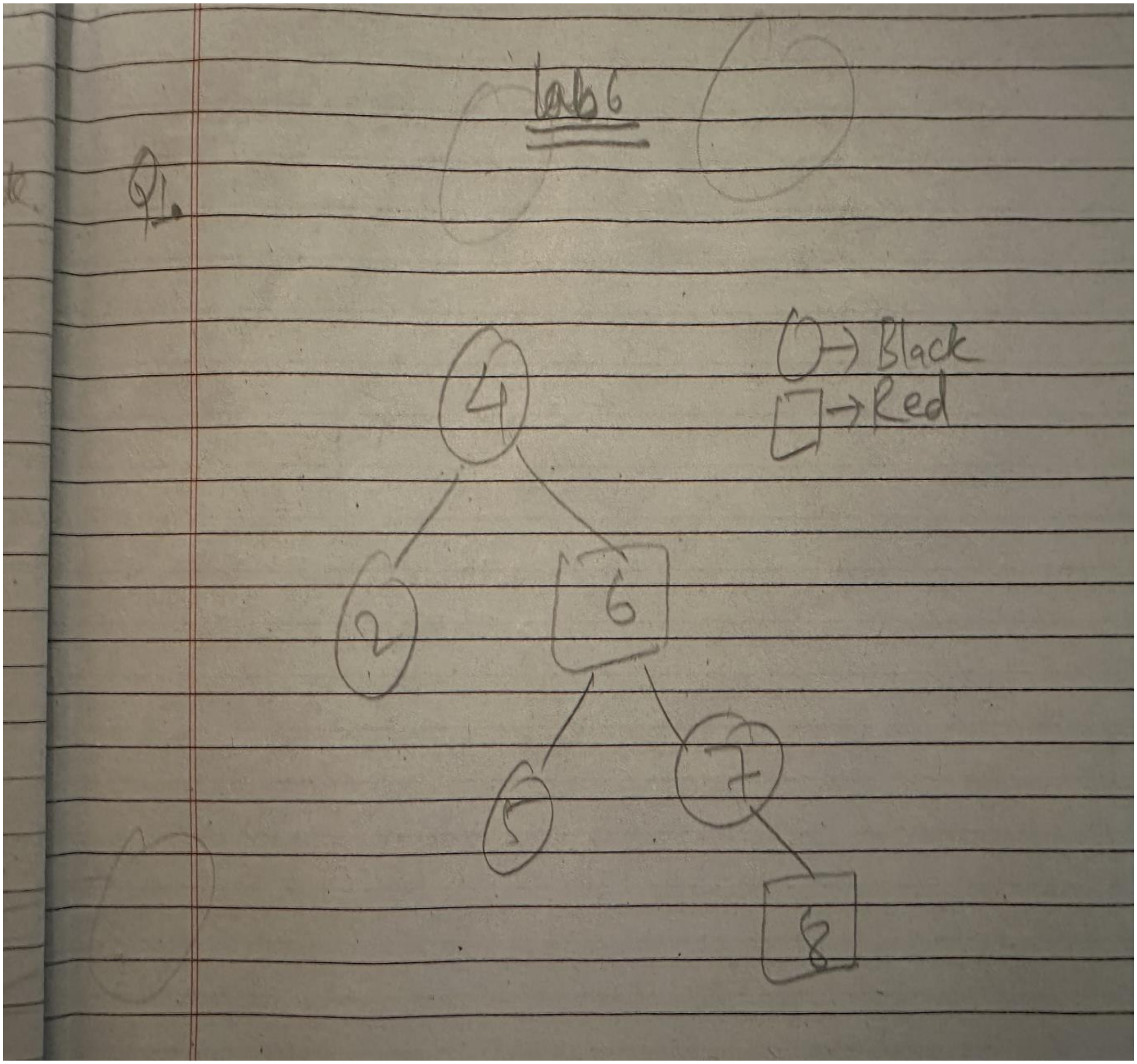
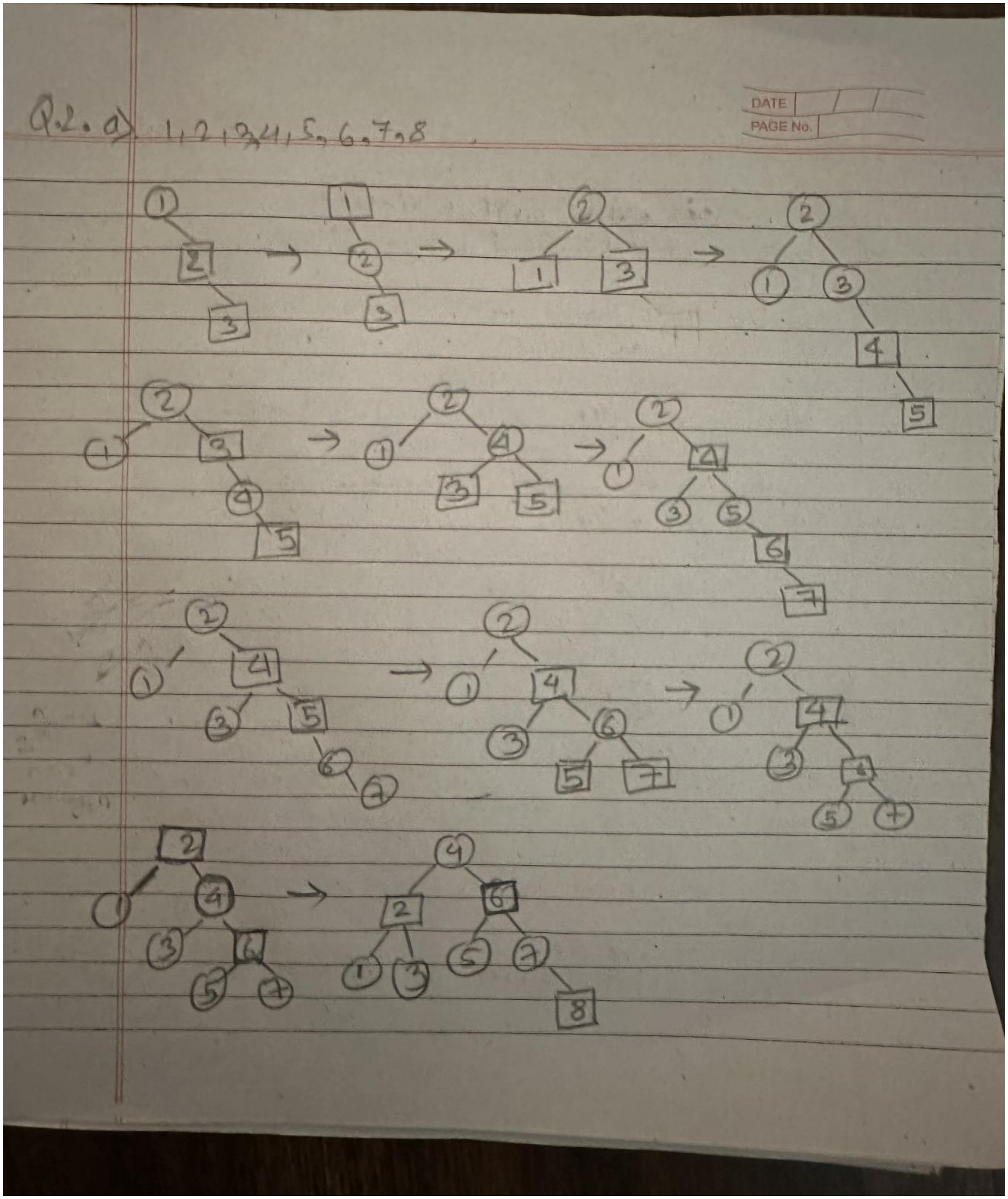
Q1.

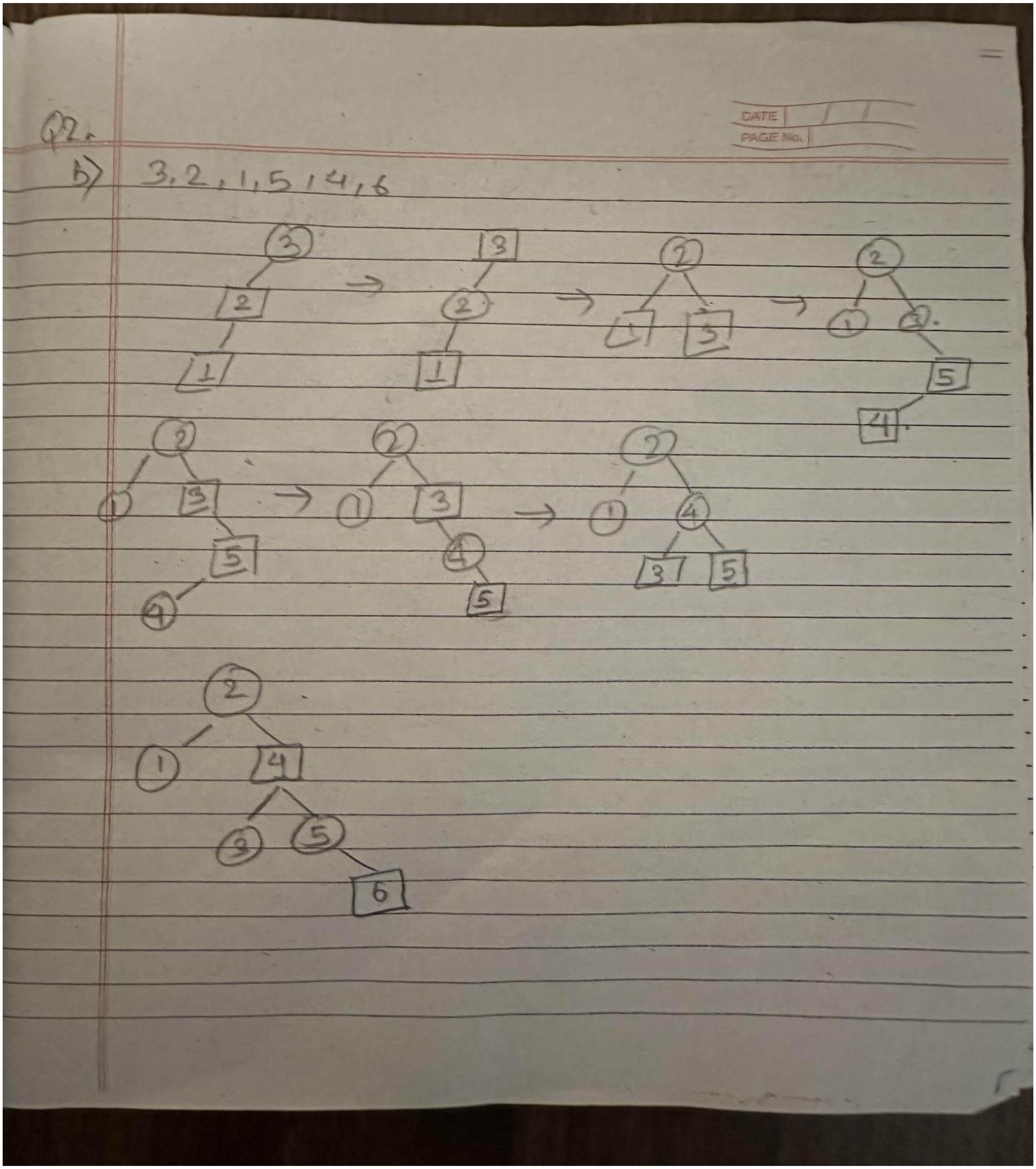


Q2.a.



Q.2. a) 1,2,3,4,5,6,7,8

Q2.b.



Q2.

b) 3,2,1,5,4,6

Q3.

```
class Trie {
```

```
class TrieNode{
TrieNode[] dictionary;
boolean eow;
TrieNode(){
dictionary = new TrieNode[26];
eow=false;
}
}

TrieNode[] dictionary;

public Trie() {
// only small letters are considered
this.dictionary = new TrieNode[26];
}
public void insert(String word) {
TrieNode currentNode = dictionary[word.charAt(0)-'a'];
if(currentNode == null){
currentNode = new TrieNode();
dictionary[word.charAt(0)-'a'] = currentNode;
}

for(int i=1;i<word.length();i++){
char c = word.charAt(i);
TrieNode node = currentNode.dictionary[c-'a'];
if(node != null){
currentNode = node;
}else{
TrieNode n = new TrieNode();
currentNode.dictionary[c-'a'] = n;
currentNode = n;
}
}
currentNode.eow = true;
}
public boolean search(String word) {
TrieNode currentNode = dictionary[word.charAt(0) - 'a'];
if(currentNode == null) return false;

for(int i=1;i<word.length();i++){
char c = word.charAt(i);
TrieNode node = currentNode.dictionary[c-'a'];
if(node != null){
```

```
currentNode = node;
}else{
return false;
}
}
return currentNode.eow;
}
public boolean startsWith(String prefix) {
TrieNode currentNode = dictionary[prefix.charAt(0) - 'a'];
if(currentNode == null) return false;

for(int i=1;i<prefix.length();i++){
char c = prefix.charAt(i);
TrieNode node = currentNode.dictionary[c-'a'];
if(node != null){
currentNode = node;
}else{
return false;
}
}
return true;
}
}
```