Sudeshna Bhushan, 1BM19CS189

21/12/20

Binary Search Tree

```
typedef struct BST {
    int data;
    struct BST* right;
    struct BST* left;
} node;

node * create (int data) {
 node *temp = (node *) malloc (sizeof(node));
 temp -> data = data;
 temp -> right = temp -> left = NULL;
 return temp;
}

void insert (node * root, node * temp) {
 if (temp -> data < root -> data) {
    if (root -> left != NULL)
      insert (root -> left, temp);
    else
      root -> left = temp;
 }
 else {
        if (root -> right != NULL)
            insert (root -> right, temp);
        else
            root -> right = temp;
        }
}
```

```
void inorder (node * root)
{
  if (root == NULL)
      return;
  inorder (root -> left);
  printf ("%d", root -> data);
  inorder (root -> right);
}
void preorder (node * root ){
  if (root == NULL)
      return;
  printf ("%d", root -> data);
  preorder (root -> left);
  preorder (root -> right);
}
void postorder (node * root {
  if (root == NULL)
      return;
  postorder (root -> left);
  postorder (root -> right);
  printf ("%d", root -> data);
}
```

— x — x —