CHAPTER FIVE

TREES AND CUT-SETS

5.1 Exactly two vertices have degree 1, all others have degree 2. Thus, the graph looks like

5.2 There are $2n + 3n + n = 6n$ vertices, and hence $6n - 1$ edges. Thus

$$2(6n - 1) = 1(2n) + 2(3n) + 3(n)$$

Hence $n = 2$, and there are 12 vertices and 11 edges.

5.3 (a) 9

(b) Let e denote the number of edges in the tree.

$$2e = n_1 + 2n_2 + 3n_3 + \ldots + kn_k$$

Also, $$2e = 2(v - 1) = 2(n_1 + n_2 + n_3 + \ldots + n_k - 1)$$

$$n_1 = n_3 + 2n_4 + \ldots + (k - 2)n_k + 2$$

5.4 $n_1 + n_2 = 51$

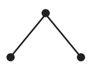
$$n_1 = e_2 = n_2 - 1$$

Thus $n_1 = 25$, $e_1 = 24$, and $n_2 = 26$, $e_2 = 25$.

T_1 has 25 vertices 24 edges, while T_2 has 26 vertices and 25 edges.

5.5 (a) There are $v - 1$ edges.

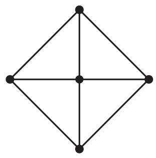
(b) $v = 2$, $d_1 = d_2 = 1$



$v = 3$ (2, 1, 1)

Assume for $n = v - 1$, $\sum_{i=1}^{v-1} d_i = 2(v - 1) - 2$ yields a tree. Let $(d_1, d_2, \ldots, d_v)$ be an ordered v-tuples such that $\sum_{i=1}^{v} d_i = 2v - 2$. There exists i such that $d_i = 1$. $\left(\text{If not, } \sum d_i \geq 2v\right)$. There exists j such that $d_j > 1$. $\left(\text{If not, } \sum d_i = v < 2v - 2 \text{ for } v > 2\right)$. We can remove d_i, change d_j to $d_j - 1$ and apply the induction hypothesis.

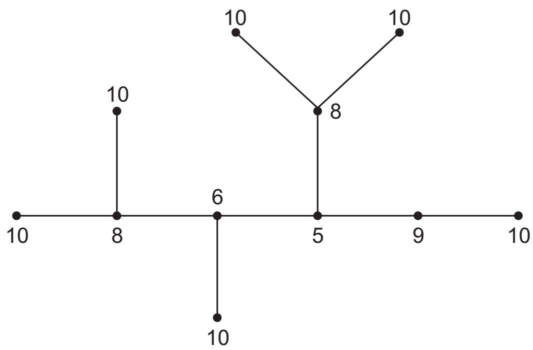
5.6 (a)



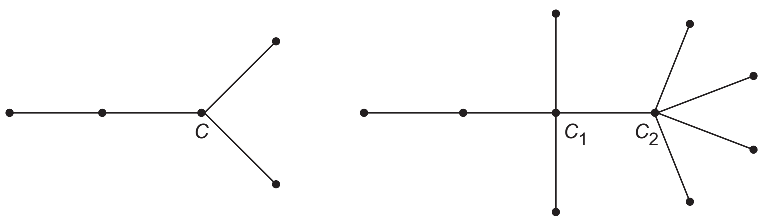
(b) K_4

(c) We define the *eccentricity* $e(v)$ of a vertex v in a connected graph G to be max $d(u, v)$ for all u in G. Let T be a tree. Let T' be a tree obtained from T by removing all leaves of T. Clearly, the eccentricity of each vertex of T' is exactly one less than the eccentricity of the same vertex of T. Hence, the vertices of T which possess minimum eccentricity in T are the same vertices having minimum eccentricity in T'. Then is, T and T' have the same center. Repeating the argument, we shall terminate either at a tree that is single vertex or at a tree that has one edge and two vertices.

5.7 (a)



(b)



(c) Let v be a centroid. Let $v_1, v_2, \ldots, v_k$ and $s_1, s_2, \ldots, s_k$ be as defined. Let $T_1, T_2, \ldots, T_k$ denote the subtrees with $v_1, v_2, \ldots, v_k$ as roots. Clearly, the weight of v_1 is at least $n - s_1 = 1 + s_2 + s_3 + \ldots + s_k$ where n is the number of vertices in the tree, since the subtree with v as the root has $n - s_1$ vertices. If there is a centroid, y_1, in subtree T_1, then $wt(y_1) \geq n - s_1 = 1 + s_2 + s_3 + \ldots + s_k$ and $wt(y_1) = wt(v) = \max(s_1, s_2, \ldots, s_k) \geq 1 + s_2 + \ldots + s_k$ and this implies $s_1 > s_2 + \ldots + s_k$. A similar argument can be applied to other subtrees $T_2, T_3, \ldots, T_k$. Therefore, at most one of the subtrees, say T_j, may contain a centroid, v_j. Furthermore, T_j is the heaviest subtree of v, and conversely, the subtree with v as the root is the heaviest subtree of y_j. Thus, if y_j is not adjacent to v, then v_j must be on the path from y_j to v and it implies $wt(y_j) \geq wt(v) + 1$ which is a contradiction. Therefore, v and y_j must be adjacent.

(d) From (c), it is shown that if there is a centroid in subtree T_j, then $s_j > s_1 + \ldots + s_k - s_j$. So, if $s_j \leq s_1 + \ldots + s_k - s_j$ then there is no such centroid in that subtree. If it is true for $1 \leq j \leq k$, then no more centroids exist besides v.

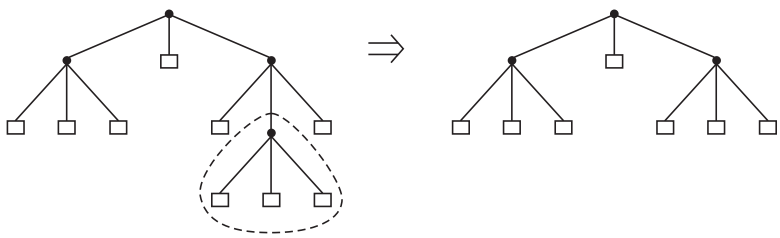
(e) From c, the two centroids v, y_j are of same weight and adjacent to each other and v is the root of the heaviest subtree of y_j and y_j is in the root of the subtree of v. Thus, these two subtrees partition the n vertices in the tree. It follows that $n = wt(v) + wt(y_j) = 2 \times wt(v)$.

5.8 There is one vertex of degree 2. All other vertices are of degree 3 or 1. There must be an even number of vertices of degree 3 or 1.

5.9 To show that number of leaves $t = (m - 1)i + 1$.

Basis: $i = 0 \qquad t = 1 = (m - 1)\,0 + 1$

Induction Step: Given a regular m-ary tree T, with i internal nodes, there must be one with m leaves as its sons. (Choose an internal nodes that is furthest from the root.) Remove this internal node and all its son and replace them with a leaf as shown below. We have a regular m-ary tree, T', with $i - 1$ internal nodes. By induction hypothesis, the number of leaves in T' $= (m - 1)(i - 1) + 1$. Therefore the number of leaves in T must be $(m - 1)(i - 1) + 1 + (m - 1) = (m - 1)\,i + 1$.

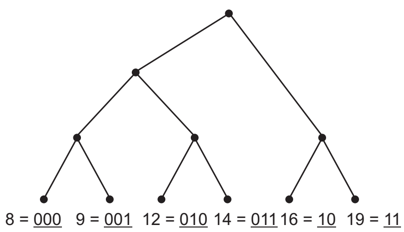


5.10 $a\ b\ d\ g\ e\ h\ i\ c\ f$
$g\ d\ b\ h\ e\ i\ a\ c\ f$
$g\ d\ h\ i\ e\ b\ f\ c\ a$

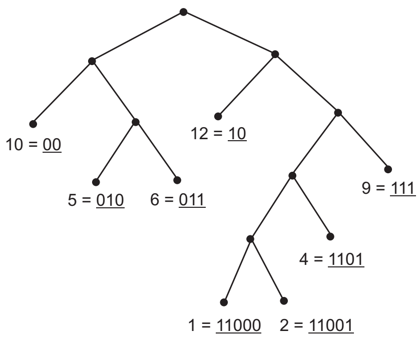
5.11 Assume $A \neq \phi$. We construct a binary tree as follows: Assign A to the root. Assign A_0 to the left son of A, and A_1 to the right son of A with the corresponding branches being 0 and 1, respectively. Such a repeated partition will yield a binary tree each of its leaves corresponds to a sequence in A.

5.12 (a) $(73 + 44 - 1) + (73 + 44 + 100 - 1) + (73 + 44 + 100 + 55 - 1) = 603$
(b) $(73 + 44 - 1) + (100 + 55 - 1) + (73 + 44 + 100 + 55 - 1) = 541$
(c) Merge A_2 and A_4, then A_1 and $A_2\ A_4$, and then A_3 and $A_1\ A_2\ A_4$. Total number of comparisons = 540.
(d) Construct a Huffman tree with the lengths of $A_1, A_2, \ldots, A_m$ as the weights of the leaves.

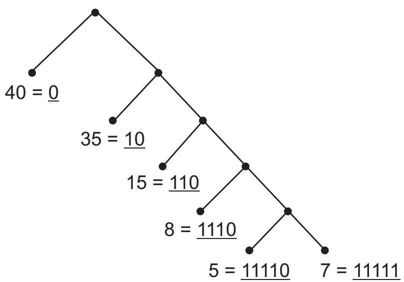
5.13 (a)



(b)

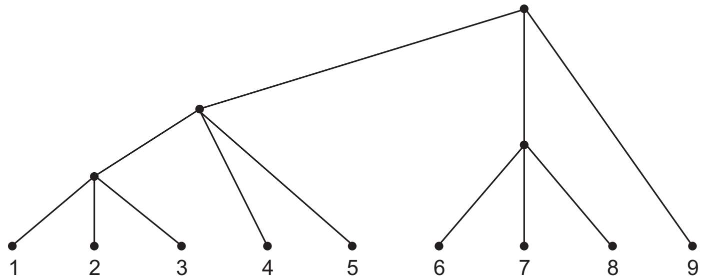


(c)

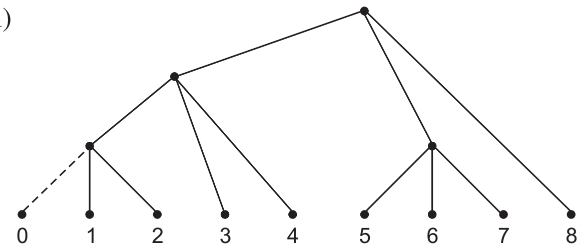


5.14 (a) Select the m smallest weights in each step.

(b)



(c) (i)



(ii) Put in dummy weights of value 0.

5.15 Choose $K_{\lfloor n/2 \rfloor}$ as the root and construct recursively with the left subtree containing the keys $K_1, K_2, \ldots, K_{\lfloor n/2 \rfloor - 1}$ and the right subtree containing the keys $K_{\lfloor n/2 \rfloor + 1}, \ldots, K_n$.

5.16 (a) (i) Examine the root. If it has one key then it will have two sons, and by comparison if it is the key we want we are done. If it is greater, we search to the left, otherwise to the right. If the root has two keys then we compare with the first. If it equals the one we want, we are done; if it is greater, we search the left subtree; if it is less, we compare with the second key. If equal, we are done; if greater, we search the center subtree; if less, we search the right subtree.

(ii) Let k denote the number of keys. Let v_1 denote the number of leaves, v_2 denote the number of internal nodes with 1 key and v_3 denote the number of internal nodes of 2 keys. Let e denote the number of edges.

(1) $k = 2v_3 + v_2$

(2) $e = 3v_3 + 2v_2$

(3) $e = v_3 + v_2 + v_1 - 1$

From (1), (2), and (3) we obtain $k = v_1 - 1$.

5.17 A spanning tree has at least one edge in common with every cut-set by Theorem 5.2. Therefore, the complement of a spanning tree lacks one or more edges from every cut-set.

A cut-set has at least one edge in common with every spanning tree, again by Theorem 5.2. Therefore, the complement of a cut-set lacks one or more edges of every spanning tree.

5.18 Let T be a spanning tree in G containing all the edges $L - a$. Let C be the fundamental cut-set, relative to T, corresponding to the tree branch b. Then,

$$\{L - a\} \cap C = b$$

and by Theorem 6.3.

$$L \cap C = \{a, b\}$$

5.19 Let C_1 be the fundamental cut-set relative to T_1 corresponding to edge a.

$$C_1 \cap T_1 = \{a\} \tag{1}$$

Since $a \notin T_2$, a is a chord of T_2 and defines a fundamental circuit L relative to T_2.

$$L \cap \overline{T_2} = \{a\} \tag{2}$$

Applying Theorem 5.3 $L \cap C_1$ contains an even number of edges, one of which is a. Let b be any other. Since $b \in L$, by (2) $b \in T_2$. Since $b \in C_1$, by (1) $b \notin T_1$, and $(T_1 - \{a\}) \cup \{b\}$ is a spanning tree. Let C_2 be the fundamental cut-set relative to T_2 corresponding to edge b.

$$C_2 \cap T_2 = \{b\} \tag{3}$$

Applying Theorem 5.3 $L \cap C_2$ contains an even number of edges, one of which is b. By (3) all other lie in T_2. By (2), $L \cap C_2 = \{a, b\}$. Since $a \in C_2$, $(T_2 - \{b\}) \cup \{a\}$ is a spanning tree.

5.20 (a) Let $L = L_1 \oplus L_2$. Then $L \subseteq L_1 \cup L_2$ and $a \notin L$, $b \in L$.

Thus, $\quad L \subseteq L_1 \cup L_2 - \{a\}$ and $b \in L$.

Now. L is either a circuit or edge-disjoint union of circuits. If L is a circuit, let $L_3 = L$. If L is an edge-disjoint union of circuits, let L_3 be the one circuit in L that contains the edge b.

(b) In the solution for part (a), replace "circuit" with "cut-set".

5.21 Exactly as that for Theorems 5.4 except interchange circuit (C) and chord with cut-set (D) and branch, respectively.

5.22 (a) 4 1 6 4

(b) Whenever an edge is deleted, the degree of the "non-leaf" vertex is decreased by 1. So, if this vertex appears i times, its degree is decreased by i. Finally, this vertex is removed as a leaf or is left as one of the two connected vertices. Its degree is 1 in either case.

(c) Let $d(i)$ be the degree of vertex i which can be calculated according to (b).

(1) set $i = 1$

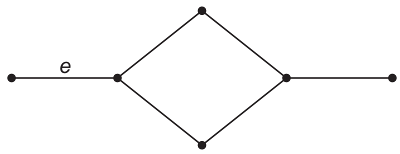
(2) Among all vertices whose present degree is 1, find the one, v_i, with the least name. Draw an edge connecting v_i to vertex a_i. Eliminate vertex v_i from the list and reduce $d(a_i)$ by one.

(3) If $i < n - 2$, increase i by 1 and go to step (2). If $i = n - 2$, draw an edge between the two remaining vertices (of degree 1) and stop.

(d) Clearly, the algorithm always yields a tree. Furthermore, the tree yields will be the only one which will produce the original number sequence.

5.23 (a) Let T be a spanning tree of G. If edge e is in T, then we are done. Otherwise, adding e in T will form a cycle. Remove an edge (other than e) from this cycle will give a tree T' containing e.

(b) No. A counter-example is the edge e below



5.24 (a) No. In G_1, there is a path from v_1 to v_2 and in G_2 there is a path from v_4 to v_3. Thus there is a cycle in G containing v_1, v_2, v_3 and v_4.

(b) Given any two vertices u and v, if they both belong to either G_1 or G_2, then clearly there is a path from u to v. Otherwise, assume that u is in G_1 and v is in G_2. Then there is a path from u to v as follows: First follow path in G_1 from u to v_1, then follow edge $\{v_1, v_3\}$ and then follow path in G_2 from v_3 to v.

(c) G_1 is a path with v_1 and v_2 as its end points. Similarly, G_2 is a path with v_3 and v_4 as its end points.

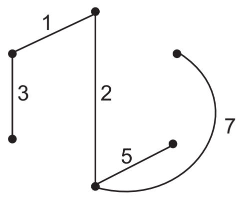
5.25 (a) 4

(b) Without $\{a, c\}$ 6, with $\{a, c\}$ $2 \cdot 4$, total 14

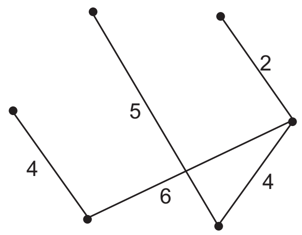
(c) 8

(d) 8^5

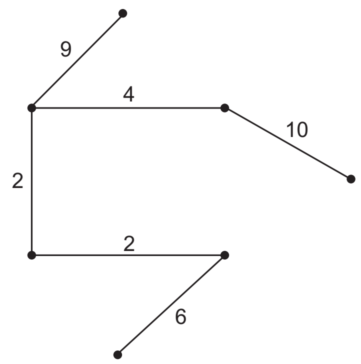
5.26



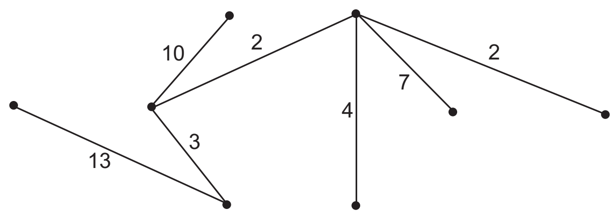
5.27



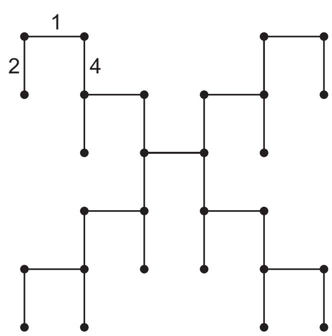
5.28



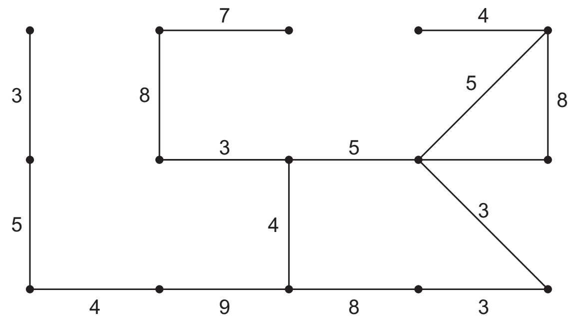
5.29



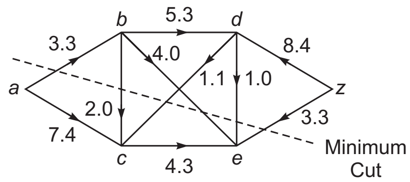
5.30



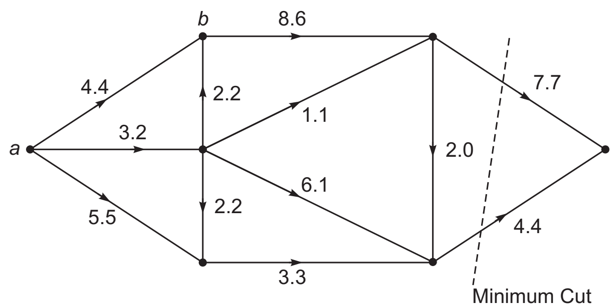
5.31



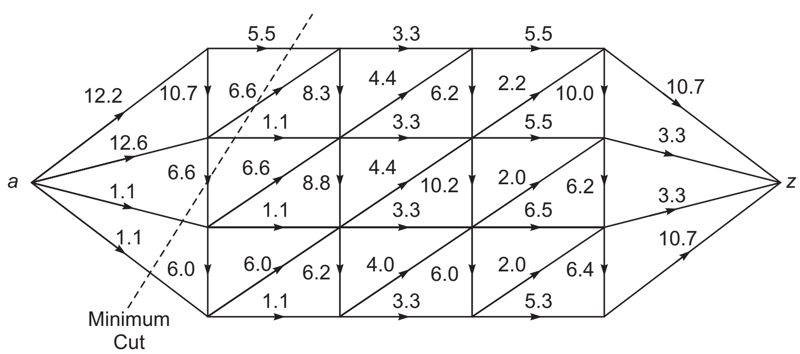
5.32 A maximum flow, with $\phi_v = 7$, is



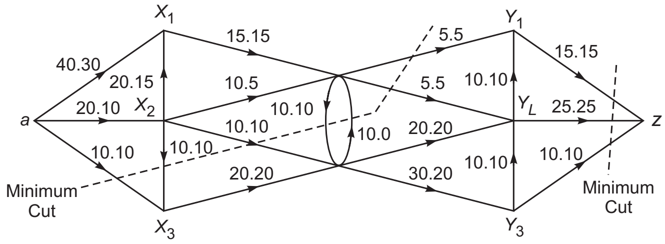
5.33 A maximum flow, with $\phi_v = 11$, is



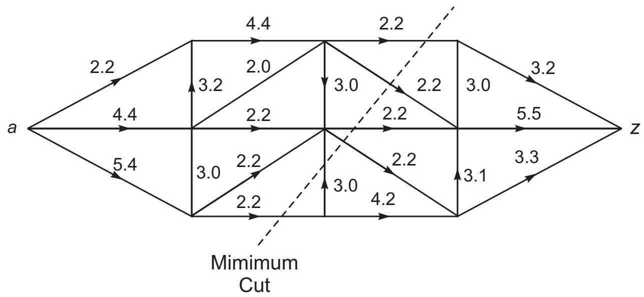
5.34 A maximum flow, with $\phi_v = 20$, is



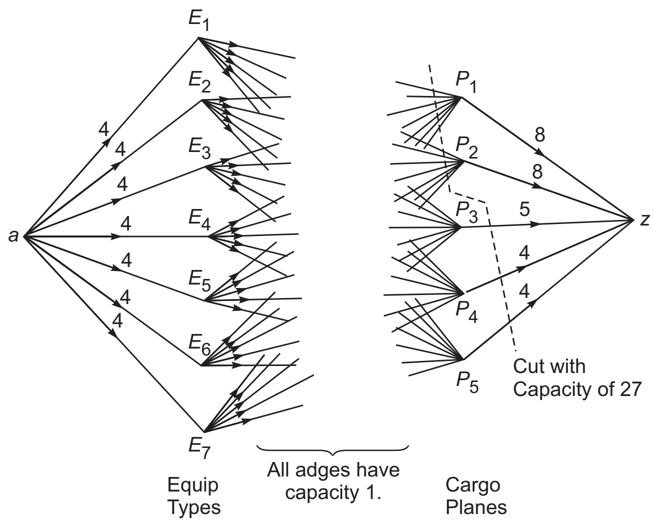
5.35 Finding a maximum flow in the following transport network reveals that the demands of all three depots can be met by having factory x_1 make 30 units, x_2 10 units and x_3 10 units.



5.36 $\phi_V = 10$



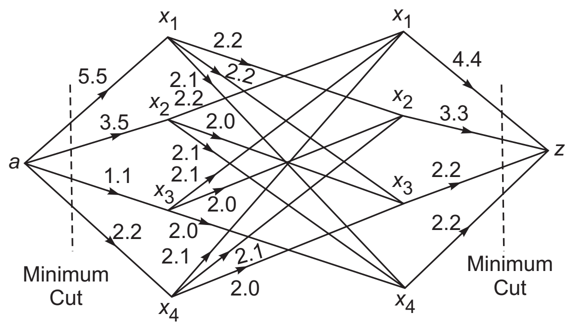
5.37 (a) A flow in the following transport network specifies a loading plan for some or all of the equipment such that no two similar units are on the same plane. Since the capacity of the cut shown below is 27, there can be no flow having value 28. Thus, not all the equipment can be so loaded.



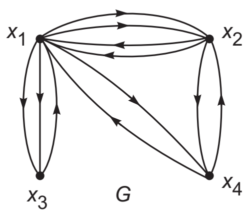
(b) Applying the labeling procedure to the transport network of part (a), with capacities of edges incident with the sink modified, yields the following loading plan as one possible solution.

	P_1	P_2	P_3	P_4	P_5
E_1	x	x	x	x	
E_2	x	x	x	x	
E_3	x	x	x	x	
E_4	x	x	x		x
E_5	x	x	x		x
E_6	x	x	x		x
E_7	x	x		x	x

5.38



$\phi(x_i, x_j')$ indicates the number of directed edges in G from x_i to x_j



5.39 (a) Define the capacity of a cut as

$$\alpha(P, P) = \begin{cases} \sum\limits_{i \in P, j \in \overline{P}} \alpha(i\ j) & \text{if there are no } edges \text{ from } \overline{P} \text{ to } P \\ 0 & \text{otherwise} \end{cases}$$

A Minimum flow-maximum cut theorem:

In a transport network in which the flow in each edge is lower bounded only, the minimum value that a flow can achieve is equal to the maximum value of the capacities of the cuts in the network.

(b) Modified labeling procedure.
Assume an initial feasible flow.
Label the sink z with $(-, \infty)$
Label a vertex y adjacent to z with $(z^-, \delta(y))$

where $\quad \delta(y) = \phi(y, z) - \alpha(y, z)$, if $\phi(y, z) > \alpha(y, z)$.

Do not label otherwise.
Label a vertex x adjacent to a labeled vertex y and $(y^-, \delta(x))$

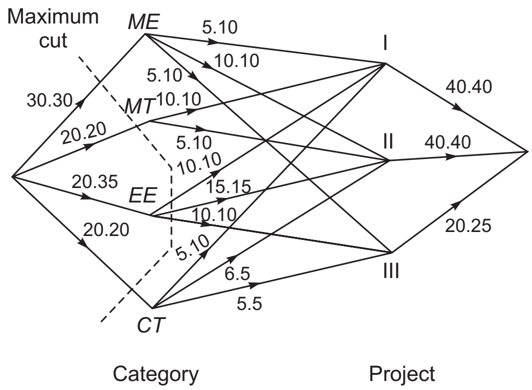
where $\quad \delta(x) = \min\,[(\phi(x, y), -\alpha(x, y)), \delta(y)]$, if $\phi(x, y) > \alpha(x, y)$.

Do not label otherwise.
Label a vertex w adjacent *from* a labeled vertex y with $(y^+, \delta(w))$ where $\delta(w) = \delta(y)$.
(It is always possible to label a vertex adjacent *from* a labeled vertex.)

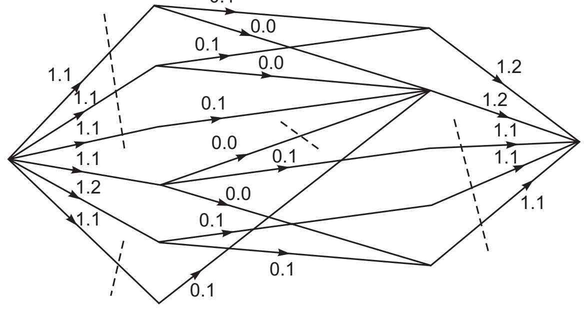
Each time the source a is labeled, reduce the flow in accordance with the labels and begin again.

The algorithm terminates when the source cannot be labeled. At that point, if the unlabeled vertices are denoted P, and the labeled vertices $\overline{P}$, then the cut $(P, \overline{P})$ is a maximum cut with no edges from $\overline{P}$ to P and each edge from P to $\overline{P}$ carrying a minimum flow.

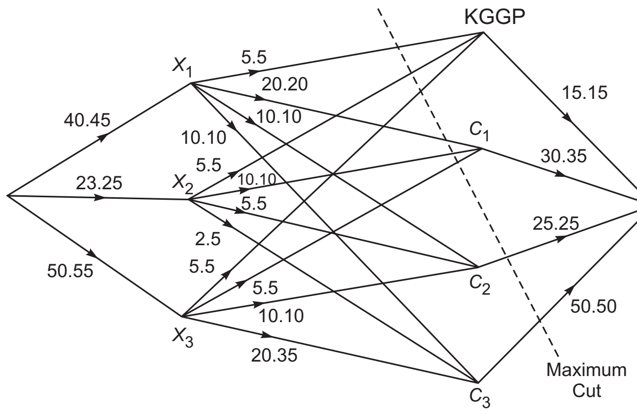
5.40



5.41



5.42



All capacities and flows are in thousands of dollars.