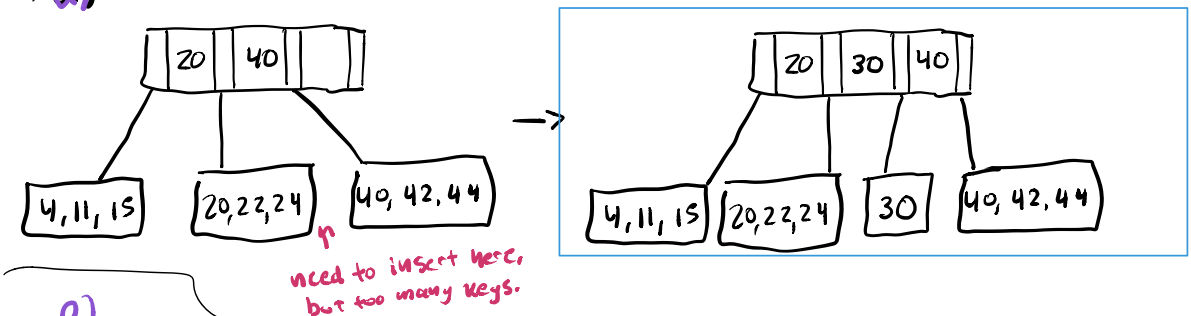
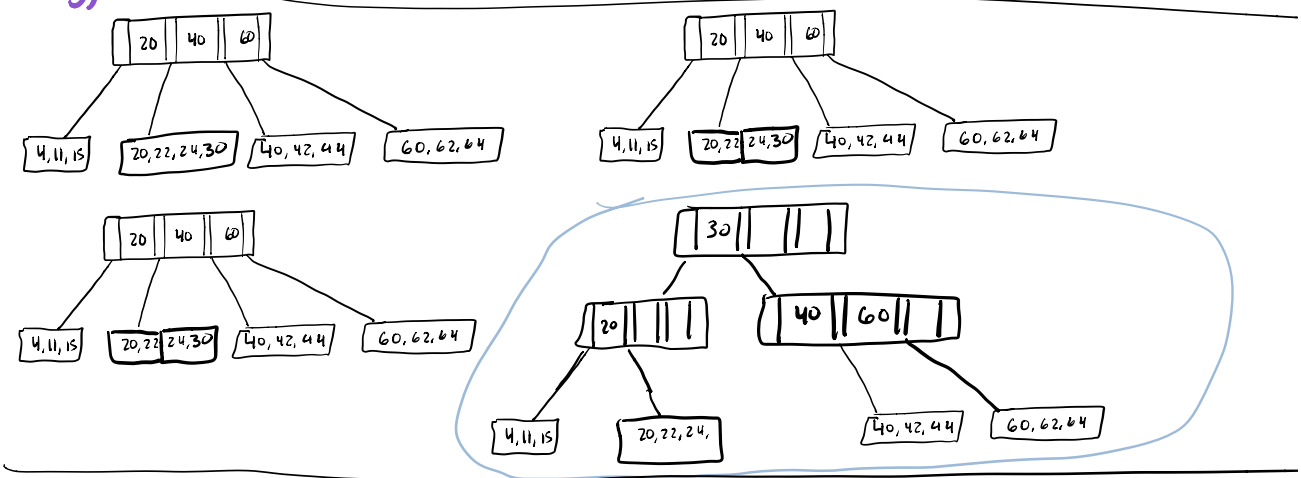
Nolan Anderson CS317 Fall20 Final Exam, Allen.

1.) a)



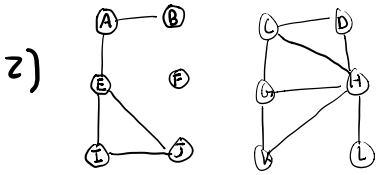
B)



C) Which space/time trade off technique...

B-trees use prestructuring to use extra space but index faster.

2)

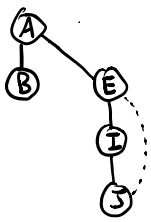


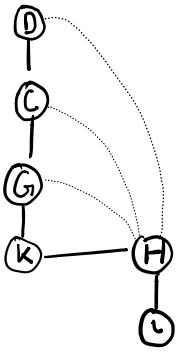
e 3, 4
b 2, 1
a 1, 5

J 5, 2
i 4, 3

k 9, 8
G 8, 9
C 7, 10
D 6, 11

F 12, 12
L 11, 6
H 10, 7





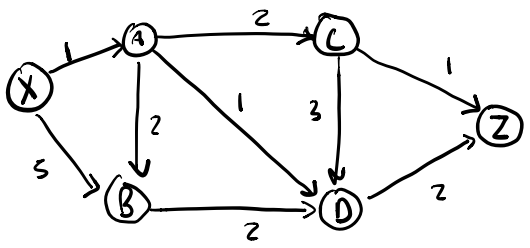
F

(3) $T(n) = 9T\left(\frac{n}{3}\right) + 3n^2 + 8n + 1$

$T(n) = 9T\left(\frac{n}{3}\right) + n(3n + 8) + 1$

$T(n) = 9\left(9T\left(\frac{n}{3}\right)\right) +$ not sure.... didn't study enough!

(4)



From X to A : X - a : 1

From X to B : X - a - b : 3

From X to C : X - a - c : 3

From X to D : X - a - D : 2

From X to Z : X - a - D - Z : 4

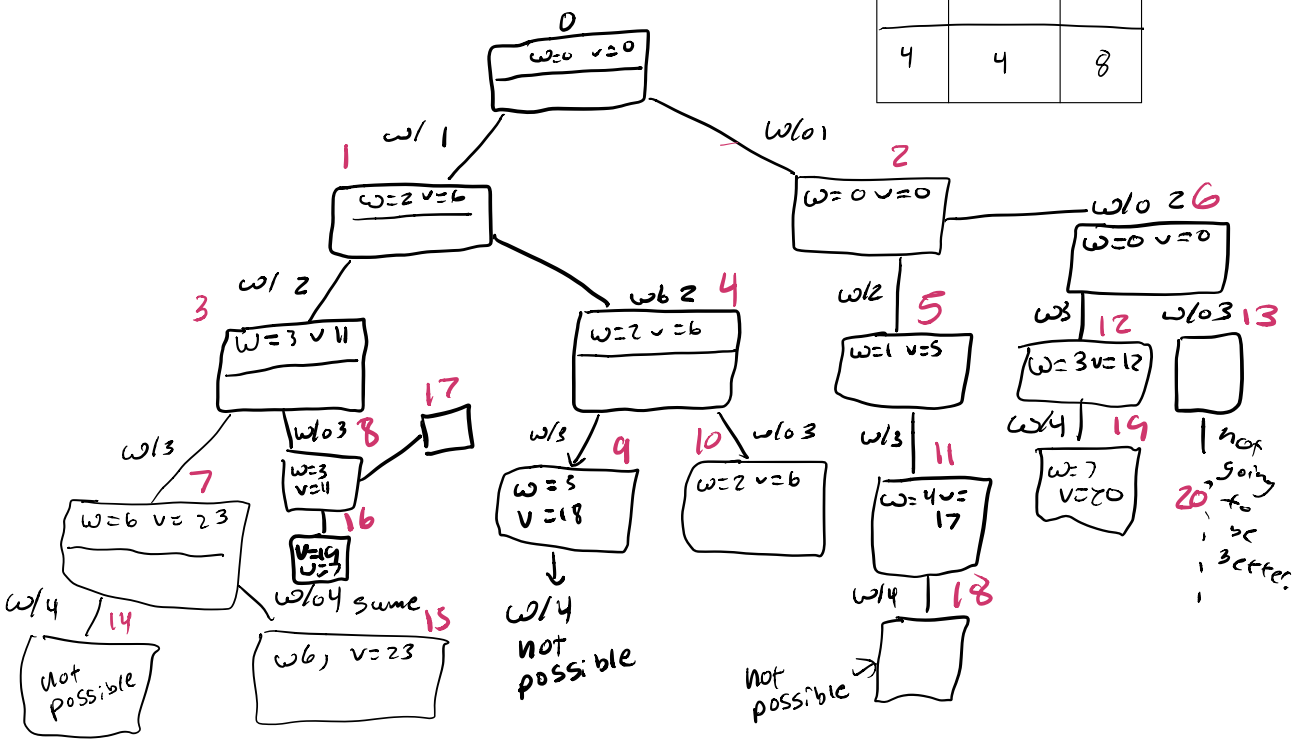
Vertices	Remaining vertices
X(-, 0)	A(X, 1) B(X, 5) C(-, ∞) D(-, ∞) Z(-, ∞)
a(X, 1)	B(a, 1+2) D(a, 1+1) C(a, 1+2) Z(-, ∞)
D(a, 2)	Z(D, 2)

3.) 5, capacity is 7

The optimal solution is items 1, 2 and 3. Weight = 6

V = 23.

Item	weight	value
1	2	6
2	1	5
3	3	12
4	4	8



Sorry for the mess... hard to keep it clean.

6)

a) For a sorted array, simply compute the difference of first and last element. O(1)

b) For unsorted linked list, you would need to traverse the whole array O(n) to find min and max. So O(n)

c) Smallest, leftmost node -> follow left until left is null.
Largest right most node -> follow right until right node is Null.
↳ | -> n nodes long, running time is O(n)

7.)

a) I would say this is dynamic programming as you are throwing out 2/3's of places the item could be.

b) T(n) = running out of time...

c)

d)

BONUS:

NP contains the knapsack problem, graph coloring, and Hamiltonian Circuits, but there are certainty more.