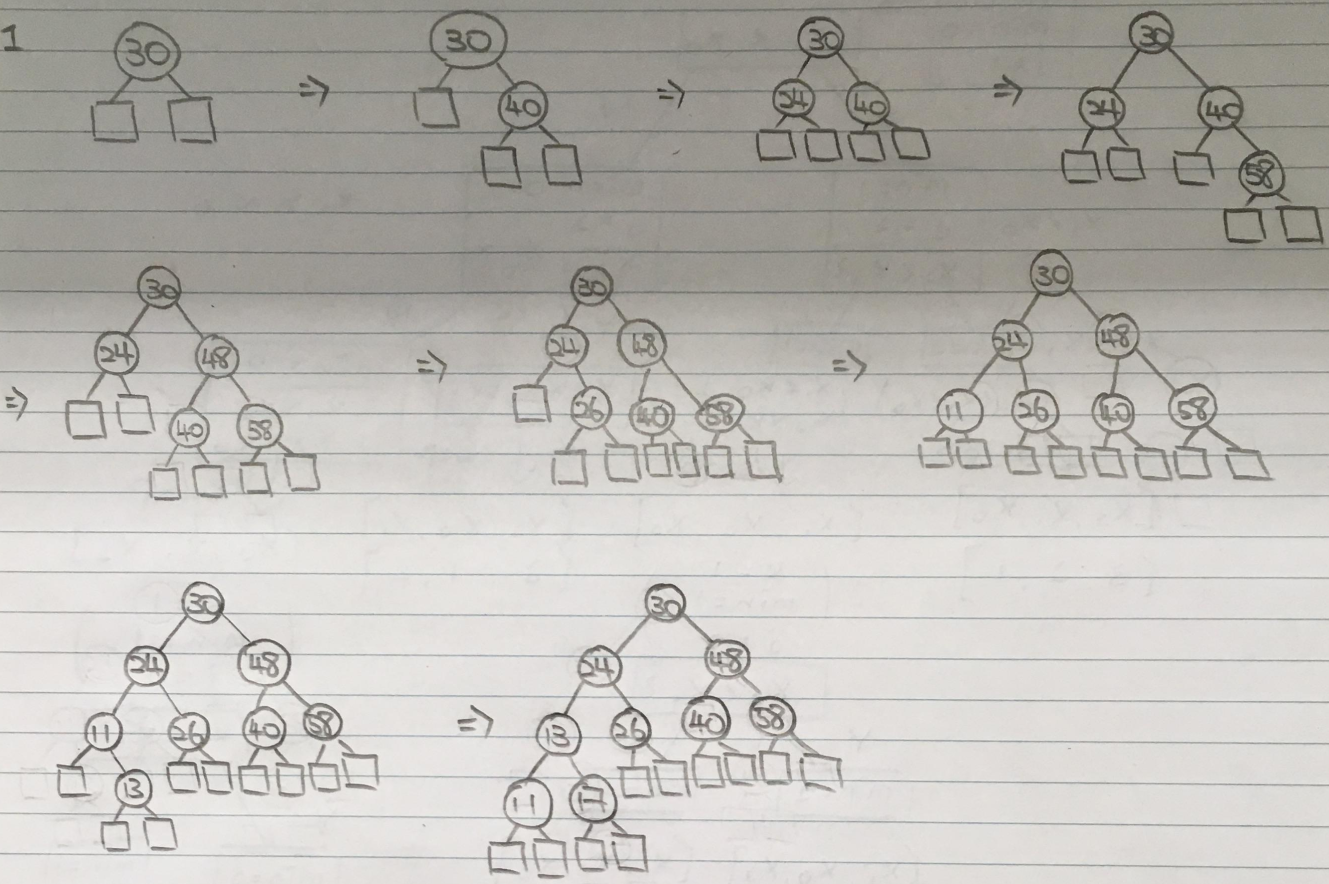
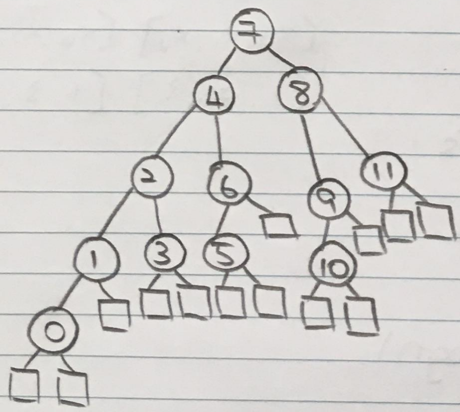
1



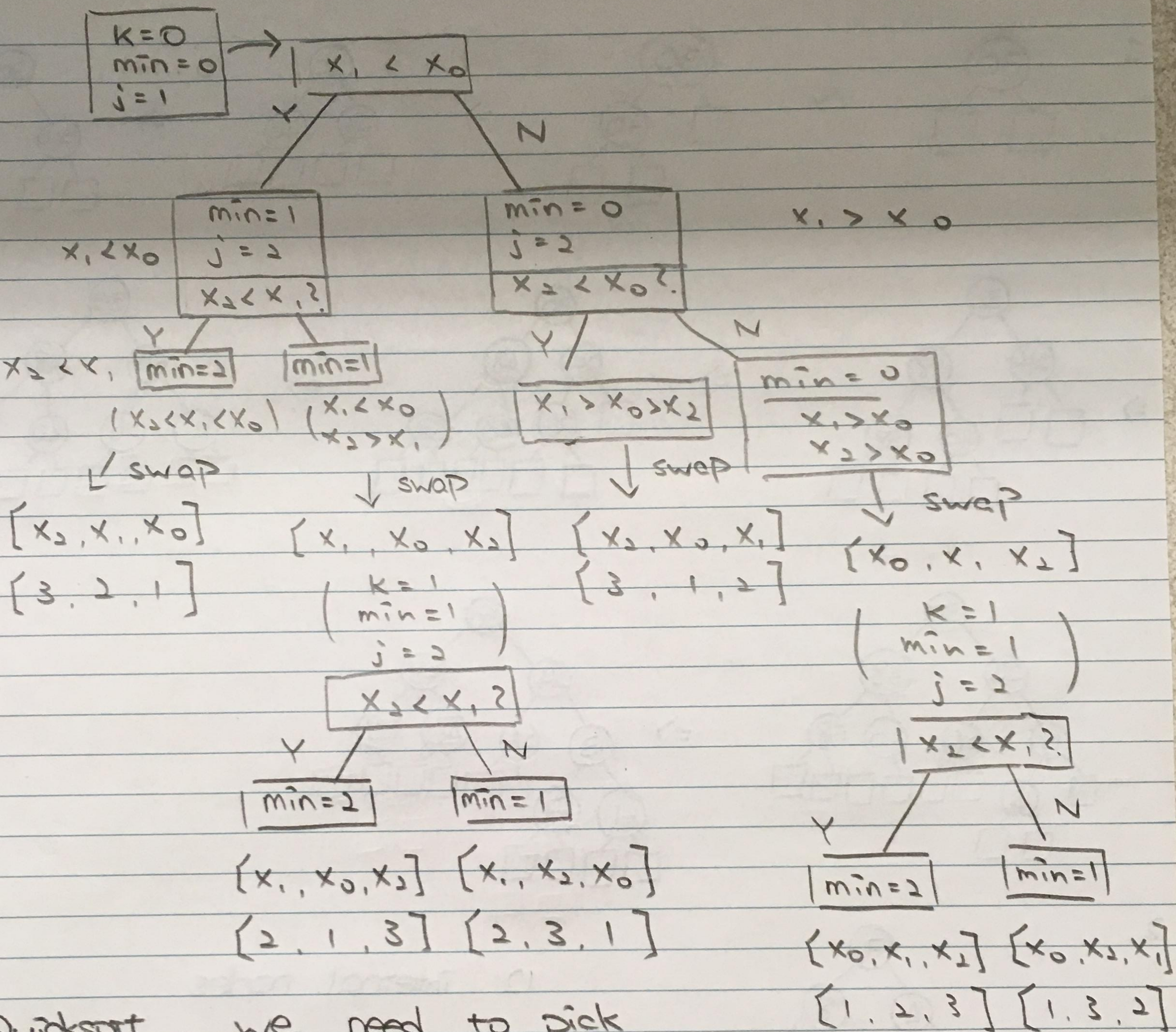
2



12 internal nodes

3 Suppose $f(n) \in \Theta(g(n))$. Then by the definition of Big-Theta notation, $f(n) \in \Theta g(n)$ if and only if there exists $c_1, c_2 > 0$ and $n_0 > 0$ such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$. Hence, by the definitions of Big-Oh and Big-Omega notations, $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$. Therefore, $f(n) \in \Theta(g(n))$ iif $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$.

4



5 In Quicksort, we need to pick pivots n times, and the height is $\log_{4/3} n$. Hence, the running time is $O(n \log_{4/3} n)$. However, there is 50% chance we pick a good pivot and 50% chance the internal disk fault. That makes $4n \log_{4/3} n$, which is $O(n \log n)$.

Input size	QuickSelect	LinearSelect
25	274568 ns	28839 ns
125	254814 ns	90074 ns
625	550715 ns	471703 ns
3125	4096779 ns	849383 ns
15625	1233382 ns	3294023 ns
78125	2650073 ns	9060737 ns
390625	6016788 ns	19263202 ns

According to the observations, LinearSelect is not much faster than QuickSelect when the input size is small. However, the running time of LinearSelect can be faster when the array size is large. After all, the worst-case of LinearSelect is $O(n)$. In contrast, the worst-case of QuickSelect is $O(n^2)$