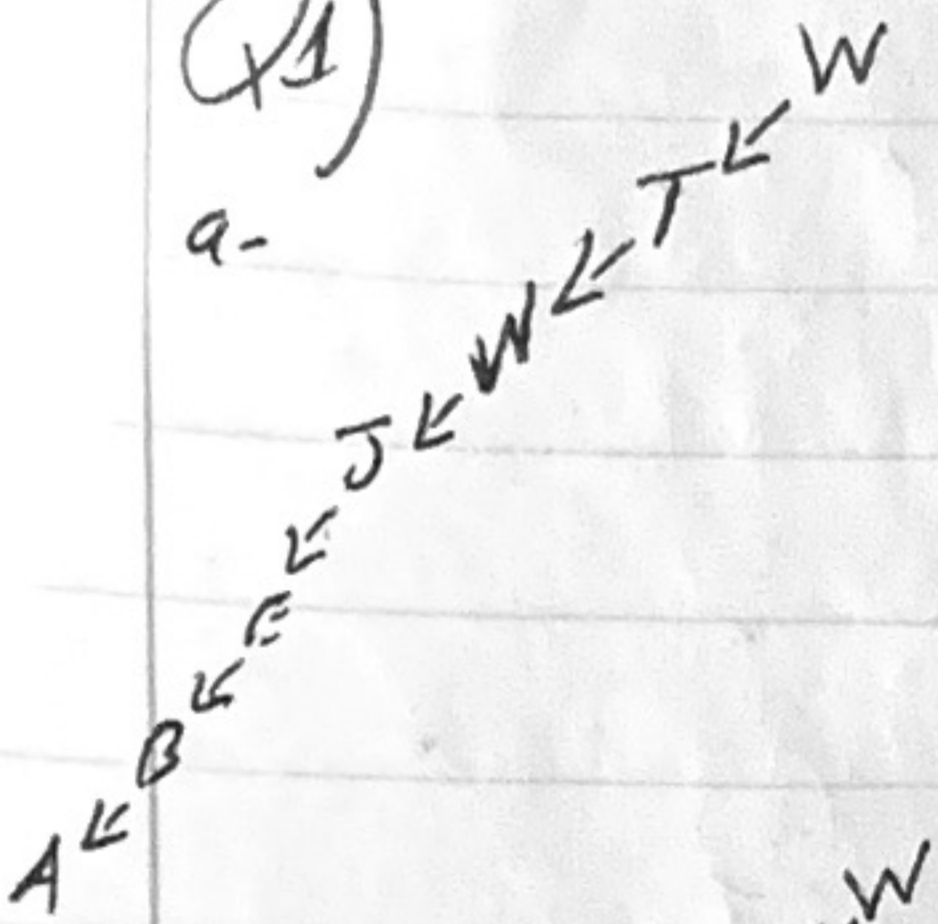
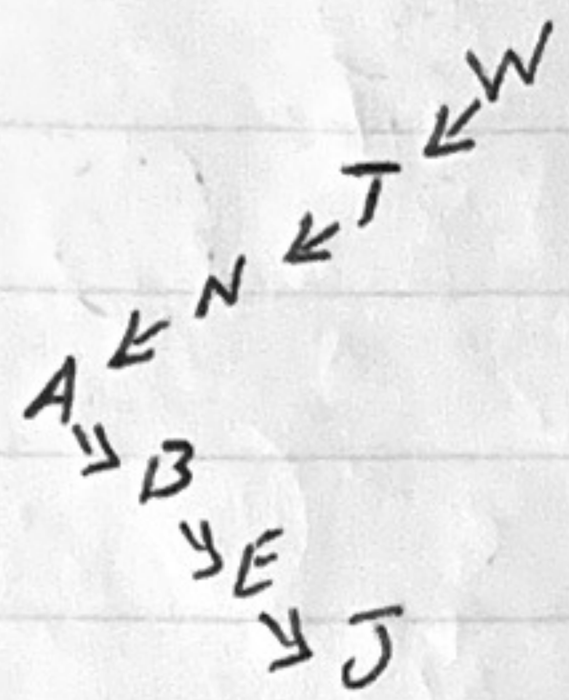
Q1)

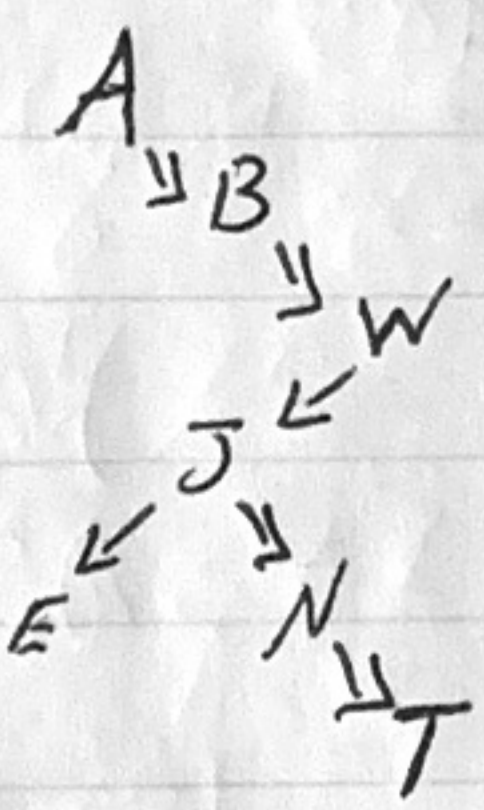
a-



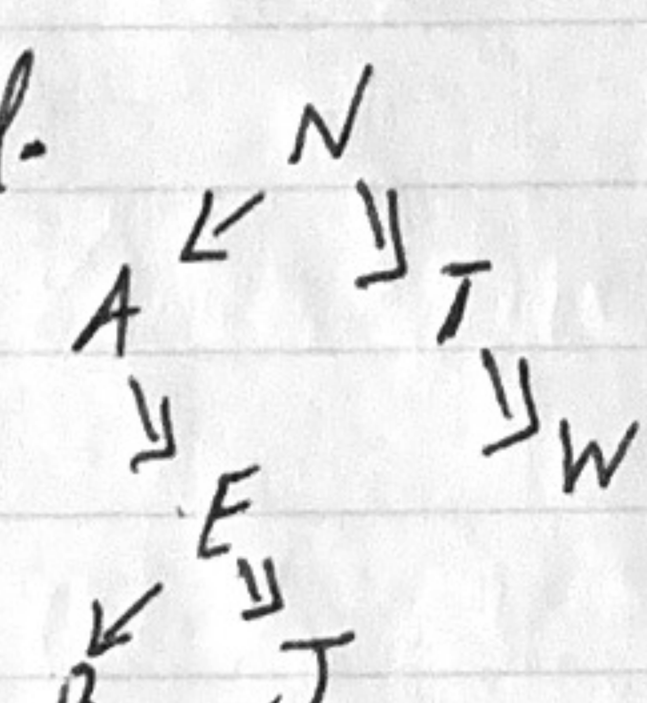
b-



c-

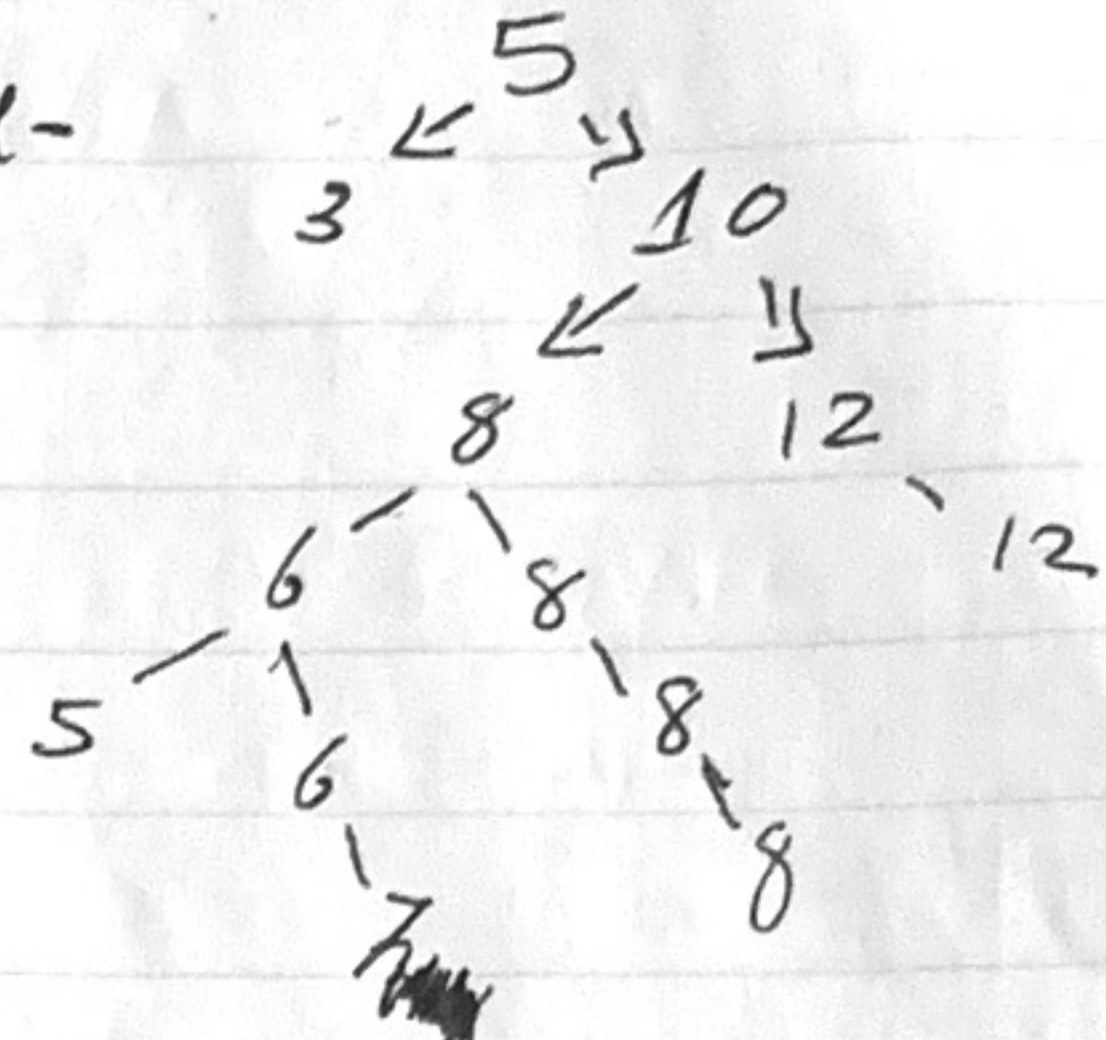


d-



Q2)

a-



b- After n insertions, the height of the three will $n-1$. Each insertion will require $n-1$ operations. $\boxed{O(n^2)}$

c- This will only take linear time since the tree is at height ~~of table~~ of 0. Inserting keys is $O(1)$. & inserting n key = $\boxed{O(n)}$.

d- Expected running time: when we are randomly choosing we are picking left half of the time so the depth will be $\boxed{O(n \log n)}$

3) $E\left[[x+s]\right] = E[X] + E[Y]$

$E\left[\sum_{i=1}^{n} (x_i)\right]$ x_i = customer to get their hat back

$E(x_i) = 1/n$

$E([x_1 + x_2 + x_3 + \cdots + x_n]$ $E([x_i]) = 1/n$

$E(x_1) + E(x_2) + \cdots + E(x_n)$

$\cancel{\sum_{n=1}^{n}}$ $\sum_{i=1}^{n} 1/n$

$\boxed{= 1}$

4) $T(n) = T(c) + ncn - \frac{c.(n-1)}{2}$

a- $= n^2 - \frac{n(n-1)}{2}$

$= \frac{n^2}{2} + \frac{n}{2}$

$= O(n^2)$

b- M= [ ] (empty list), L= empty list, R= empty list

if a< A[2] then

L append(a)

elif a> A[2]

R append(a)

else

M append(a)

4b) q = Random(p,r) // generate a random number in the range of [p,r]

L = empty list , M = empty list , R = empty list

for each element a in A except A[q] do

if a < A[q] then

L.append(a)

else if a > A[q] then

R.append(a)

else

M.append(a)

A = append(L, A[q], M, R);

return A, q, q + len(M);

c) QS(A, p, q - 1)

QS(A, t+1, r)

if p < r

q, t = Partition Prime (A, p, r)

QS_Prime (A, p, q - 1)

QS_Prime (A, t + 1, r)

return A