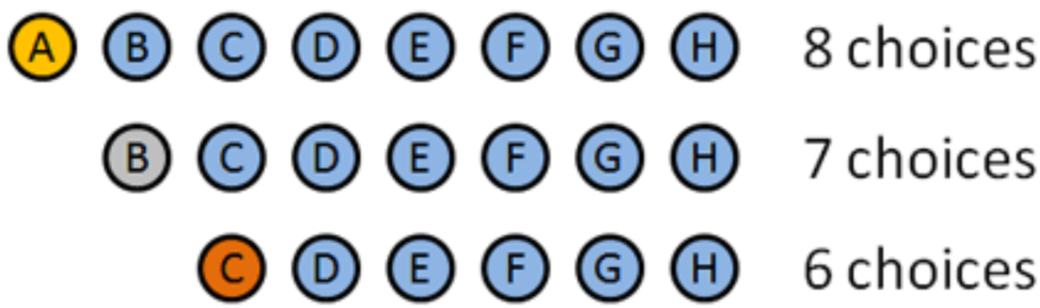
Quiz 06

Chapter 6: Counting Methods

Exercise 1:

How many ways can we award a 1st, 2nd and 3rd place prize among eight contestants? (Gold / Silver / Bronze)



We're going to use permutations since the order we hand out these medals matters. Here's how it breaks down:

Gold medal: 8 choices: A B C D E F G H (Clever how I made the names match up with letters, eh?). Let's say A wins the Gold.
Silver medal: 7 choices: B C D E F G H. Let's say B wins the silver.
Bronze medal: 6 choices: C D E F G H. Let's say… C wins the bronze.
We picked certain people to win, but the details don't matter: we had 8 choices at first, then 7, then 6. The total number of options was $8 \cdot 7 \cdot 6 = 336$.

Let's look at the details. We had to order 3 people out of 8. To do this, we started with all options (8) then took them away one at a time (7, then 6) until we ran out of medals.

And why did we use the number 5? Because it was left over after we picked 3 medals from 8. So, a better way to write this would be: $\frac{8!}{(8-3)!}$

Example 2: In how many ways can we select a committee of two women and three men from a group of five distinct women and six distinct men?

Since a committee is an ordered group of people, we find the two women can be selected in

C(5, 2) = P(5, 2) / 2! = 10 ways

Since a committee is an ordered group of people, we find the three men can be selected in

C(6, 3) = P(6, 3) / 3! = 20 ways

The committee can be constructed in two successive steps: Select the women; select the men. By the Multiplication Principle, the total number of committees is

10 x 20 = 200.

Chapter 7: Recurrence relations

Exercise: Solve the recurrence relation

1) Fn = 10Fn−1 − 25Fn−2 where F0 = 3 and F1 = 17

Solve the recurrence relation $F_n = 10F_{n-1} - 25F_{n-2}$ where $F_0 = 3$ and $F_1 = 17$

Solution

The characteristic equation of the recurrence relation is −

$x^2 - 10x - 25 = 0$,

So, $(x - 5)^2 = 0$

Hence, there is single real root $x_1 = 5$

As there is single real valued root, this is in the form of case 2

Hence, the solution is −

$F_n = ax_1^n + bnx_1^n$

$3 = F_0 = a.5^0 + b.0.5^0 = a$

$17 = F_1 = a.5^1 + b.1.5^1 = 5a+5b$

Solving these two equations, we get a = 3 and b = 2/5

Hence, the final solution is −

$F_n = 3.5^n + (2/5) * n.2^n$

2) Fn = 5Fn−1 − 6Fn−2 where F0 = 1 and F1 = 4

Solution

The characteristic equation of the recurrence relation is −

$x^2 - 5x + 6 = 0$,

So, $(x - 3)(x - 2) = 0$

Hence, the roots are −

$x_1 = 3$ and $x_2 = 2$

The roots are real and distinct. So, this is in the form of case 1

Hence, the solution is −

$F_n = ax_1^n + bx_2^n$

Here, $F_n = a3^n + b2^n$ (As $x_1 = 3$ and $x_2 = 2$)

Therefore,

$1 = F_0 = a3^0 + b2^0 = a+b$

$4 = F_1 = a3^1 + b2^1 = 3a+2b$

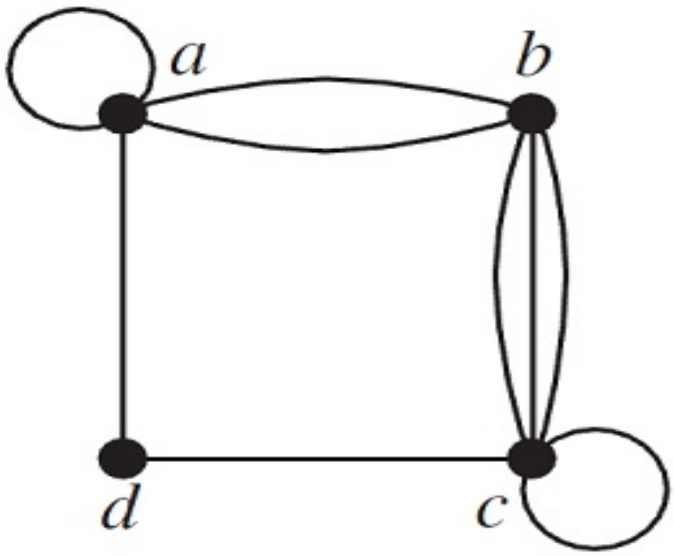
Solving these two equations, we get $a = 2$ and $b = -1$

Hence, the final solution is −

$F_n = 2.3^n + (-1) * 2^n = 2.3^n - 2^n$

__Chapter 8:__

<u>Exercise 1:</u> Consider the graph G:



1) Find the adjacency Matrix of the graph G

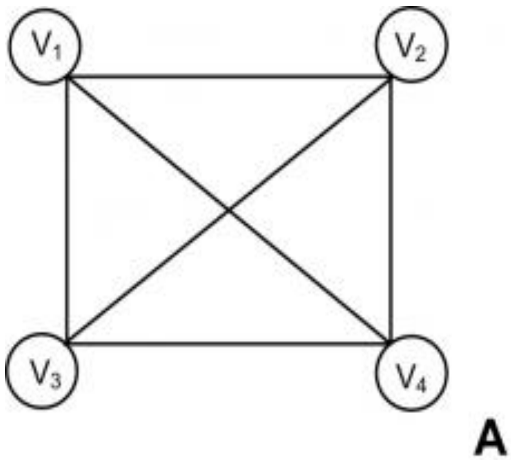
	a	b	c	d
a	2	2	0	1
b	2	0	3	0
c	0	3	2	1
d	1	0	1	0

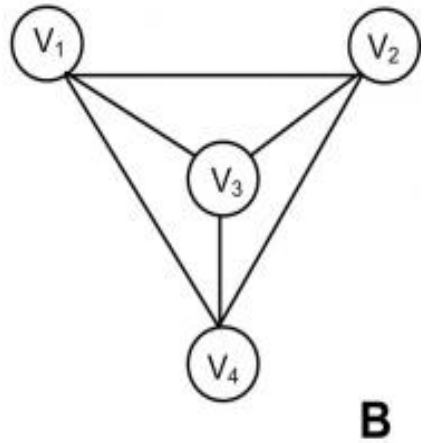
2) Find the square of the Matrix A

	a	b	c	d
a	9	4	7	2
b	4	13	6	5
c	7	6	14	2
d	2	5	2	2

Exercise 2:

I. Let consider the Graphs A and B:





1) Can we define A and B as Isomorphic graphs. Why?

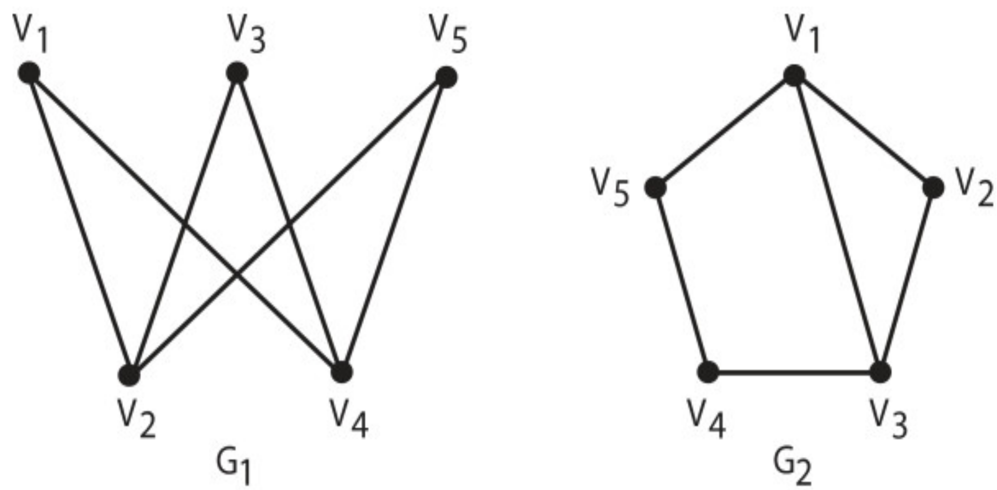
Graph Isomorphism. V = {V 1 , V 2 , V 3 , V 4 }, |V| = 4, E = {(V 1 , V 2), (V 1 , V 3), (V 1 , V 4), (V 2 , V 3), (V 2 , V 4), (V 3 , V 4)}, |E| = 6. Graphs A and B have different topology but they are isomorphs. The graph is fully connected and every node is connected to any other so that it forms a fully connected clique.

2) Find the Adjacency Matrix of the graphs A and B.

	V1	V2	V3	V4
V1	0	1	1	1
V2	1	0	1	1
V3	1	1	0	1
V4	1	1	1	0

Both graph A and B have the same matrix.

II. Let consider the Graphs G1 and G2:



1) Can we define G1 and G2 as Isomorphic graph. Why?

Two non-isomorphic graphs each with five vertices. Note that both graphs have two vertices of degree 3 and three vertices of degree 2. To see that the two graphs are not isomorphic, consider the following: we say a graph is bipartite if the vertices of the graph can be partitioned into two sets such that no edges of the graph have end points within the same partition class. For instance, G1 is bipartite; consider the partition of its vertices into the sets {v1, v3, v5} and {v2, v4}. On the other hand, it is impossible to partition the vertices of G2 into two such sets. As the property of being bipartite is invariant under permutations of the vertices of a graph, it follows that G1 and G2 are not isomorphic.

2) Find the Adjacency Matrix of the graphs G1 and G2.

For G1

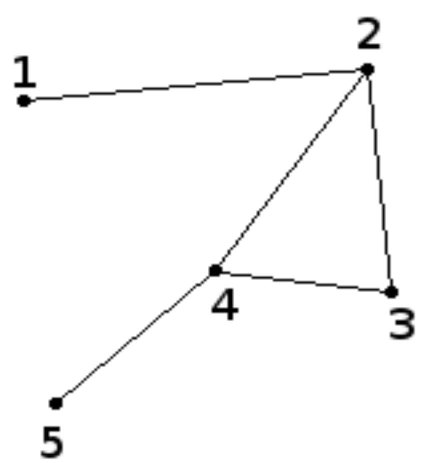
	V1	V2	V3	V4	V5
V1	0	1	0	1	0
V2	1	0	1	0	1
V3	0	1	0	1	0
V4	1	0	1	0	1
V5	0	1	0	1	0

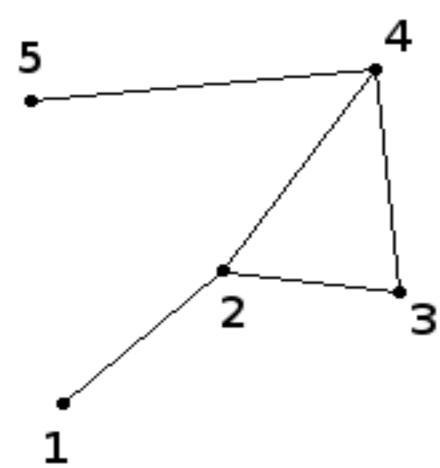
For G2

	V1	V2	V3	V4	V5
V1	0	1	1	0	1
V2	1	0	1	0	0
V3	1	1	0	1	0
V4	0	0	1	0	1
V5	1	0	0	1	0

III. Let consider the Graphs G1 and G2

G1 **G2**





1) Can we define G1 and G2 as Isomorphic graph. Why?

Two graphs are isomorphic if they are the same, except that the vertices are labelled differently. The following two graphs are isomorphic.

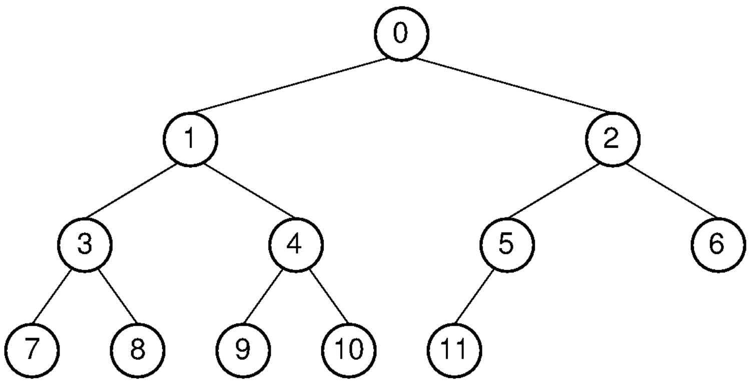
2) Find the Adjacency Matrix of the graphs G1 and G2.

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	1	0
3	0	1	0	1	0
4	0	1	1	0	1
5	0	0	0	1	0

Both graph G1 and G2 have the same matrix.

Chapter 9: Trees.

Exercise 1: Let consider this graph:



(a)

1) Is this graph a full tree or a complete tree? Explain.

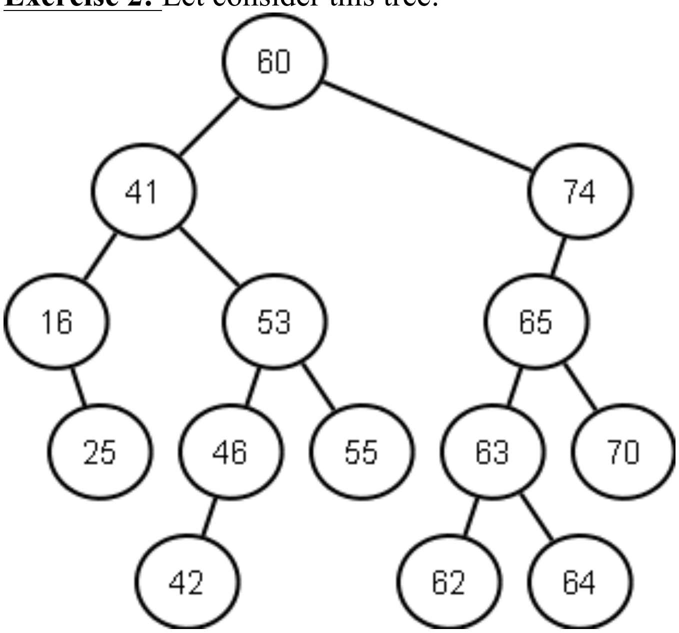
   This graph is a complete tree

2) Determine the root node, the internal node and the terminal node

A complete binary tree of 12 nodes, numbered starting from 0.

Terminal node = 6

Exercise 2: Let consider this tree:



1. Is this tree a binary search tree? Why?
   Yes, because 41 < 60 < 74, etc.

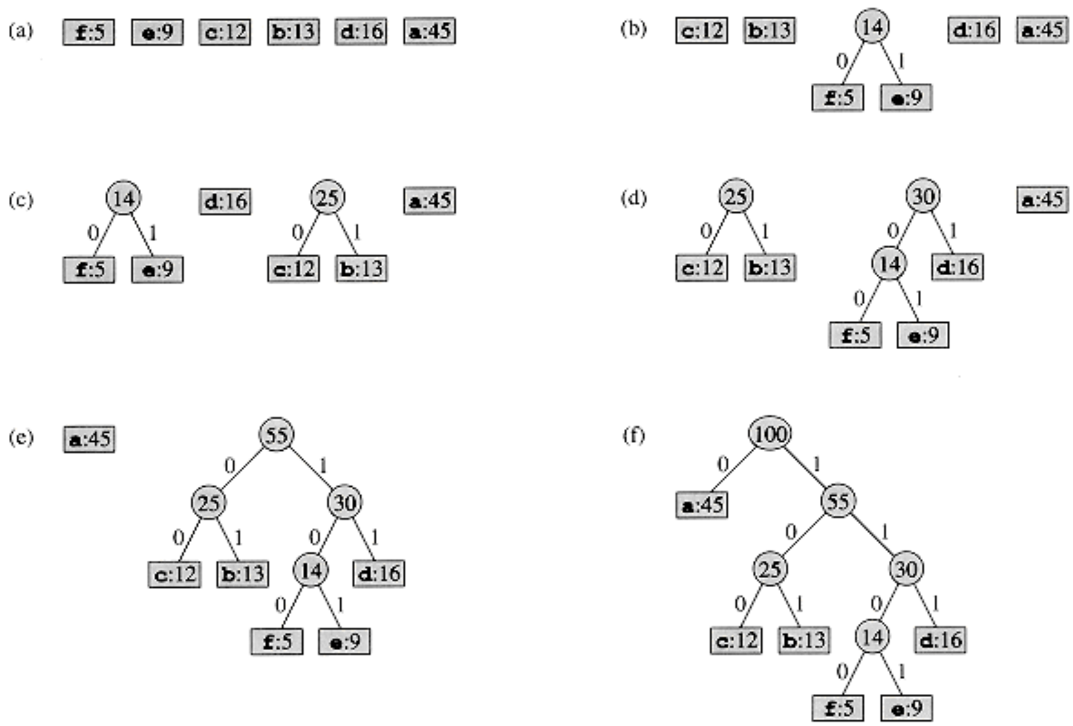
2. Define the internal node, the terminal node, the height and the root node.
   the internal node = 8
   the terminal node = 6
   the height = 4
   the root node = 60

<u>Exercise 3:</u>

Make the construction of this Huffman Code:

f:5 e:9 c:12 b:13 d:16 a:45

<u>The solution:</u>



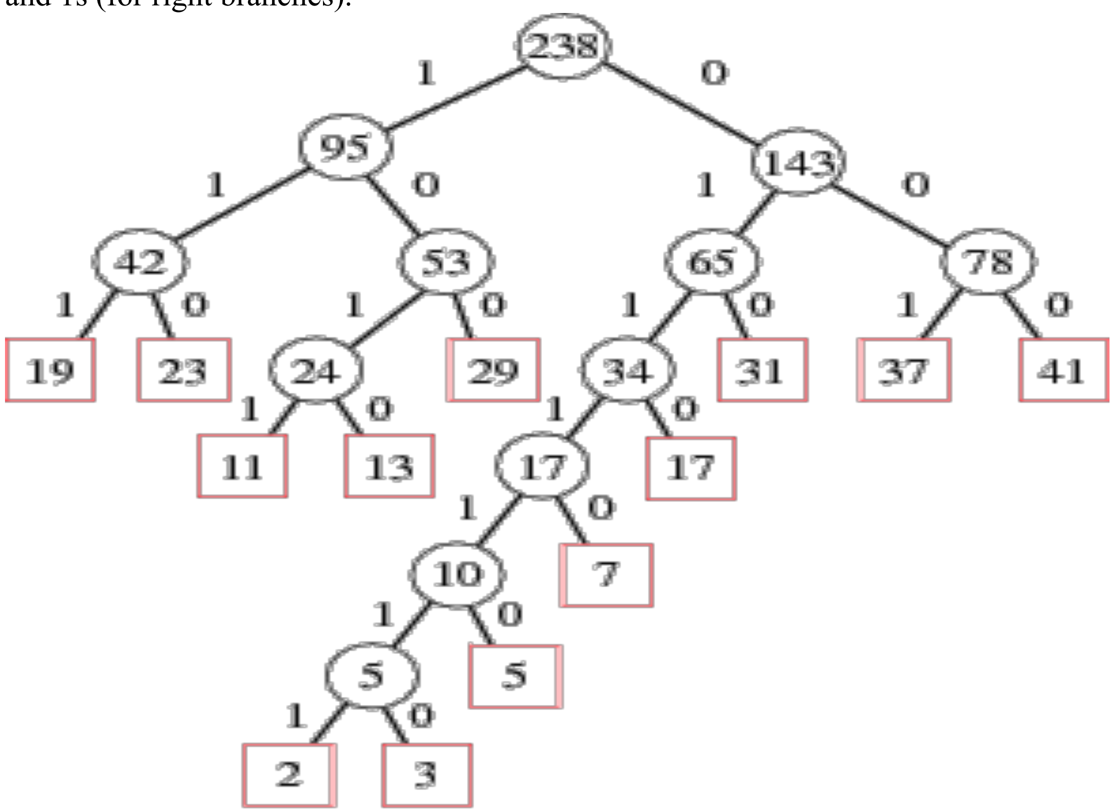
Exercise 2: Construct an optimal Huffman Code.

2	**3**	**5**	**7**	**11**	**13**	**17**	**19**	**23**	**29**	**31**	**37**	**41**

Huffman Coding

A lossless data compression algorithm which uses a small number of bits to encode common characters. Huffman coding approximates the probability for each character as a power of 1/2 to avoid complications associated with using a nonintegral number of bits to encode characters using their actual probabilities.

Huffman coding works on a list of weights $\{w_i\}$ by building an extended binary tree with minimum weighted external path length and proceeds by finding the two smallest ws, w_1 and w_2, viewed as external nodes, and replacing them with an internal node of weight $w_1 + w_2$. The procedure is them repeated stepwise until the root node is reached. An individual external node can then be encoded by a binary string of 0s (for left branches) and 1s (for right branches).



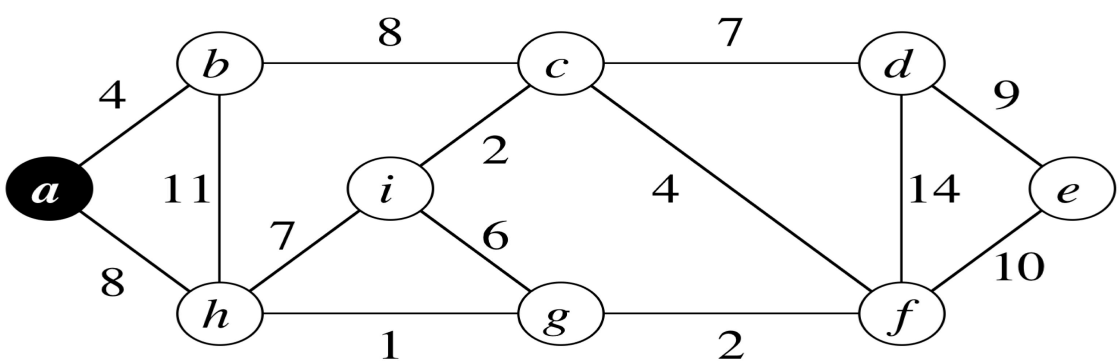
The procedure is summarized below for the weights 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, and 41 given by the first 13 primes, and the resulting tree is shown above (Knuth 1997, pp. 402-403). As is clear from the diagram, the paths to the larger weights are shorter than those to the smaller weights. In this example, the number 13 would be encoded as 1010.

2	3	5	7	11	13	17	19	23	29	31	37	41

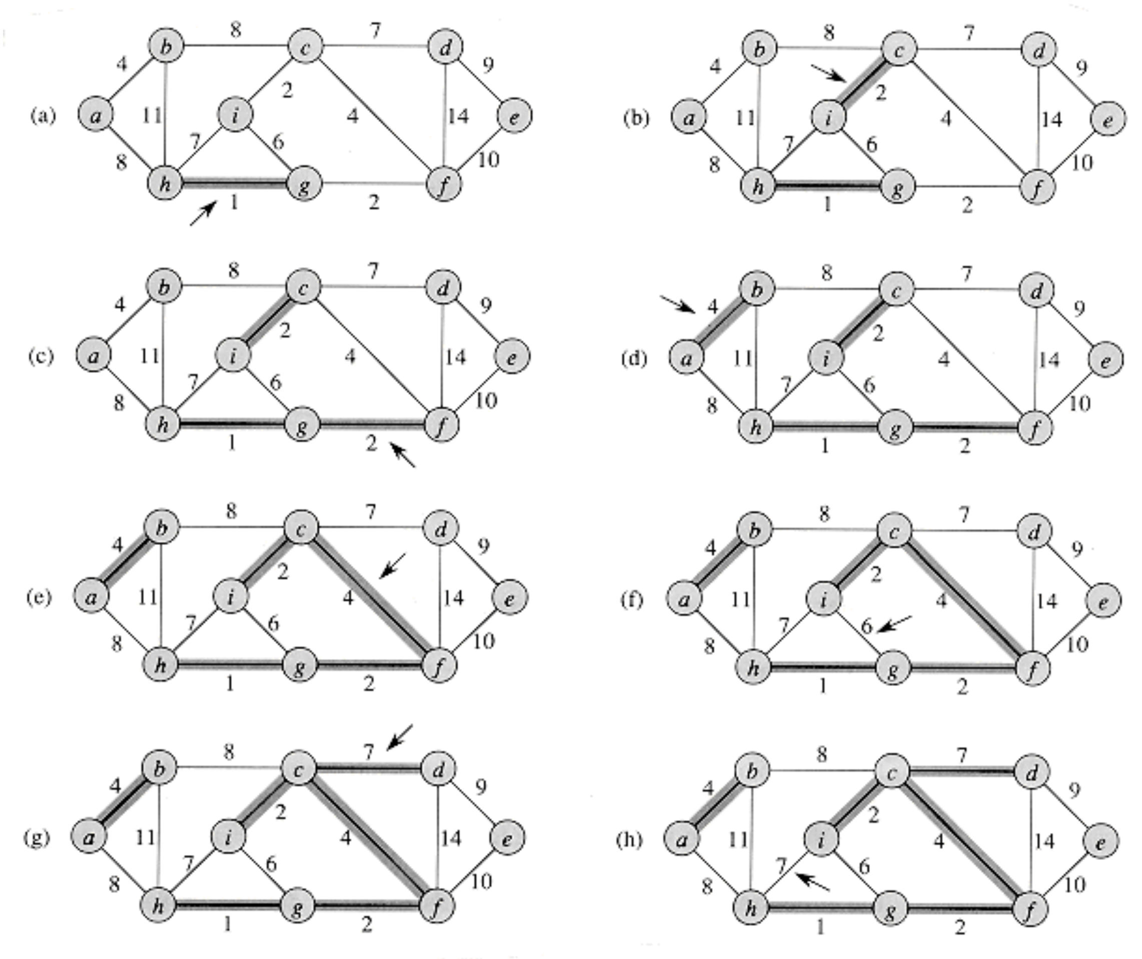
	5	5	7	11	13	17	19	23	29	31	37	41
		10	7	11	13	17	19	23	29	31	37	41
			17	11	13	17	19	23	29	31	37	41
			17		24	17	19	23	29	31	37	41
					24	34	19	23	29	31	37	41
					24	34		42	29	31	37	41
						34		42	53	31	37	41
								42	53	65	37	41
								42	53	65		78
									95	65		78
									95			143
												238

Exercise 4:

Find the Minimum Spanning Tree using Kruskal's algorithm.

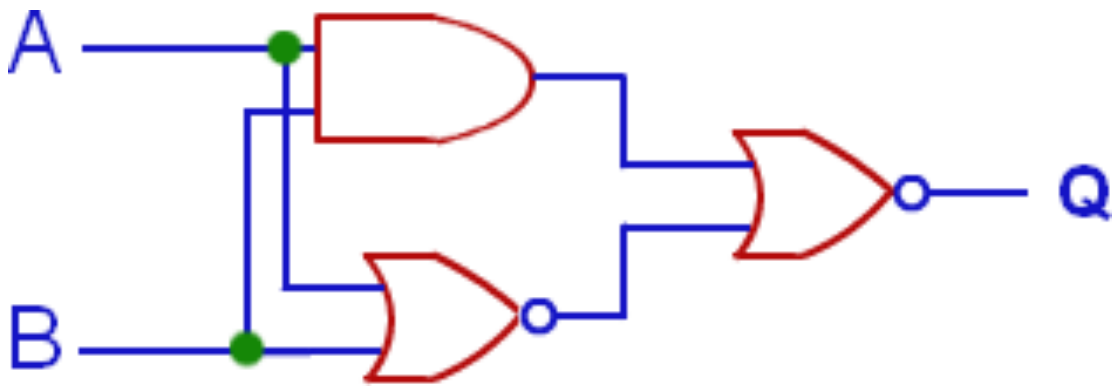


The solution:



__Chapter 10: Boolean algebra__

__Exercise 1:__ Let consider the following system.



1) Determine the final expression of the Boolean algebra.

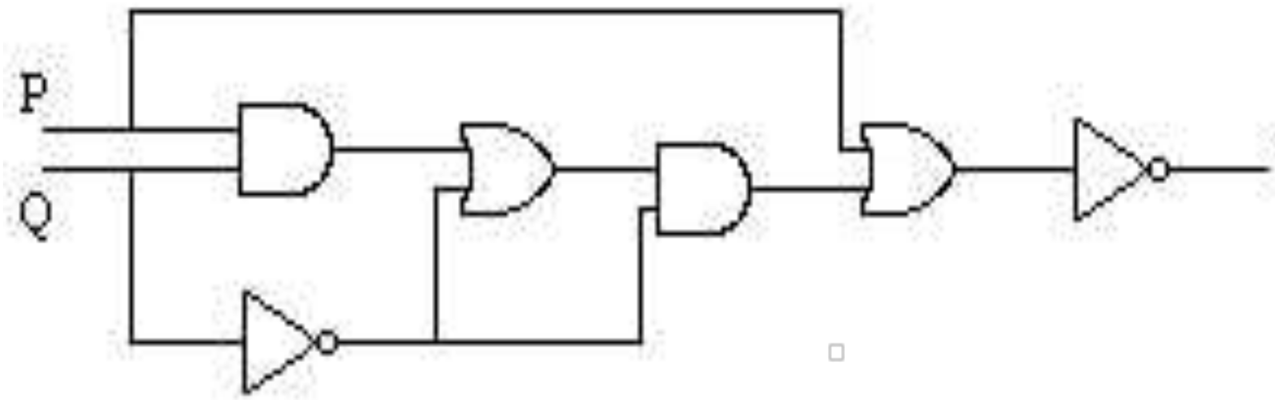
$$\mathbf{Q = (AB + (A + B)')'}$$

2) Determine a table to represent the Boolean algebra.

A	**B**	**AB**	$\mathbf{A + B}$	$\mathbf{(A + B)'}$	$\mathbf{AB + (A + B)'}$	**Q**
0	0	0	0	1	1	0
1	0	0	1	0	0	1
0	1	0	1	0	0	1
1	1	1	1	0	1	0

Exercise 2:

Let consider the following system.



1) Determine the final expression of the Boolean algebra Q.

$$\mathbf{Q} = (\boldsymbol{P} + (\boldsymbol{Q}' * (\boldsymbol{Q}' + (\boldsymbol{P} * \boldsymbol{Q})))'$$

2) Determine a table to represent the Boolean algebra Q.

P	**Q**	$\boldsymbol{Q}'$	$\boldsymbol{P} * \boldsymbol{Q}$	$\boldsymbol{Q}' + (\boldsymbol{P} * \boldsymbol{Q})$	$\boldsymbol{Q}' * (\boldsymbol{Q}' + (\boldsymbol{P} * \boldsymbol{Q})$	$\boldsymbol{P} + (\boldsymbol{Q}' * (\boldsymbol{Q}' + (\boldsymbol{P} * \boldsymbol{Q}))$	**Q**
0	0	1	0	1	1	1	0
0	1	0	0	0	0	0	1
1	0	1	0	1	1	1	0
1	1	0	1	1	0	1	1