

## İkili Arama Ağacı (ekleme, Listeleme, Arama, En büyük eleman, en küçük eleman)

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Agac
{
    int eleman;
    struct Agac *sag, *sol;
}Agaclar;

Agaclar *kok=NULL;

Agaclar *DegerAl()
{
    Agaclar *yeni;
    yeni=(Agaclar *)malloc(sizeof(Agaclar));
    if(yeni==NULL)
    {
        puts("\nBellek dolu...");
        return NULL;
    }
    puts("\n Elemani Giriniz...: ");
    scanf("%d",&(yeni->eleman));
    yeni->sag=NULL;
    yeni->sol=NULL;
    return yeni;
}

void ekle(Agaclar *yeni) // döngüsel olarak ekleme
{
    Agaclar *gecici, *parent;
    if(kok==NULL) kok=yeni; // ilk eklenen eleman kök düğümüdür
    else
    {
        gecici= kok;
        while(gecici!=NULL)
        {
            parent= gecici;
            if (yeni->eleman < gecici->eleman) // küçükse
                gecici= gecici->sol;
            else // büyükse
                gecici= gecici->sag;
        }
        if (yeni->eleman < parent->eleman)
            parent->sol= yeni;
        else
            parent->sag= yeni;
    }
}

void ekle_rec(Agaclar *agackok, Agaclar *yeni) // recursive olarak ekleme
{
    if(kok==NULL) // ilk eklenen eleman kök düğümüdür
        kok= yeni;
    else
    {
        if(yeni->eleman<agackok->eleman) // küçükse
        {
            if(agackok->sol==NULL) // sol link boş ise ekle
                agackok->sol= yeni;
            else // sol link dolu ise aynı fonksiyonu sol link ile çağır
                ekle_rec(agackok->sol, yeni);
        }
        else // büyükse
        {
            if(agackok->sag==NULL) // sağ link boş ise ekle
                agackok->sag= yeni;
            else // sağ link dolu ise aynı fonksiyonu sağ link ile çağır
                ekle_rec(agackok->sag, yeni);
        }
    }
}
```

```

Agaclar *arama(Agaclar *kok, int aranan)
{
    if (kok==NULL)
        return NULL;    // başarısız arama
    if (aranan<kok->eleman)
        return arama(kok->sol,aranan);
    else if (aranan>kok->eleman)
        return arama(kok->sag,aranan);
    else return kok;    // başarılı arama
}

Agaclar * enkucuk(Agaclar *kok)
{
    if (kok->sol==NULL)
        return kok;
    return enkucuk(kok->sol);
}

Agaclar * enbuyuk(Agaclar *kok)
{
    if (kok->sag==NULL)
        return kok;
    return enbuyuk(kok->sag);
}

void inlistele(Agaclar *kok)
{
    if(kok!=NULL)
    {
        inlistele(kok->sol);
        printf("\n %d",kok->eleman);
        inlistele(kok->sag);
    }
}

void main()
{
    Agaclar *yeni,*bul,*ekucuk,*ebuyuk;
    int eleman, aranan,sil,tut;
    char secim;
    while(1)
    {
        clrscr();
        puts("\n1- Ekleme\n2- InOrder Listeleme\n3- Arama\n4- Silme\n5-
Enkucuk\n6- Enbuyuk\n7-Cikis\nSeciminiz...: ");
        secim=getch();
        switch(secim)
        {
            case '1': yeni=DegerAl();
                        ekle_rec(kok,yeni);
                        break;
            case '2': if(kok!=NULL)
                        {
                            inlistele(kok);
                            getch();
                        }
                        else
                            puts("\nBos Liste...");
                        break;
            case '3': printf("\n Aranan eleman");
                        scanf("%d",&aranan);
                        if(kok!=NULL)
                        {
                            bul=arama(kok,aranan);
                            if (bul!=NULL)
                                printf("Bulunan eleman=%d",bul->eleman);
                            else printf("eleman bulunamadi");
                        }
                        getch();

```

```

    }
    else
        puts("\nBos Liste...");
    break;
    case '4': if(kok!=NULL)
    {      ekucuk=enkucuk(kok);
            printf("\n Agacin En kucuk
Elemani=%d",ekucuk->eleman);

            getch();
        }
    else
        puts("\nBos Liste...");
        break;
    case '5': if(kok!=NULL)
    {      ebuyuk=enbuyuk(kok);
            printf("\n Agacin En kucuk
Elemani=%d",ebuyuk->eleman);

            getch();
        }
    else
        puts("\nBos Liste...");
        break;
    case '6':      exit(0);
    default: puts("\nYanlis Secim...");
}
}
}

```