Binary Search Trees

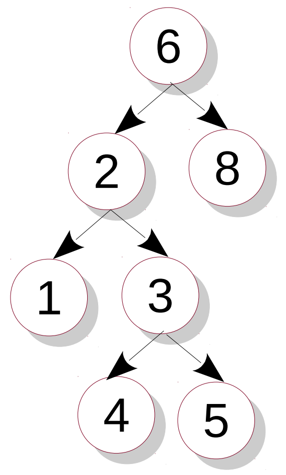
Rupesh Nasre.
rupesh@iitm.ac.in

Web: ~rupesh/teaching/pds/spw2019

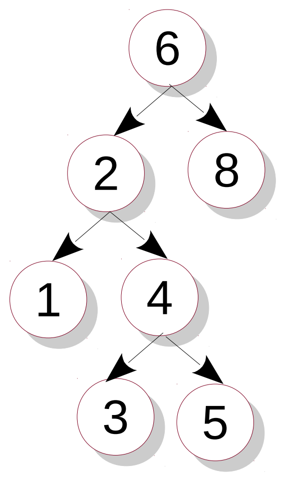
Summer Projects and Workshop
June 2019

Definition

- A BST is a binary tree.
- If it is non-empty, the value at the root is larger than any value in the left-subtree, and
- the value at the root is smaller than any value in the right-subtree.
- The left and the right subtrees are BSTs.
- Assumption: All values are unique.



Not a BST



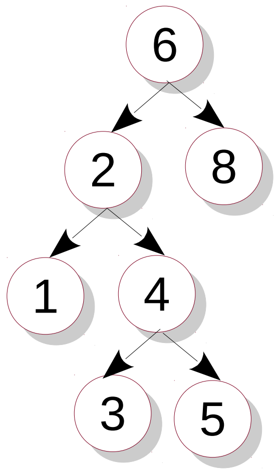
BST

BST Operations

```
class BST {
        ...
public:
        ...
        PtrToNode search(DataType element);
        PtrToNode findMin();
        PtrToNode findMax();
        PtrToNode insert();
        void remove();
};
```

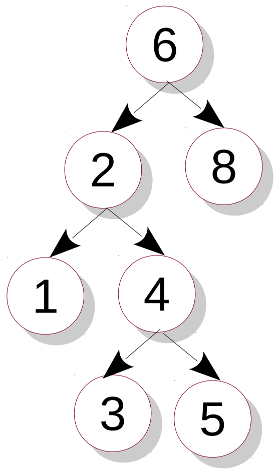
Search

```
PtrToNode search(DataType element) {
    return search(element, root);
}
PtrToNode search(DataType element, PtrToNode rr) {
    if (rr) {
        if (rr→data == element) return rr;
        if (element < rr→data)
            return search(element, rr→left);
        return search(element, rr→right);
    }
    return NULL;
}
```



FindMin

```
PtrToNode findMin() {
    ptr = root;
    if (ptr) {
        while (ptr→left) {
            ptr = ptr→left;
        }
    }
    return ptr;
}
```



```
PtrToNode findMin(PtrToNode ptr) {
    if (ptr) {
        if (ptr→left)
            return findMin(ptr→left);
    }
    return ptr;
}
```

Some Questions?

- What if a BST has duplicates?
- Can a BST node contain strings? Other types?
- Can I store more pointers in a node?

Exercises

Learning Outcomes