Exercise 1

Exercise 2
Consider a simple open addressing scheme, let’s say linear probing with a hash code $f'(x)$. We start with an array of size $n$, use hash function $f_n(x) = f'(x) \mod n$. When this array gets full we move to an array of size $2n$ with hash function $f_{2n}(x) = f'(x) \mod n$. When this gets full we again double the size and so on. Clearly a single insert could take a long time if rehashing is to be done. Show that the amortized insert time is $\Theta(1)$.

Exercise 3
We are given a skip list $S$ with promotion parameter $p$ (the probability with which we promote elements). On this we define the finger search operation which is a generalization of the normal find operation. We are given a direct link to a node containing key $x \in S$ and we are asked to find a $y > x$. If $x$ is the $i$th element and $y$ is the $j$th element of the base list of $S$ (where $j > i$), explain how to implement finger search so that its expected running time is $O(\log_2(2 + j - i))$. If $y \notin S$ you may assume that $y_-$ is the $j$th element of the base list of $S$.

Exercise 4
Recall that we define the height of tree as the maximum depth of any node of the tree (root has depth 0). We are given an AVL tree of height $h$. Remove all nodes of depth exactly $h$ from this tree. Now prove that the remaining tree is also an AVL tree.