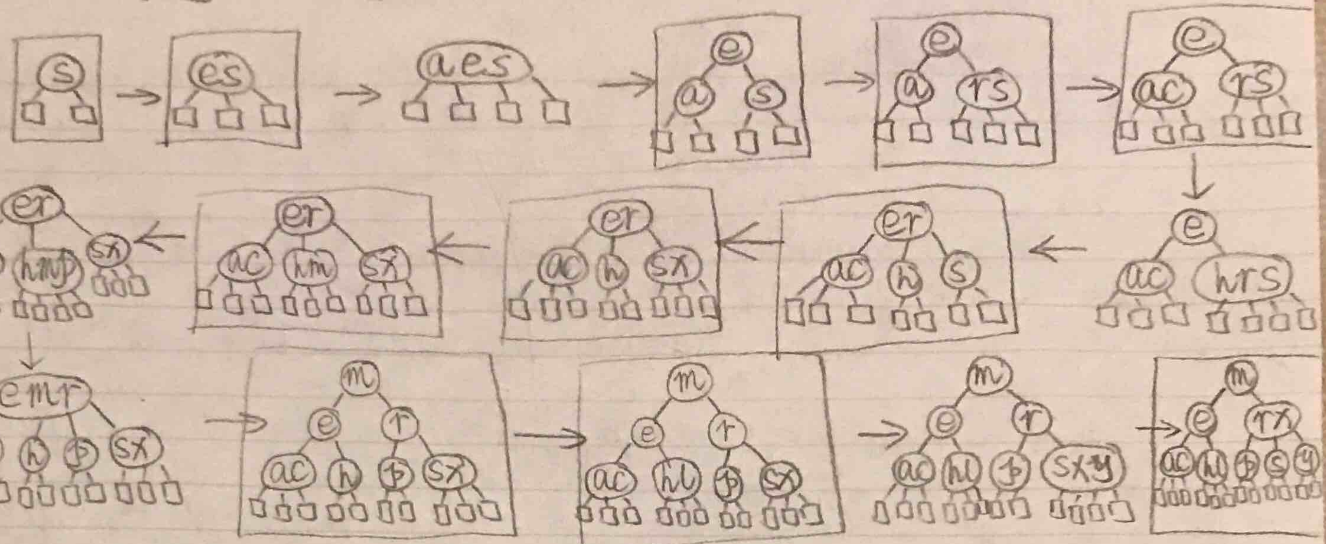
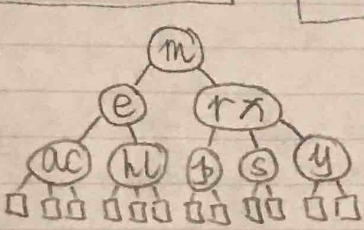
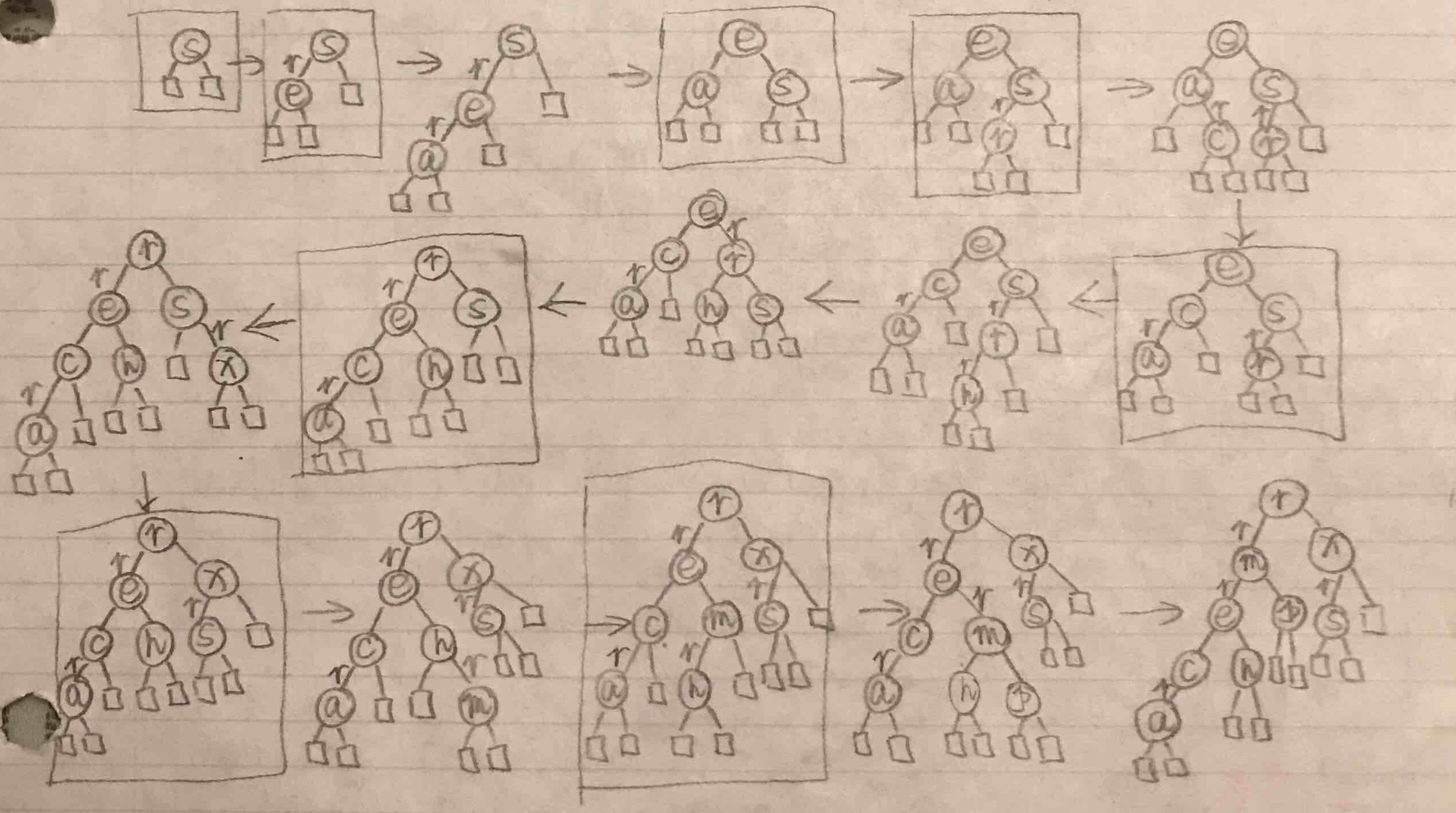
① Xiaoqing Zhang Assignment 2 CSC 226

1. @



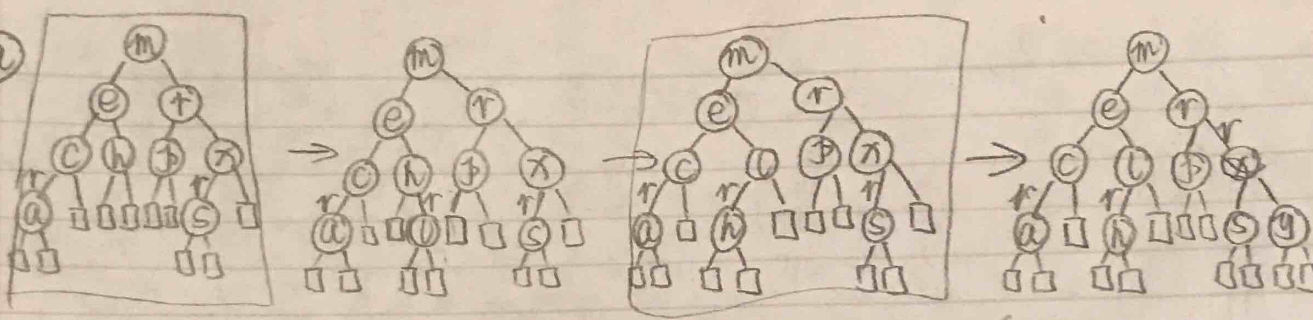
2-3 tree :



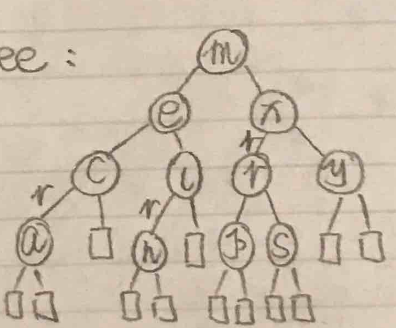


②

1. ⓐ



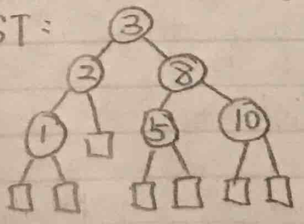
red-black Tree :



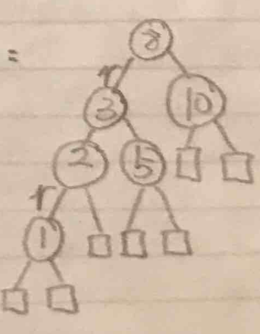
1. ⓑ There is a sequence of keys to insert into a BST and a red-black BST such that the height of the BST is less than the height of the red-black BST :

insert 3, 2, 8, 10, 5, 1

BST :



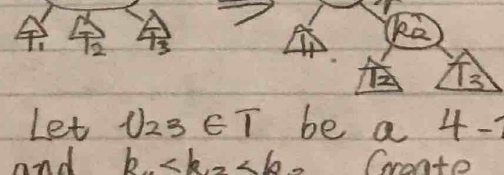
Red-black BST :



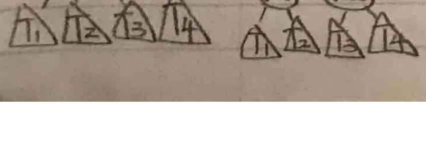
2. ⓐ
```
public void put(Key key, Value val){
        root = put(root, key, val);
        root.color = BLACK;
}
private Node put(Node h, Key key, Value val){
    if (h == null) {
        return new Node(key, val, 1, RED);
    }
    int comp = key.compareTo(h.key);
    if      (comp <0) h.left = put(h.left, key, val);
    else if (comp>0) h.right = put(h.right, key, val);
    else h.val = val;

    if (isRed(h.left) && !isRed(h.right)){
        h = rotateRight(h);
    }
    if (isRed(h.right) && isRed(h.right.right)){
        h = rotateLeft(h);
    }
    if (isRed(h.right) && isRed(h.left)) {
        flipColors(h);
    }
    h.N = size(h.left) + size(h.right) + 1;
    return h;
}
```

2. ⓑ and k_2, $k_1 < k_2$. Create a T' such that k_2 is k_1's right child and k_1 is black, k_2 is red

ex.


<u>case 3</u>: Let $v_{23} \in T$ be a 4-node with k_1, k_2, k_3 and $k_1 < k_2 < k_3$. Create a T' such that k_2 is the parent of k_1 and k_3, k_1 is left child, k_3 is right child, and k_2 is black, k_1 and k_3 are red.



③

2. ⓑ <u>Proof</u>: ① Let T be a red-black right-leaning tree, we will construct the corresponding 2-3 tree T'.

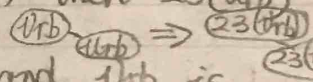
<u>Nodes</u>:

<u>case 1</u>: Let $v_{rb} \in T$ be a node with key k such that it has no red incident edges. Create a 2-node in T' called v_{23} and insert key k into it. Map v_{rb} to v_{23} via $23(v_{rb}) = v_{23}$

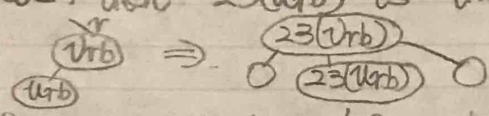
<u>case 2</u>: For every red edge in T, let v_{rb} and w_{rb} be the incident nodes such that k_1 in v_{rb} > k_2 in w_{rb} : $w_{rb} \xrightarrow{r} v_{rb}$. Let $v_{23} \in T'$ be a 3-node with key k_1 and k_2 in that order. Map $23(v_{rb}) = v_{23}$ and $23(w_{rb}) = v_{23}$: $(k_2\ k_1)\ v_{23}$

<u>Edges</u>: Let $u_{rb}v_{rb}$ be an edge in T such that it's black, then $23(u_{rb})(v_{rb})$ is an edge in T' with are of the following cases.

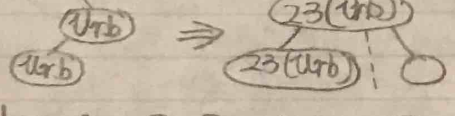
<u>case 1</u>: If u_{rb} is the right child v_{rb}, then $23(u_{rb})$ is the right child of $23(v_{rb})$:



<u>case 2</u>: If u_{rb} is the left child v_{rb} and v_{rb} is red, then $23(u_{rb})$ is the middle child of $23(v_{rb})$



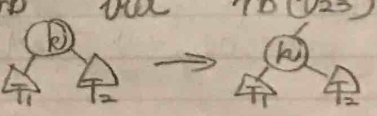
<u>case 3</u>: If u_{rb} is the left child of v_{rb} and v_{rb} is black then $23(u_{rb})$ is the left child of $23(v_{rb})$



② Let T be a 2-3 tree T, we will construct the corresponding red-black right-leaning tree T'

<u>case 1</u>: Let a 2-node $v_{23} \in T$ be a node with key k, create a black node in T' called v_{rb}. Map v_{23} to v_{rb} via $rb(v_{23}) = v_{rb}$

ex.



<u>case 2</u>: Let $v_{23} \in T$ be a 3-node with key k_1

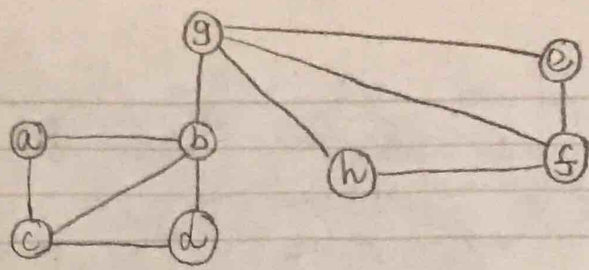
2. (b) ∴ We proven for every Red-black right-leaning tree, there is a corresponding 2-3 Tree

∵ We proven for every Red-black left-leaning ~~tree~~, there is a corresponding 2-3 tree in class and class slides.

∴ For every right-leaning red-black tree there is a correspond left-leaning red-black tree.

2. b ∵ We proven for every Red-black right-leaning tree, there is a corresponing 2-3 Tree. ∵ in class and class slide proven

3. ⓐ



ⓑ
a - b - g - h
a - b - g - e - f - h
a - b - g - f - h

a - c - b - g - h
a - c - b - g - e - f - h
a - c - b - g - f - h

a - c - d - b - g - h
a - c - d - b - g - e - f - h
a - c - d - b - g - f - h

∴ There are 9 paths in G from a to h

ⓒ paths have length less than 5:

a - b - g - h
a - b - g - f - h
a - c - b - g - h

4. <u>Base case</u>: $n=1$, graph has 1 vertice, 0 edge, $0 \leq 1^2 - 1 = 0$

$n=2$, graph has 2 vertics, 1 edge: $m=1$, $n=2$

$2 \cdot 1 = 2 \leq 2^2 - 2 = 2$

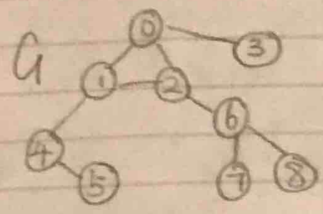
<u>Induction Hypothesis</u>: Assume there is a $k \geq 1$ such that $2m \leq k^2 - k$ where $k \leq n$ and $k \in N$

<u>Induction Step</u>: if there are k vertices, when we add $(k+1)$th vertice, the edge will add up to k. Thus, the edge total now would be $m+k$

$$2(m+k) \leq (k^2 - k) + 2k \text{ (by I.H.)}$$
$$2m + 2k \leq k^2 - k + 2k$$
$$2k + 2m \leq k^2 + k$$
$$2k + 2m \leq k^2 + 2k - k + 1 - 1$$
$$2k + 2m \leq (k^2 + 2k + 1) - (k+1)$$
$$2k + 2m \leq (k+1)^2 - (k+1) \implies 2(m+k) \leq (k+1)^2 - (k+1)$$

∴ The indentity holds for $n+1$, and by induction, the indentity holds for all $n \geq 0$.

5. G

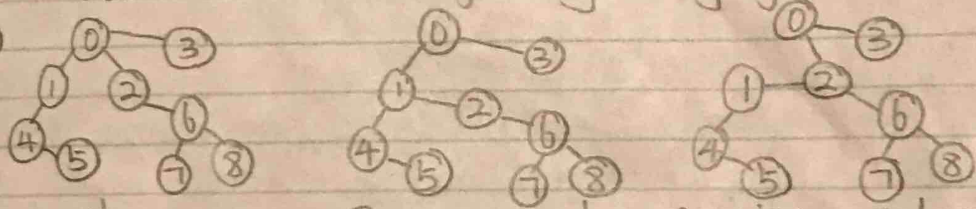


ⓐ # of spanning subgraph with no edge = 1
# of spanning subgraph with 1 edge = 9
# of spanning subgraph with 2 edges = 36
# of spanning subgraph with 3 edges = 84
# of spanning subgraph with 4 edges = 126
# of spanning subgraph with 5 edges = 126
# of spanning subgraph with 6 edges = 84
# of spanning subgraph with 7 edges = 36
# of spanning subgraph with 8 edges = 9
# of spanning subgraph with 9 edges = 1

∴ $1+9+36+84+126+126+84+36+9+1 = 512$

∴ there are 512 spanning subgraphs

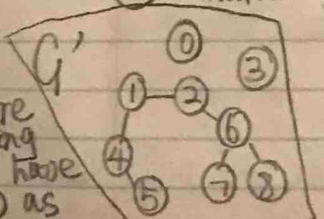
ⓑ



∴ There are 3 connected spanning subgraphs

ⓒ make graph G to G' which vertex 0 is isolater vertex.

G'



# of spanning subgraph with no edge in G' = 1
# of spanning subgraph with 1 edge in G' = 6
# of spanning subgraph with 2 edge in G' = 15
# of spanning subgraph with 3 edge in G' = 20
- - - 4 edge in G' = 15
- - - 5 edge in G' = 6
- - - 6 edge in G' = 1

$1+6+15+20+15+6+1 = 64$

∴ There are 64 spanning subgraph have vertex 0 as isolated vertex