

```

public class Csomopont
{
    13 references
    public int Adat { get; set; }
    14 references
    public Csomopont Bal { get; set; }
    15 references
    public Csomopont Jobb { get; set; }
    1 reference
    public Csomopont(int adat)
    {
        Adat = adat;
        Bal = null;
        Jobb = null;
    }
}

```

```

public class BinFa
{
    8 references
    public Csomopont Gyoker { get; private set; }
    7 references
    public void Beszuras(int adat)
    {
        Gyoker = BeszurasRekurziv(Gyoker, adat);
    }
    3 references
    private Csomopont BeszurasRekurziv(Csomopont csomopont, int adat)
    {
        if (csomopont == null)
            return new Csomopont(adat);

        if (adat < csomopont.Adat)
            csomopont.Bal = BeszurasRekurziv(csomopont.Bal, adat);
        else if (adat > csomopont.Adat)
            csomopont.Jobb = BeszurasRekurziv(csomopont.Jobb, adat);

        return csomopont;
    }
}

```

```
public bool Kereses(int adat)
```

```
{  
    return KeresesRekurziv(Gyoker, adat);  
}
```

3 references

```
private bool KeresesRekurziv(Csomopont csomopont, int adat)
```

```
{  
    if (csomopont == null)  
        return false;  
    if (adat == csomopont.Adat)  
        return true;  
    return adat < csomopont.Adat  
        ? KeresesRekurziv(csomopont.Bal, adat)  
        : KeresesRekurziv(csomopont.Jobb, adat);  
}
```

1 reference

```
public void Torles(int adat)
```

```
{  
    Gyoker = TorlesRekurziv(Gyoker, adat);  
}
```

```
private Csomopont TorlesRekurziv(Csomopont csomopont, int adat)
```

```
{  
    if (csomopont == null)  
        return null;  
    if (adat < csomopont.Adat)  
        csomopont.Bal = TorlesRekurziv(csomopont.Bal, adat);  
    else if (adat > csomopont.Adat)  
        csomopont.Jobb = TorlesRekurziv(csomopont.Jobb, adat);  
    else  
    {  
        // Nincs gyermek  
        if (csomopont.Bal == null && csomopont.Jobb == null)  
            return null;  
        // Egy gyermek  
        if (csomopont.Bal == null)  
            return csomopont.Jobb;  
        if (csomopont.Jobb == null)  
            return csomopont.Bal;  
        // Két gyermek: utódot találunk (legkisebb érték a jobb ágon)  
        csomopont.Adat = MinimumErtek(csomopont.Jobb);  
        csomopont.Jobb = TorlesRekurziv(csomopont.Jobb, csomopont.Adat);  
    }  
    return csomopont;  
}
```

```
private int MinimumErtek(Csomopont csomopont)
{
    while (csomopont.Bal != null)
        csomopont = csomopont.Bal;
    return csomopont.Adat;
}
```

2 references

```
public void InorderBejaras()
{
    Console.WriteLine("Inorder bejárás:");
    InorderRekurziv(Gyoker);
    Console.WriteLine();
}
```

3 references

```
private void InorderRekurziv(Csomopont csomopont)
{
    if (csomopont == null) return;
    InorderRekurziv(csomopont.Bal);
    Console.Write(csomopont.Adat + " ");
    InorderRekurziv(csomopont.Jobb);
}
```

```
public void PreorderBejaras()
```

```
{
```

```
    Console.WriteLine("Preorder bejárás:");
```

```
    PreorderRekurziv(Gyoker);
```

```
    Console.WriteLine();
```

```
}
```

3 references

```
private void PreorderRekurziv(Csomopont csomopont)
```

```
{
```

```
    if (csomopont == null) return;
```

```
    Console.Write(csomopont.Adat + " ");
```

```
    PreorderRekurziv(csomopont.Bal);
```

```
    PreorderRekurziv(csomopont.Jobb);
```

```
}
```

1 reference

```
public void PostorderBejaras()
```

```
{
```

```
    Console.WriteLine("Postorder bejárás:");
```

```
    PostorderRekurziv(Gyoker);
```

```
    Console.WriteLine();
```

```
}
```

3 references

```
private void PostorderRekurziv(Csomopont csomopont)
```

```
{
```

```
    if (csomopont == null) return;
```

```
    PostorderRekurziv(csomopont.Bal);
```

```
    PostorderRekurziv(csomopont.Jobb);
```

```
    Console.Write(csomopont.Adat + " ");
```

```
}
```

```
}
```

```
internal class Program
```

```
{
```

```
    0 references
```

```
    static void Main(string[] args)
```

```
    {
```

```
        BinFa fa = new BinFa();
```

```
        // Elemszámok beszúrása
```

```
        fa.Beszuras(50);
```

```
        fa.Beszuras(30);
```

```
        fa.Beszuras(70);
```

```
        fa.Beszuras(20);
```

```
        fa.Beszuras(40);
```

```
        fa.Beszuras(60);
```

```
        fa.Beszuras(80);
```

```
        // Bejárások
```

```
        fa.InorderBejaras(); // Kiíratás: 20 30 40 50 60 70 80
```

```
        fa.PreorderBejaras(); // Kiíratás: 50 30 20 40 70 60 80
```

```
        fa.PostorderBejaras(); // Kiíratás: 20 40 30 60 80 70 50
```

```
        // Keresés
```

```
        Console.WriteLine("\nKeresés 40: " + fa.Kereses(40)); // True
```

```
        Console.WriteLine("Keresés 100: " + fa.Kereses(100)); // False
```

```
        // Elem törlése
```

```
        Console.WriteLine("\nTörlés 50 (gyökér):");
```

```
        fa.Torles(50);
```

```
        fa.InorderBejaras(); // Kiíratás: 20 30 40 60 70 80
```

```
    }
```

```
}
```