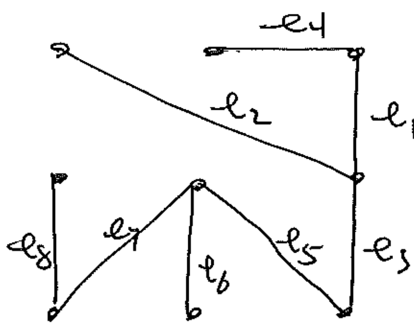
$|S \cap T| = 3$

EX

G

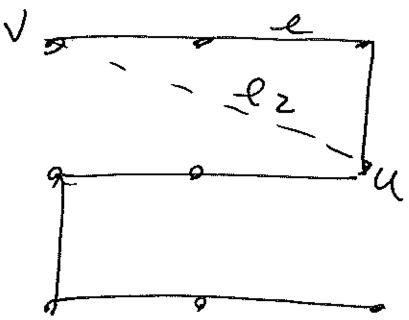


T

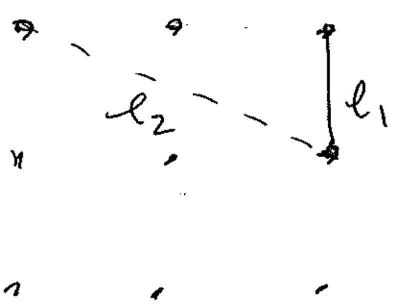


S

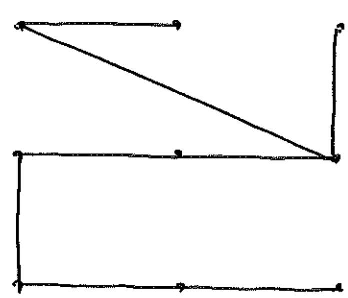
$k = 2$



$e = e_4 \in T$!!



T_1



R

$|R \cap T| = 3$

FORM $R = (S + e_2) - e$

$w(R) \le w(S)$

BUT: THE FIRST EDGE IN T WHICH IS NOT IN R IS e_3 AND $3 > 2$.