

```

class ArvoreBinaria {

    private No raiz;
    int conta, soma;
    int maior = 0;
    int menor = 2147483647;

    public ArvoreBinaria(){
        this.raiz = null;
    }

    public No ObtemRaiz()
    {
        return this.raiz;
    }

    public void InsOrd(int novo)
    {
        No n = new No(novo);
        this.raiz = this.Insere(n, this.raiz);
    }

    private No Insere(No novo, No inicio)
    {
        if (inicio != null)
        {
            if (novo.ObtemValor() < inicio.ObtemValor())
                inicio.p_Esq=this.Insere(novo, inicio.p_Esq);
            else inicio.p_Dir=this.Insere(novo, inicio.p_Dir);
            return inicio;
        } else return novo;
    }

    //Ordem crescente
    public void ImprimirCrescente(No raiz) {
        if (raiz != null) {
            this.ImprimirCrescente(raiz.p_Esq);
            System.out.print(raiz.ObtemValor() + ", ");
            this.ImprimirCrescente(raiz.p_Dir);
        }
    }

    // Ordem decrescente
    public void ImprimirDecrescente(No raiz)
    {
        if (raiz != null){

            this.ImprimirDecrescente(raiz.p_Dir);
            System.out.print(raiz.ObtemValor() + ", ");
            this.ImprimirDecrescente(raiz.p_Esq);
        }
    }

    public int ContaElementos(No raiz)
    {
        if (raiz != null){
            conta=this.ContaElementos(raiz.p_Esq)+this.ContaElementos(raiz.p_Dir)+1;
            return conta;
        } else return 0;
    }
}

```

```
public int SomaElementos(No raiz)
{
    if (raiz != null){
        soma=this.SomaElementos(raiz.p_Esq)+this.SomaElementos(raiz.p_Dir)+raiz.ObtemValor();
    } else return 0;
}

public int MaiorElemento(No raiz)
{
    if (raiz != null){
        if (maior < raiz.ObtemValor())
            maior = raiz.ObtemValor();
        this.MaiorElemento(raiz.p_Esq);
        this.MaiorElemento(raiz.p_Dir);
        return maior;
    } else return 0;
}

public int MenorElemento(No raiz)
{
    if (raiz != null){
        if (menor > raiz.ObtemValor())
            menor = raiz.ObtemValor();
        this.MenorElemento(raiz.p_Esq);
        this.MenorElemento(raiz.p_Dir);
        return menor;
    } else return 0;
}
```