2-AVL Trees

Let a 2-AVL tree be a binary search tree where for every node, the difference of heights of its left and right subtree is at most 2. Prove that a 2-AVL tree has height at most $3 \log n$ where $n$ is the number of nodes in the tree.

Scapegoat Tree Insertion

Insert the element 13 in the following scapegoat (2/3)-tree. The numbers in the brackets are the number of nodes in that subtree.
**Perfectly Balanced BST**

Construct a perfectly balanced binary search tree out of an array $A$ of $n$ sorted elements in linear time.

**Skip Lists**

Suppose you have a skip list with only three levels. The top level contains only the sentinels. The lowest level has $n + 2$ keys: $-\infty, a_0, ..., a_{n-1}, \infty$, while the middle level contains $k + 2$ keys including the sentinels. Assume $k$ divides $n$. Suppose that the $k$ entries are evenly spread out and the first entry corresponds to $a_0$.

- What is the worst case time for a query? Give a tight bound involving $k$ and $n$.
- Given $n$, how should you choose $k$ to minimise the worst case, and what does the worst case become in that case?