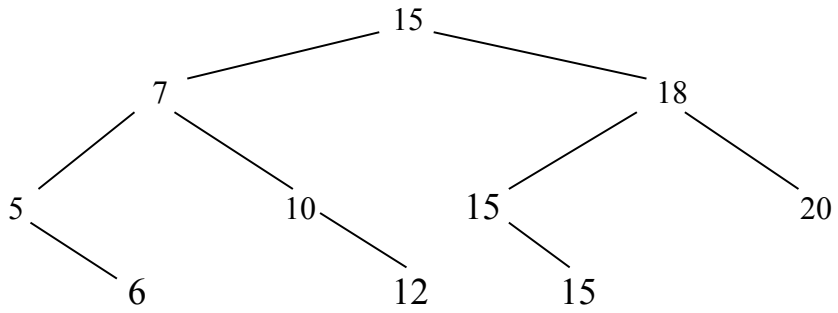
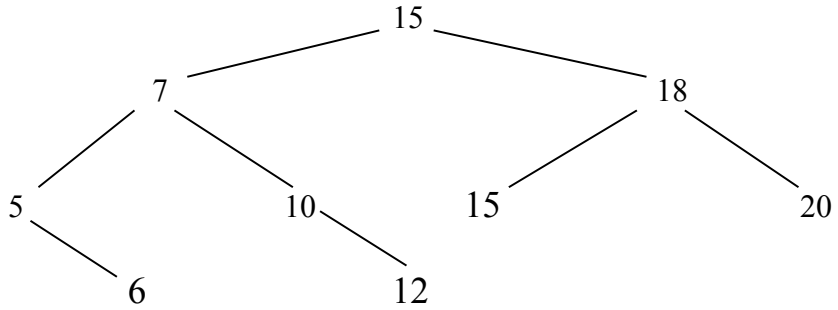
EECS560 **Practice HW#3 Solution** **10/24/18**

Instruction: You must show all your work clearly for credit. Partial credit will only be given to meaningful answers.

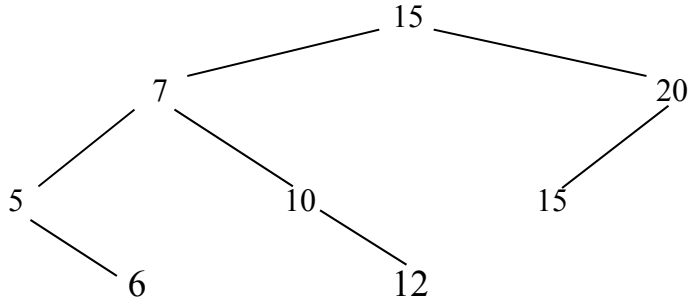
1. (30) Given a set S of 10 records with priorities {15, 7, 10, 5, 18, 20, 15, 12, 6, 15}.
   (a) Construct the BST T1 for S by inserting the records, in the given order, into an initially empty binary search tree. When done, delete 15 and then 18 from T1.
   (b) Construct the BST T2 for S by inserting the records, in the reversed given order, into an initially empty binary search tree. When done, delete 15 and then 18 from T2.
   (c) Illustrate the data structure for the final BST for T1 and T2.
   Remark: You must show your BST clearly after each insert/delete operation.
   Solution:
   (a) BST T1:



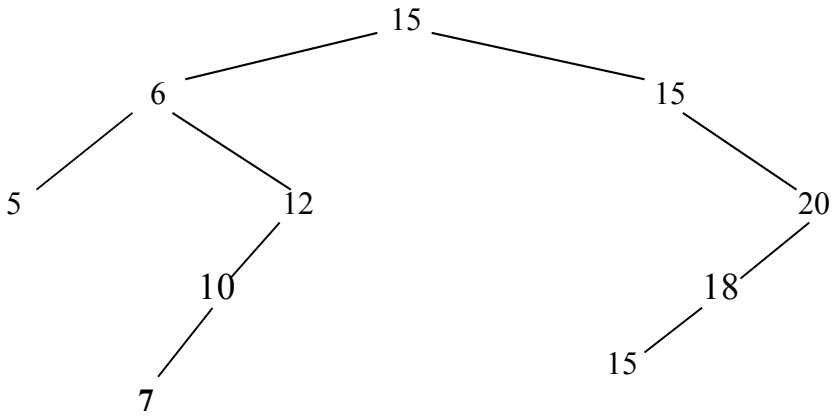
delete(15, T1):



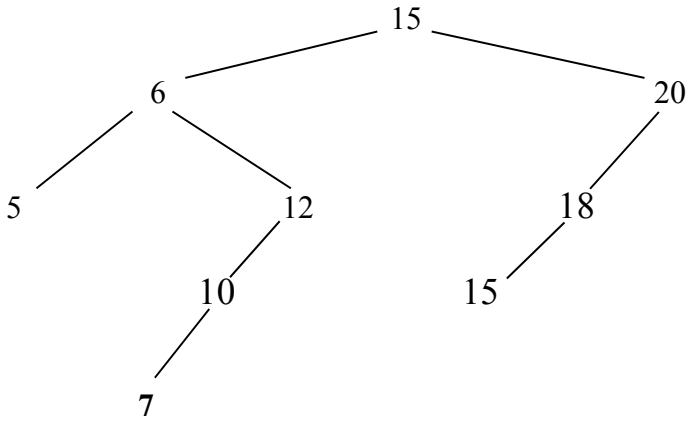
delete(18, T1):



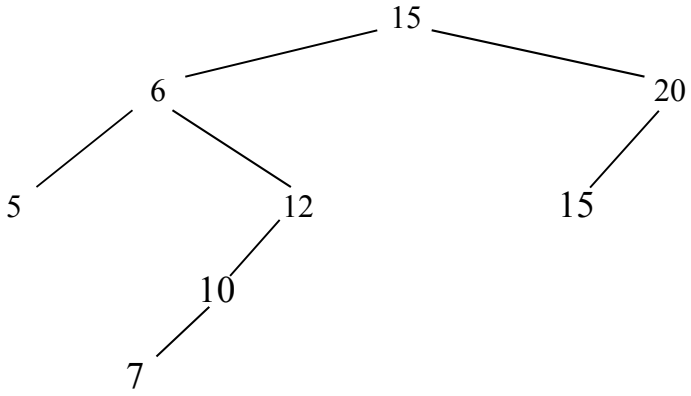
(b) BST T2:



delete(15,T2):

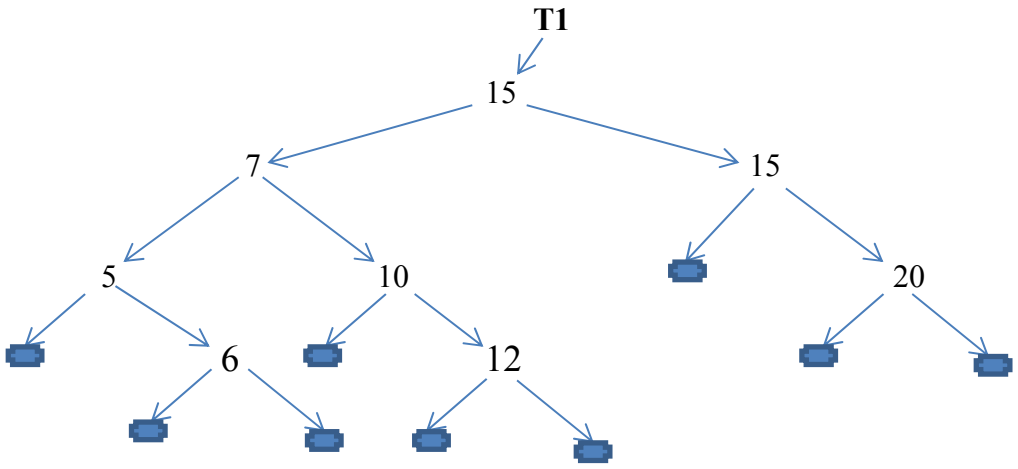


delete(18,T2):

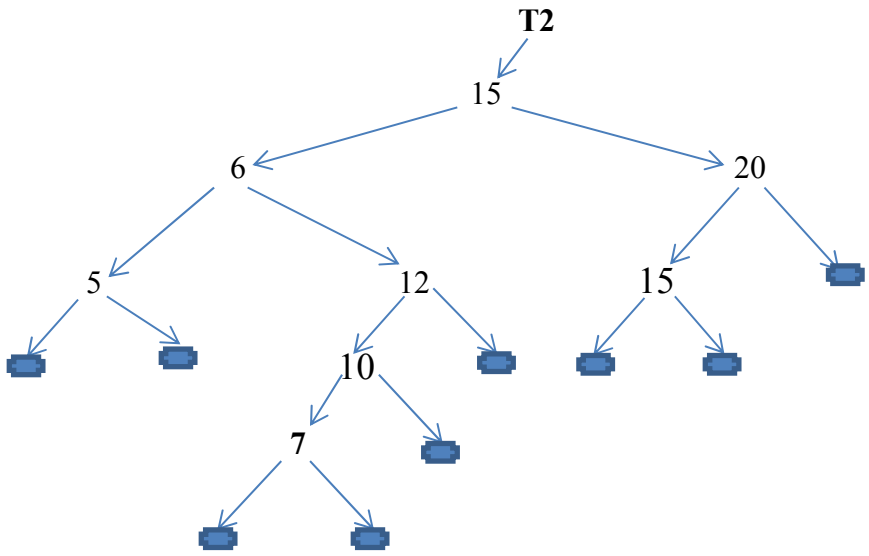


Remark: You must show all your BST after each insert/delete operation.

(c) DS for T1:



DS for T2:



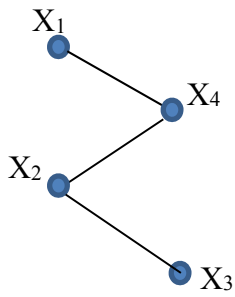
Remark: You must illustrate all pointers, including root pointer and all null pointers, for credit.

2. Given a set R of 4 records with keys x_i, $x_1 < x_2 < \ldots < x_4$, a key x, and the probability function $Pr(x = x_i) = p_i$, with $p_1 = 0.35$, $p_2 = 0.2$, $p_3 = 0.15$, $p_4 = 0.3$, $1 \le i \le 4$.
   (a) (10) Construct a greedy optimal binary search tree T3 for R using greedy approach as discussed in class. Compute the average number of comparisons in finding x in T3.
   (b) (20) Construct an optimal binary search tree T4 for R using dynamic programming technique such that the average number of comparisons in finding x in T4 is minimized.
   Remark: You must show your computations for all $c_{i,j}$, $t_{i,j}$, and the reconstruction of the optimal binary search tree clearly as discussed in class. Also, use smallest k to resolve ambiguity.

Solution:

(a) Greedy optimal BST T3:

use highest p as a root for each tree
p1,p4,p2,p3



$T_a(n) = 1*0.35 + 2*0.3 + 3*0.2 + 4*0.15 = 2.15$

(b) Constructing optimal BST using DP:

Given $\{x_1, x_2, x_3, x_4\}$ with $p_1 = 0.35$, $p_2 = 0.2$, $p_3 = 0.15$, $p_4 = 0.3$.

$c_{1,1} = 0.35,\ c_{2,2} = 0.2,\ c_{3,3} = 0.15,\ c_{4,4} = 0.3.$

$$c_{1,2} = \min\{c_{1,0} + c_{2,2}, c_{1,1} + c_{3,2}\} + \sum_{l=1}^{2} p_l = \min\{0.2, 0.35\} + 0.55 = 0.75, t_{1,2} = 1.$$

$$c_{2,3} = \min\{c_{2,1} + c_{3,3}, c_{2,2} + c_{4,3}\} + \sum_{l=2}^{3} p_l = \min\{0.15, 0.2\} + 0.35 = 0.5, t_{2,3} = 2.$$

$$c_{3,4} = \min\{c_{3,2} + c_{4,4}, c_{3,3} + c_{5,4}\} + \sum_{l=3}^{4} p_l = \min\{0.3, 0.15\} + 0.45 = 0.6, t_{3,4} = 4.$$

$$c_{1,3} = \min\{c_{1,0} + c_{2,3}, c_{1,1} + c_{3,3}, c_{1,2} + c_{4,3}\} + \sum_{l=1}^{3} p_l = \min\{0.5, 0.5, 0.75\} + 0.7 = 1.20, t_{1,3} = 1.$$

$$c_{2,4} = \min\{c_{2,1} + c_{3,4}, c_{2,2} + c_{4,4}, c_{2,3} + c_{5,4}\} + \sum_{l=2}^{4} p_l = \min\{0.6, 0.5, 0.5\} + 0.65 = 1.15, t_{2,4} = 3.$$

$$c_{1,4} = \min\{c_{1,0} + c_{2,4}, c_{1,1} + c_{3,4}, c_{1,2} + c_{4,4}, c_{1,3} + c_{5,4}\} + \sum_{l=1}^{4} p_l$$

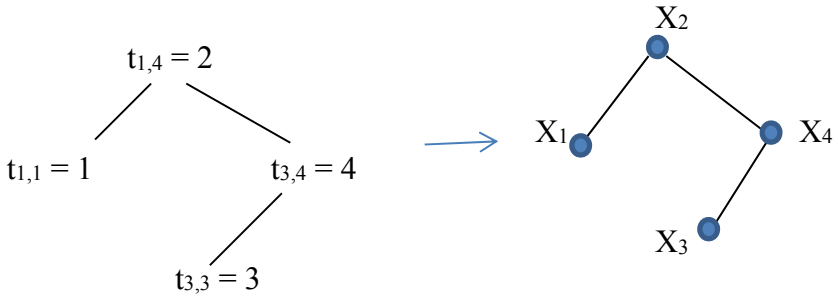
$$= \min\{1.15, 0.95, 1.05, 1.2\} + 1.0 = 1.95, t_{1,4} = 2.$$

DP-Table:

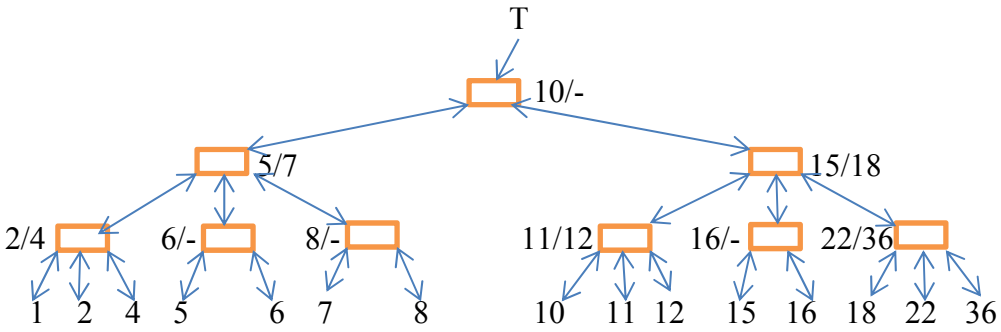
$c_{1,1} = 0.35$ $t_{1,1} = 1$	$c_{2,2} = 0.20$ $t_{2,2} = 2$	$c_{3,3} = 0.15$ $t_{3,3} = 3$	$c_{4,4} = 0.30$ $t_{4,4} = 4$
$c_{1,2} = 0.75$ $t_{1,2} = 1$	$c_{2,3} = 0.50$ $t_{2,3} = 2$	$c_{3,4} = 0.60$ $t_{3,4} = 4$	
$c_{1,3} = 1.20$ $t_{1,3} = 1$	$c_{2,4} = 1.15$ $t_{2,4} = 3$		
$c_{1,4} = 1.95$ $t_{1,3} = 2$			

$t_{1,4} = 2$

Constructing Optimal BST T4:

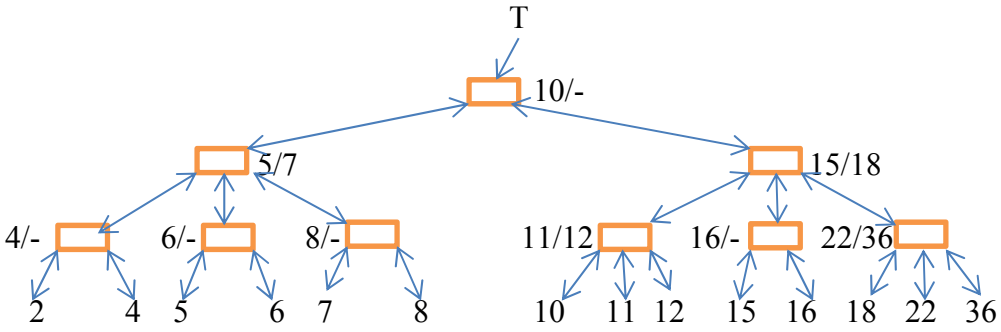


3. Given a set H of 15 records with priorities {10, 5, 2, 15, 8, 12, 4, 1, 18, 6, 16, 11, 22, 36, 7}.
   (a) (15) Construct the 2-3 tree T5 for H by inserting the records, in the given order, into an initially empty 2-3 tree. When done, delete 1, 15, and then 36 from T5.
   (b) (15) Construct the 2-3 tree T6 for H by inserting the records, in the reversed given order, into an initially empty 2-3 tree. When done, delete 1, 15, and then 36 from T6.
   (d) (10) Illustrate the data structure for the final 2-3 tree for T5 and T6.
   Remark: You must show your 2-3 tree, including the content of each interior node clearly after each insert/delete operation.
   Solution:
   (a) 2-3 tree T5:

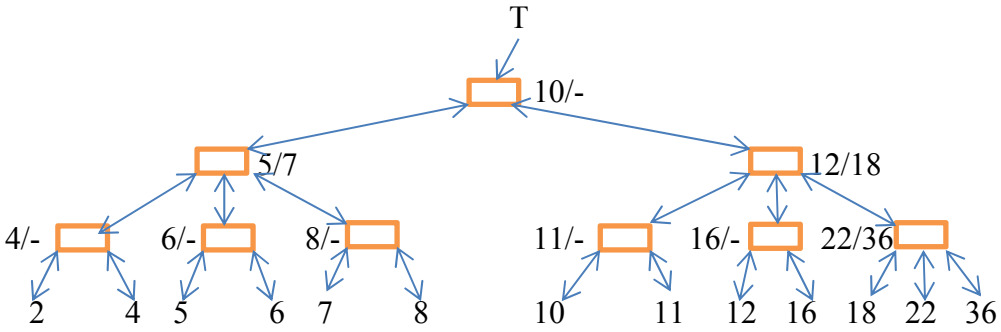


Remark: Whenever a 3-node becomes a 4-node, you must perform node splitting.

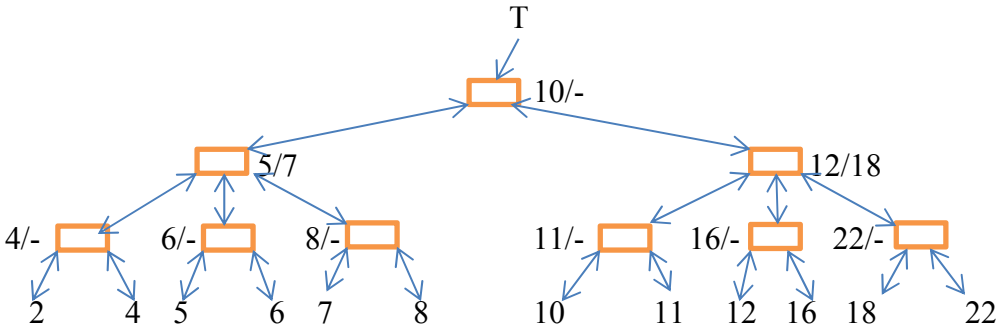
delete(1,T5):



delete(15,T5):

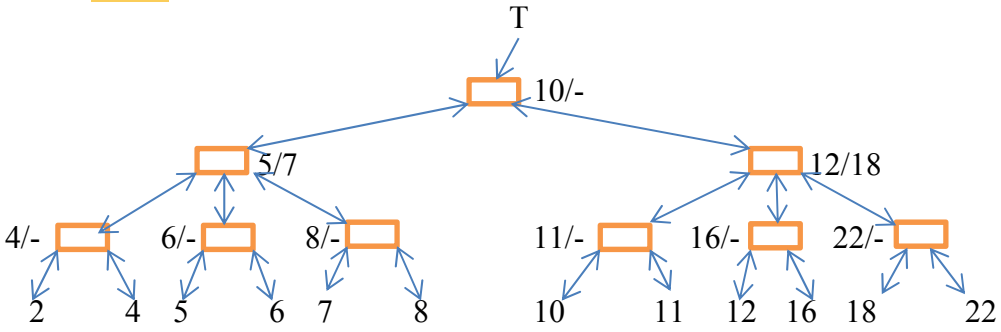


delete(36,T5):

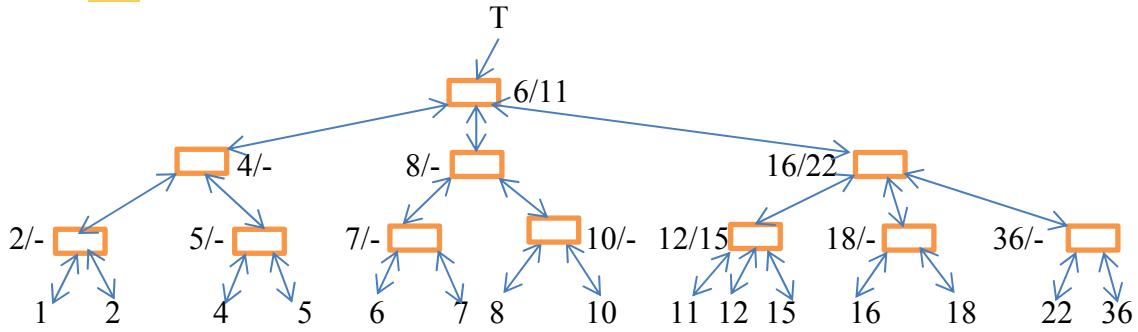


Remark: To resolve ambiguity, you must check the left sibling before the right one. Also, an 1-node will always try to adopt from its immediate sibling before giving up its only child for adoption.

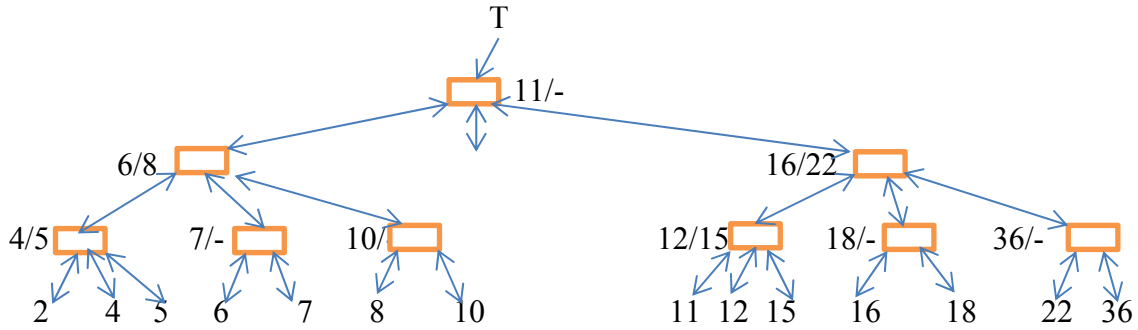
DS for final 2-3 tree T5:



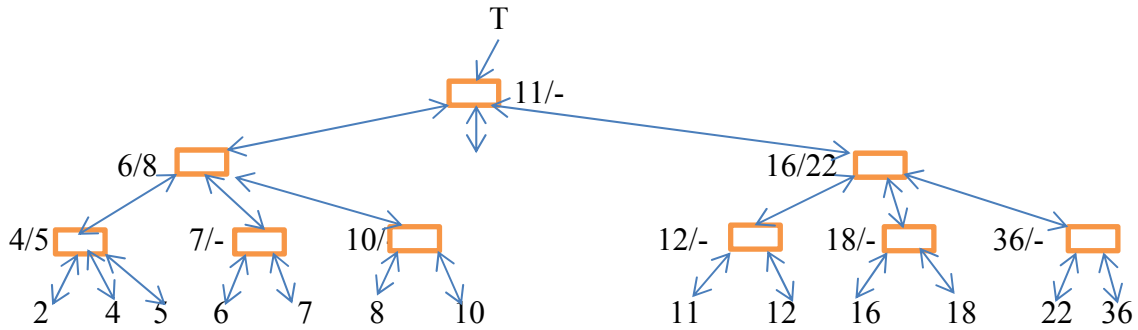
(b) 2-3 tree T6:



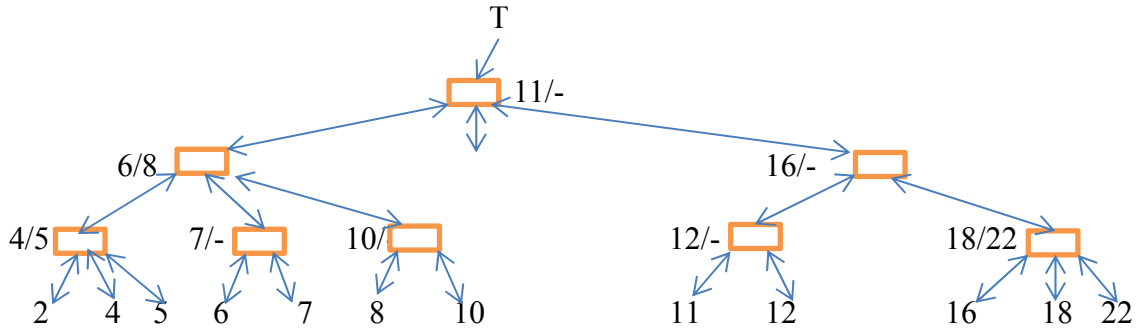
delete(1,T6):



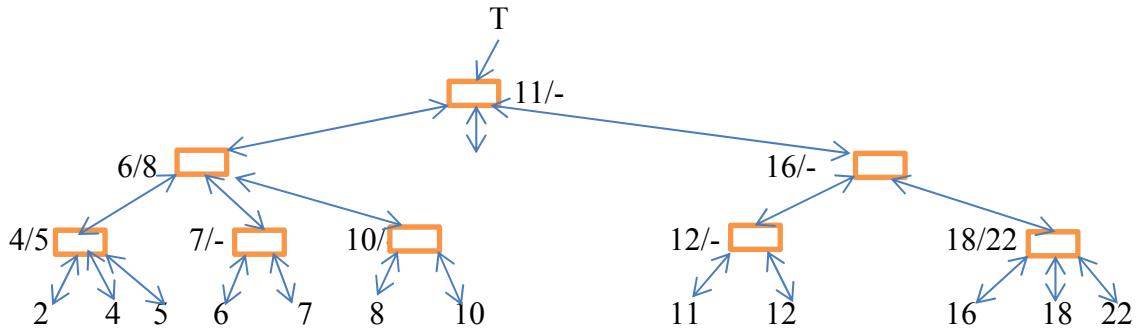
delete(15,T6):



delete(36,T6):



DS for final 2-3 tree T6:



10/16/18