ADS Lab Writeup Shikha-N 1 BM18CS149 AUL Trees Insection Node \* insert (Node \* nodo, int key) if node == NULL setuen New Node (key)
// normal BST insection if key / node -> key node - left = insert (node - kleght, key) else if bey > node = key node -> right = insert (node -> right, setuen node rode - height = 1 + max (height (node > left), // up date his get height (node-right)) Il calculate Bolance Jactor int balance = getBalance (node) llif Unbalanced. 11 Case 1 => Left rotation. if (balance L-1 & & key) > node - s sight - s return leftRotate (node) 11 case 2 3 sight sotation. if (balavee > 14g key < node > left > Setun sight Rotate (node)

11 lest Right Rotation. Shikhan IBNI (805/49 if balance > 1 & S. Key > node > Rest > key

h node > lest = lest Robate (node > lest

ketuen eight Robate (node) 11 Right - Left Rotation if balance <-1 ff key (node - light-sky node = right = Right Rotate (node) Letuen left Rotate (node) }. Letuen mode m. I de proprié de la financia del la financia de l and provide the has been deal . I'm

Deletion >> Shikha N Node delete Node (Node mode, int ky) 11 MIECSILIA Who -sout if (node = = NULL) if (key < node > key) I node - lest = delete Node (mode > lest, else if (key > rode > key) I node -> right = deleterode (node -> right, 11 node with only one child/no child if (node > left = = NULL ! node -> right == NULL) Node \* x = lost > left? root -> left: root > sight if de nou node = NULL se a rode = & & } 11 one child & bree (+)

node with a children. Shikha-N else 1 BMIBCS 149 Node \* temp = inorder - successor (node > right) node - key temps key node > right = delete Node (node > right, temp = key) poble = = NULL leturn node Mode -> height = 1 + max (height (node -> (eft), height (node->right)) int balance = get Balance (no de) 11 Right Rotation (balance > ) & & get Balance (node > left) // Lest Rotation + Right Rotation if (balance > 1 & & getBalance (node -> lest) < 9 h node = lest = lest Rotate (node = lest) letuen right Rotate (node) 11 lest Rotation if (balance <-1 & & get Balance (node -> right) (20) setuen lytrotate (node)

// Right left sotation if (balance <- 1 & §

(get Galance (mode > right) > 0) node -> night = light Rotate (node-signi) setuen sext Rotate (no de) Letur node No de \* lest Rotate (No de \* temp) Node \* y = temp-> right Node &x 2 y -> lest # > left = temp. temp -> right = 1. temp -> height = max (height (temp > (est), height (temp szight)) y -> height = mare ( height (y > left) 9 height (y -> sight))+ Julium y

Node & Right Rotate (Node +y) Shisho-N Node + x = y > left Node & temp = 2 -> Bight X -> sight = Y y -> left = temp. y > height = max / height (y > left), neight (y = sight)) + 1 7 2 height - mad (height (2 - left), height (x=>zight))+) Leturn 1 mil int get Balance (Node & node) if rode = = NULL letnen height (node - s left) height ( node > right)