```
---------------------------------Tree------------------------
------BFT--------DFT------------------

class Node {
  constructor(data){
    this.data = data;
    this.children = [];
  }

  add(data){
    this.children.push(new Node(data));
  }

  remove(data){
    this.children = this.children.filter(node => {
      return node.data !== data;
    })
  }
}

class Tree {
  constructor(){
    this.root=null;
  }

  traverseBF(fn){
    const arr = [this.root];
    while(arr.length){
      const node = arr.shift();
      arr.push(...node.children);
      fn(node);

    }
  }

  traverseDF(fn){
    const arr = [this.root];
    while(arr.length){
      const node = arr.shift();
      arr.unshift(...node.children);
      fn(node);
    }
  }
}

const node = new Node(1);
node.add(2)
node.add(3)
const tree = new Tree();
tree.root= node;
tree.traverseBF(node => console.log(node))
```