1)

a) The height of the complete binary tree depends on number of nodes binary tree contains. To see relation between height and number of nodes, some examples must be analyzed. Let number of nodes = n and height = h.

h	n
0	0
1	1
2	3
3	7
4	15

As seen from the example, number of nodes increases by height power of 2, when height increases by one. So, their relation is exponential / logarithmic:

$$h = \log_2^{(n+1)}$$

Total depth of complete binary tree depends on sum of the each node on binary tree. If we formulate this situation,

$d(n) = \sum_{i=1}^{n} \log_2^{(i+1)}$. In this way we sum up all elements height to find total depth.

$$d(1) = \sum_{i=1}^{n} \log_2^{(i+1)} = 1 \checkmark$$

$$d(k) = \sum_{i=1}^{k} \log_2^{(i+1)} = x$$

$$d(k+1) = \sum_{i=1}^{k+1} \log_2^{(i+1)} = \underbrace{\sum_{i=1}^{k} \log_2^{(i+1)}}_{x} + \log_2^{(k+2)} = x + \log_2^{(k+2)}$$

If we assume that the formula is correct for n=k, the equation is also correct for n=k+1 because its result equals to result of d(k) + height of the (k+1)th element.

b) The number of search operation on binary search tree can be at most height of the tree. Because the tree is a complete tree the height of the tree is $\log_2^{(n+1)}$ which has found at previous question. For this reason, to find a node in complete binary search tree, minimum 1, maximum $\log_2^{(n+1)}$ operations should be performed.

$$\text{Avarage operation number} = \frac{1+\log_2^{(n+1)}}{2}$$

c) Yes, there is a restriction on the number of nodes in a full binary tree. The number of nodes in a full binary tree must be odd because every node has two or no childeren and there is a root node.
Let we have a full binary tree with n number of nodes.

$$n = \underbrace{1}_{\text{root node}} + \underbrace{0.(x)}_{\text{x nodes with 0 child}} + \underbrace{2.(y)}_{\text{y nodes with 2 child}}$$

Due to fact that the multiplication of one even number and one odd number or two even number is even, $2y$ is an even number. So, $2y+1$ is odd and n is odd.

Number of leaves is always one more than internal nodes. Because when a full binary tree is created there is one leaf and no internal nodes, if two childen is added to first node, first node becomes an internal node. So, number of internal nodes increased by one when number of leaves increased by one. The equation is;

number of leaves $= \frac{n+1}{2}$, number of internal nodes $= \frac{n-1}{2}$