Drew Pulliam – DTP180003

CS3345.0U1

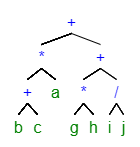
Assignment 4

1. A. a c d e n w q r s

B. n e d q f g h k r s y t z

C. d q e g h k f y s z t r n

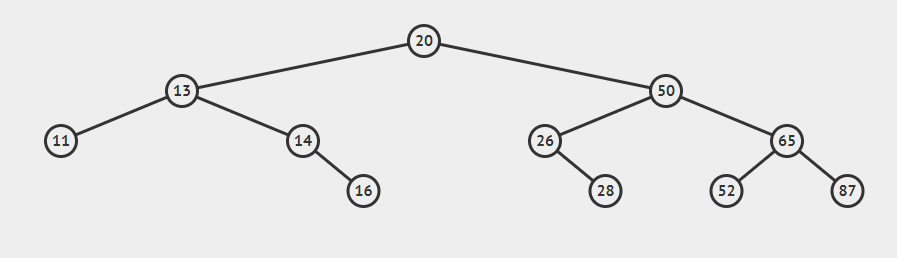
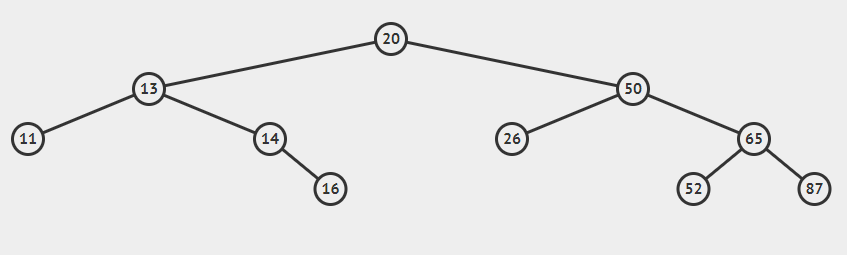
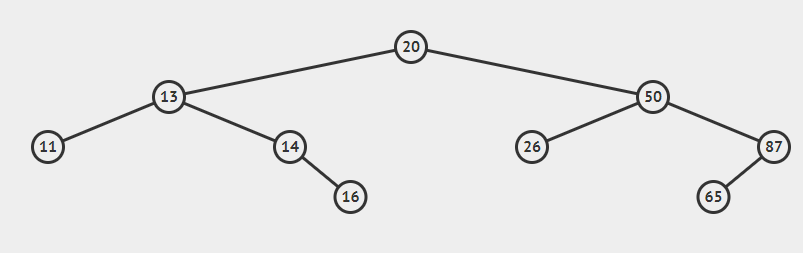
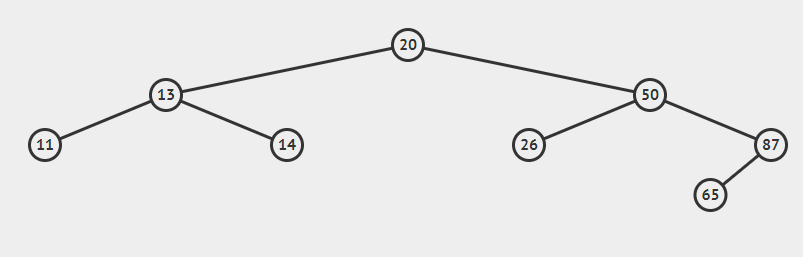
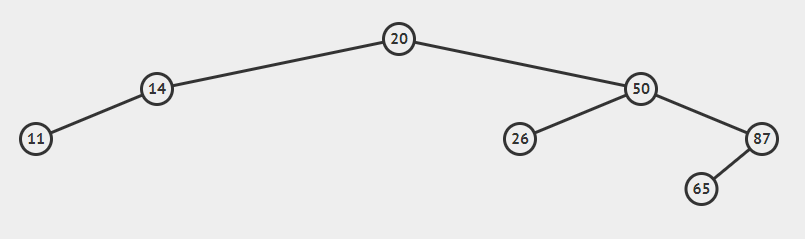
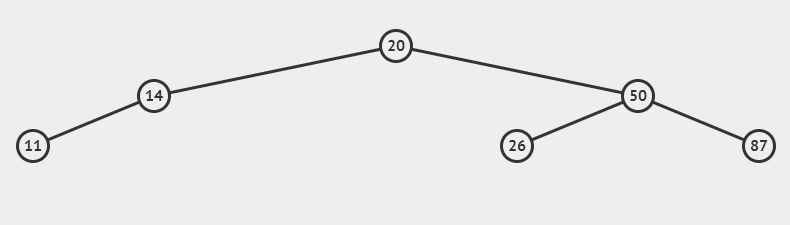
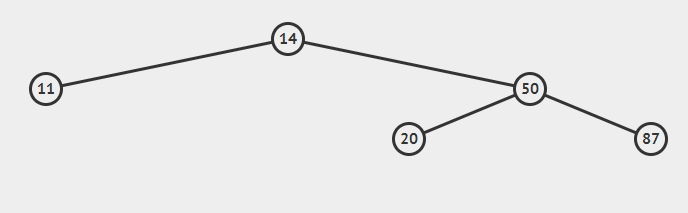
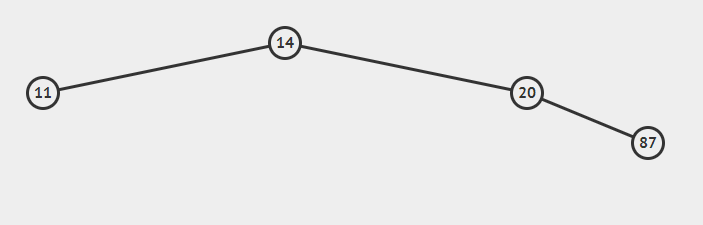
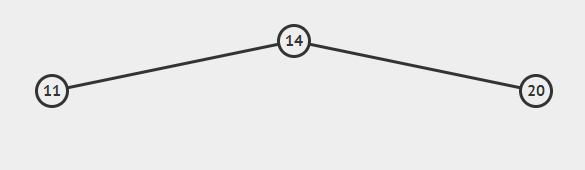
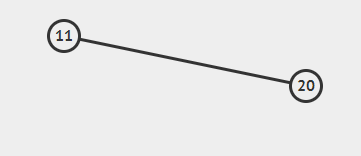
------------------------------------------------------------------------------------------------------------------------------------------

1. abc+\*gh\*ij/++ 

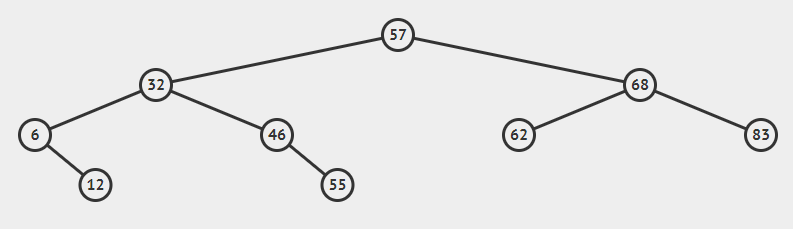
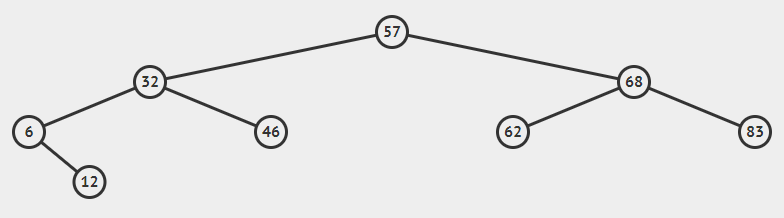
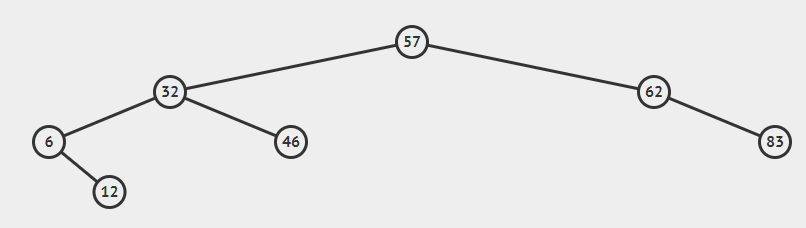
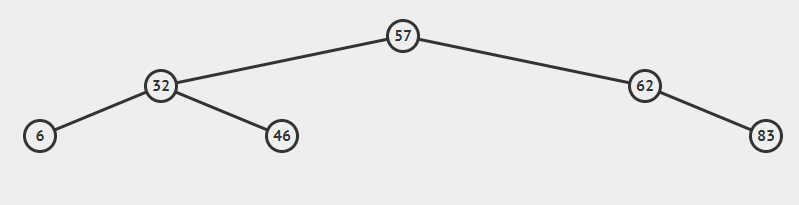
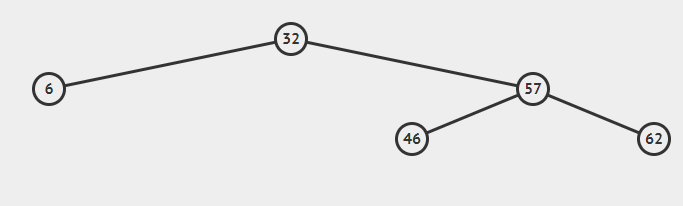
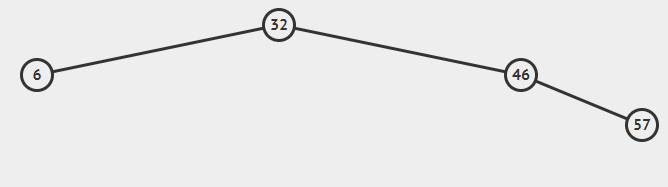
------------------------------------------------------------------------------------------------------------------------------------------

1. +\*a+bc+\*gh/ij

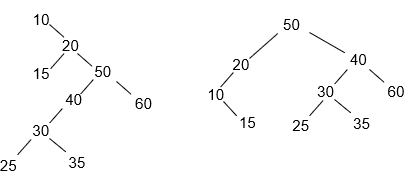
------------------------------------------------------------------------------------------------------------------------------------------

1. I wasn’t sure what the best way to show all rotations of the tree was, so I decided to show screenshots of the tree after adding each element (which shows the rotations as well).

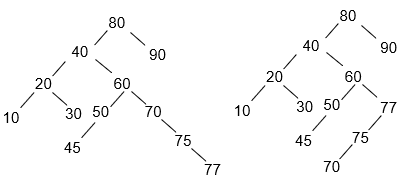
------------------------------------------------------------------------------------------------------------------------------------------

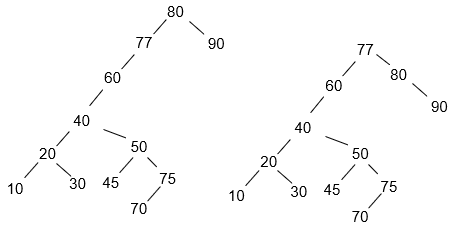
1. 

------------------------------------------------------------------------------------------------------------------------------------------

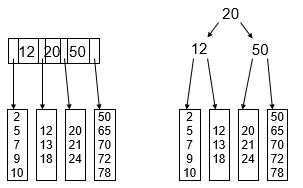
1. 

------------------------------------------------------------------------------------------------------------------------------------------

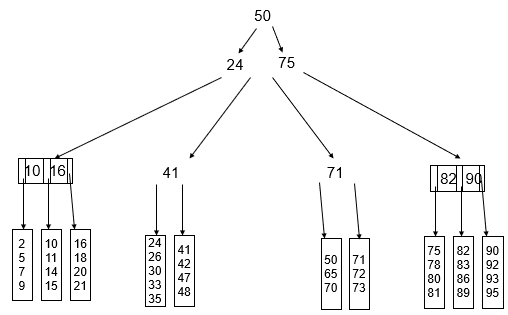
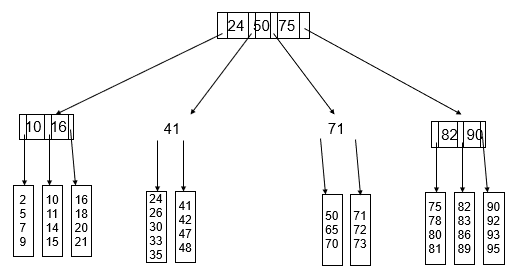
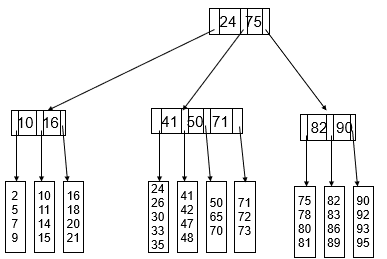
1. 



------------------------------------------------------------------------------------------------------------------------------------------

1. 

------------------------------------------------------------------------------------------------------------------------------------------

1. 

------------------------------------------------------------------------------------------------------------------------------------------

1. M-1 keys, so 22M-18 bytes. Solve for M, 2048+18/22 = M = 93

L = 2048/256 = 8

Worst case height for 500,000 records is 4

------------------------------------------------------------------------------------------------------------------------------------------

1. An algorithm that prints a tree level by level is a breadth-first traversal. A breadth-first traversal is made using a queue. You start with the root, and travel to all adjacent (level 1) vertices, putting each vertex in queue as you go. After finishing all level one vertices start doing the same thing for all level two vertices. Going to each vertex in queue and traveling to their adjacent vertices (level 2).