VERİ ANALİZİ ARASINAV - (11 Nisan 2011)

SORU 1) Her bir düğümünde tamsayı değerler bulunan bir bağlı liste bilgisayar belleğinde bulunmaktadır (ilk dugümün işaretcisi biliniyor). Bu listedeki tek değere sahip düğümlerin değerleri toplamını yeni bir düğüm olarak listenin başına, çift değere sahip düğümlerin değerler toplamını yeni bir düğüm olarak listeye sondan bir önceki düğüm olarak ekleyen C fonksiyonu yazınız.

SORU 2) 48, 125, 66, 90, 5, 56, 77, 16, 100, 200, 60, 120 değerlerini kullanarak

a) Dengeli ikili arama ağacı oluşturunuz.
b) Heap oluşturunuz ve kökünü siliniz.

SORU 3)

a) A^(B/C-D)-E+F/G/L*D ifadesini yığt yapısı kullanarak postfiks gösteriliminde yazınız.

b) Bir ikili arama ağacında ortalama başarılı ve başarısız karşılaştırma sayılarını tanımlayınız. En kötü durumda başarılı karşılaştırma sayısını belirleyiniz.

SORU 4) Arka sayfada verilen C programınında önce

48, 22, 90, **k**, 75, 17, 36, **m**, 400, 175, **n** , 11, 43

değerleri ile bir ikili arama ağacı oluşturuluyor (programın bu parçası verilmemiştir). Daha sonraki adımları açıklayarak oluşturulan bu ağaç için yazdırılacak değerleri belirleyiniz.

k: Numaramızdaki rakamları tolamı.

m: Adınızın ilk harfinin afabetik sıralamadaki yeri.

n: Soyadınızın ilk harfinin alfabetik sıralamadaki yeri.

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
typedef struct agac {
   int bilgi;
   struct agac *sol;
   struct agac *sag;
 }AGAC;
 int top = 0;
int main()
{    AGAC* kokptr  =  NULL;
      int sag_top = 0, sol_top = 0, toplam;
   AGAC* agac_olustur(void);
     void bak( AGAC* );
     int boz( AGAC* );
kokptr =  agac_olustur();
      bak( kokptr -> sol );
      sol_top  =  top;
      top  =  0;
      bak( kokptr  ->  sag );
      sag_top  =  top;
 printf("\n  Soltop  =  %d,   Sagtop  =  %d", sol_top,
sag_top);
 toplam = boz( kokptr -> sol )+ boz( kokptr -> sag );
     printf("\n toplam = %d",toplam);
       return 0;}
 void bak( AGAC *ptr )
{
   if( ptr != NULL )
    {
      bak( ptr -> sol );
      top=top + ptr->bilgi;
      bak( ptr -> sag );
     }
}
/*****************************/
int boz( AGAC *ptr )
{   AGAC* temp = ptr;
    int toplam = temp -> bilgi;
    while( temp -> sol != NULL )
     temp = temp -> sol;
      toplam+= temp -> bilgi;
      while( ptr -> sag != NULL )
     ptr = ptr -> sag;
     toplam+= ptr -> bilgi;
    return ( toplam );
}
```