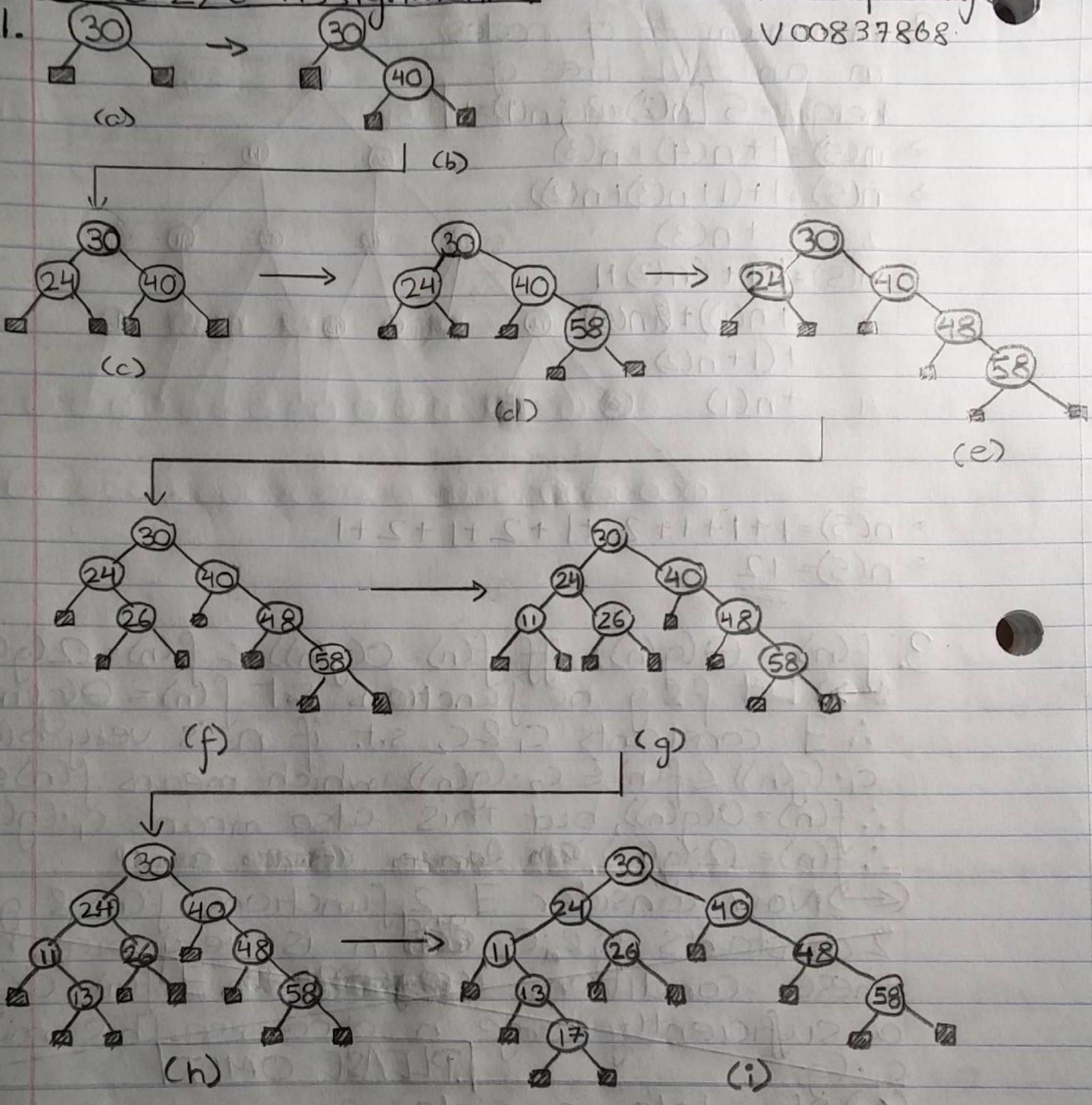
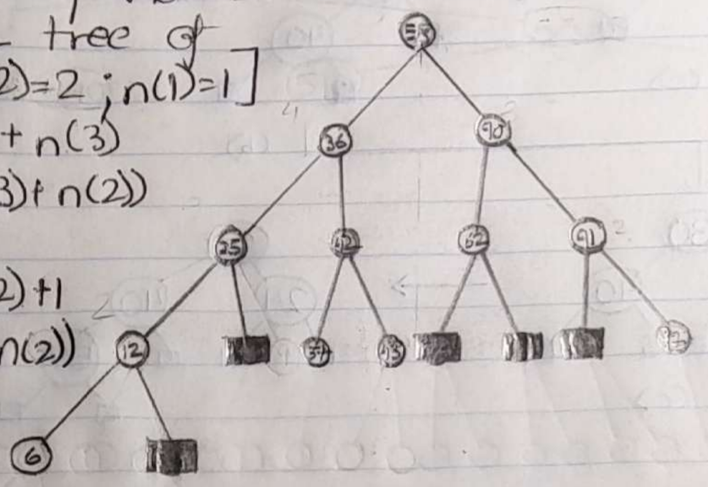
22/09/16 CSC 226 Assignment 1

Devroop Banerjee
V00837868

1.



2. minimum # of nodes in an AVL tree of height = 5 [n(2) = 2 ; n(1) = 1]



$\Rightarrow n(5) = 1 + n(4) + n(3)$

$\Rightarrow n(5) = 1 + (1 + n(3) + n(2)) + n(3)$

$\Rightarrow n(5) = 1 + (1 + (n(2) + 1 + n(1)) + n(2)) + (1 + n(2) + n(1))$

$\Rightarrow n(5) = 1 + 1 + 1 + 2 + 1 + 2 + 1 + 2 + 1$

$\Rightarrow n(5) = \underline{12}$

3. $f(n) = \Theta(g(n))$ iff $f(n) = O(g(n))$ & $f(n) = \Omega(g(n))$

($\rightarrow$) Let f & g be functions s.t. $f(n) = \Theta(g(n))$.

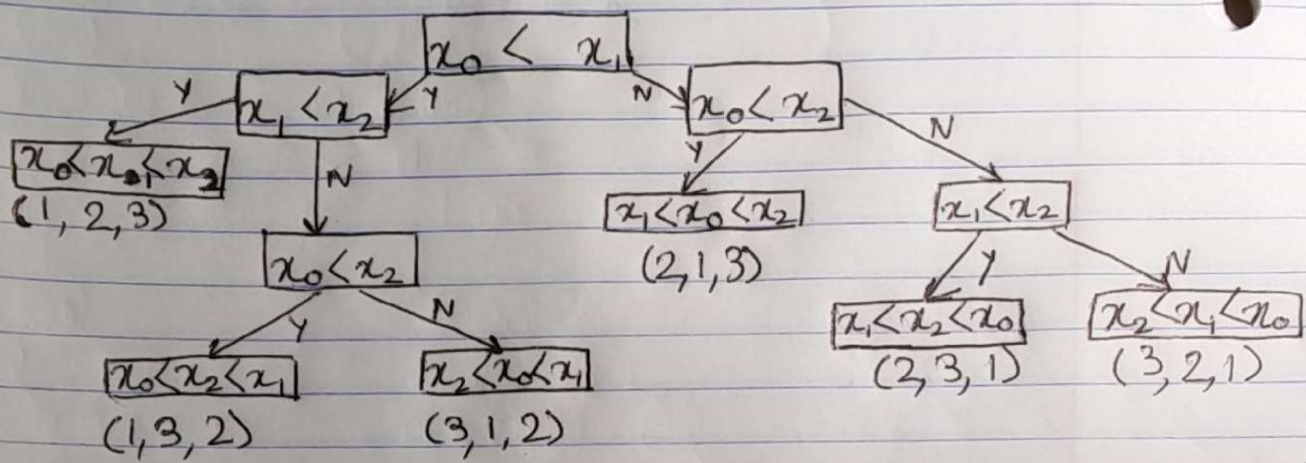
$\therefore \exists$ constants c_1 & c_2 s.t. if n is very big, then $c_1 \cdot (g(n)) \leq f(n) \leq c_2 \cdot (g(n))$ which means $f(n) \leq c_2 \cdot (g(n))$

$\therefore f(n) = O(g(n))$, but this also means $c_1 \cdot (g(n)) \leq f(n)$

$\therefore f(n) = \Omega(g(n))$.

(⇐) Now let's consider a constant k_1 and 2 functions $f(n)$ & $g(n)$. If $f(n) = \Omega(g(n))$ then $k_1 \cdot (g(n)) \leq f(n)$, for a sufficiently large n. If another constant, k_2, was introduced and the same functions were used to show $f(n) = O(g(n))$ then $f(n) \leq k_2 \cdot (g(n))$. Considering all the functions and constants, $k_1 \cdot (g(n)) \leq f(n) \leq k_1 \cdot (g(n))$, which by definition is $f(n) = \Theta(g(n))$.

4.



5. Since Rustbucket has an independent 50-50 chance of an internal disk fault, a randomized quicksort seems like Rustbucket's best sorting algorithm.

- Take Rustbucket's n music files
- Select a pivot from $n/4 - 3n/4$
- These pivots would sort the files into lesser thans, greater thans & equals (based on 0, 1, and -1).
- Keep sorting the lesser thans & greater thans recursively till all the files are sorted.
- Voilà.

Let random variable x_{ij} compare ith smallest & jth largest files.
Let x denote the total # of comparisions.

$x = \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_{ij}$ $\quad f_i$ = smallest file ; f_j = largest file.
Pivot is between f_i or f_j

$\Rightarrow E[x] = \sum_{i=1}^{n} \sum_{j=i+1}^{n} E[x_{ij}]$ Under various conditions & circumstances, f_i & f_j might end up in the same block or in different block (lesser, equal, or greater). Either way the probability that the pivot is -1 or not at each step is $\frac{2}{(j-i+1)}$.

$E[x] = \sum_{i=1}^{n} 2\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n-i+1}\right)$ $\quad [1 + 1/2 + 1/3 + \dots = H_n]$

$\Rightarrow E[x] < 2n(H_n - 1) \leq 2n \ln n$ $\quad [H_n = n^{th}$ Harmonic #$]$
$\Rightarrow E[x] = O(2n \ln n)$ ie:- $\underline{E[x] = O(n \log n)}$