# CSE222 HW5

## Q1

a) At every depth $h$ there exists $2^{h-1}$ nodes. Therefore, total depth of nodes at the same depth is $h2^{h-1}$. To find the total depth of the complete binary tree we must sum this for every depth. Therefore, total depth of a complete binary tree is:

$$
\begin{aligned}
f(h) &= \sum_{i=1}^{h} i2^{i-1} \\
&= 1 \cdot 2^0 + 2 \cdot 2^1 + \cdots + \mathrm{h}2^{\mathrm{h}-1} \\
&= 1 \cdot 2^0 + 2\left(2 \cdot 2^0 + \cdots + h2^{h-2}\right) \\
&= 1 \cdot 2^0 + 2\left[\left(1 \cdot 2^0 + \cdots + (h-1)2^{h-2}\right) + \left(2^0 + \cdots + 2^{h-2}\right)\right] \\
&= 1 \cdot 2^0 + 2[f(h-1) + 2^{h-1} - 1] \\
&= 1 + 2(f(h-1) + 2^{h-1} - 1) \\
&= 2f(h-1) + 2^h - 1
\end{aligned}
$$

With $f(0) = 0$ and $f(1) = 1$ the recurrence relation is solved as:

$$f(h) = 2^h(h-1) + 1$$

b) With a complete binary tree length of $h$ half of its nodes are at the bottom. So, when a random value that is inside the tree is given its probability of being at the bottom is $\frac{1}{2}$ while its probability of being at depth $h-1$ is $\frac{1}{2}\left(1-\frac{1}{2}\right) = \frac{1}{4}$. For a randomly selected node value in a complete tree with depth $h$ its probability of being at depth $d$ is $\frac{1}{2^{h-d+1}}$. If it's found at depth $d$ then that many comparisons have been made. So, the average number of comparisons for successful search is:

$$
\begin{aligned}
f(h) &= \sum_{i=1}^{h} \frac{i}{2^{h-i+1}} \\
&= \frac{1}{2^{\mathrm{h}}} \sum_{i=1}^{h} i2^{i-1} \\
&= \frac{1}{2^h}[2^h(h-1) + 1] \\
&= h - 1 + \frac{1}{2^h}
\end{aligned}
$$

We know that for a complete binary tree its depth is $log_2 n$, assuming that it has $n$ elements. So, for number of elements the equation becomes:

$$
\begin{aligned}
f(n) &= log_2 n - 1 + \frac{1}{n} \\
&= \Theta(log\, n)
\end{aligned}
$$

c) If a leaf node is turned into an inner node with two leaf nodes the total change in number of leaf nodes is plus one. We know that for a full tree with one inner node there is two leaf nodes. Now assume $f(n) = n + 1$ to be the function that gives number of leaf nodes for $n$ many inner nodes. Assume that the function holds for $n = k$. Now change one of the leaf nodes to an inner node, making $n = k + 1$, function becomes:

$$
\begin{aligned}
f(k+1) &= (k+1) + 1 \\
&= f(k) + 1
\end{aligned}
$$

The function holds the one increment of number of leaf nodes upon one increment of number of inner nodes, therefore it is proven that for $n$ $\mathrm{inner\ nodes\ there\ must\ be}$ $n+1$ leaf nodes making that number of total nodes has to be $2n + 1$. So, there can't be a full binary tree with even number of nodes. For a full tree with n elements there are $\frac{n-1}{2}$ many inner nodes and $\frac{n+1}{2}$ many nodes.

## Q2