## VERI ANALIZI FINAL SINAV (28 Mayıs 2008)

SORU 1) Bir tek yönlü bağlı liste bilgisayar belleğinde bulunmaktadır (ilk düğümün işaretçisi biliniyor). Herbir düğüm tamsayı değerler içermektedir. Listedeki düğüm sayısını ve düğümlerdeki değerlerin toplamını bulan, düğüm sayısı tek ise bu toplamı yeni bir düğüm olarak listenin başına çift ise sonuna ekleyen C fonksiyonu yazınız.

SORU 2) Arka sayfada verilen C programını adım adım açıklayarak yazdırılan değerleri belirleyiniz.

SORU 3) 60, 120, 48, 12, 210, 83, 10, 30, 400, 38, 5, 8

Anahtar değerlerini sırasıyla kullanarak

a) AVL ağacı oluşturunuz.

b) Yeniden sıralanmış hash tablosu oluşturunuz.

c) Heap oluşturunuz ve kökünü siliniz.

SORU 4) Şekilde verilen graf için V1 düğümünden başlayarak tüm düğümlere olan en kısa yolları Dijkstra algoritması ile bulunuz.

SORU 5)

n düğümden oluşan bir ikili arama ağacında I toplam içyol uzunluğunu göstermek üzere ortalama başarılı karşılaştırma sayısı  $C_n = (2I+n)/n$  olarak verilir. Bu sayının en kötü dueumda  $C_n = 2lg(n)$  -3 ve en iyi durumda  $C_n = n$  olacağını gösteriniz.

```
#include<stdio.h>
#include<conio.h>
 #define DS 4
 #define MXEL 4
 int DUG[DS];
 int KOM[DS][DS] = \{0,0,1,1,
 0,0,1,0,
  1,0,0,0,
  0,1,1,0};
  int KUY[DS];
  int arka = 0, on = 0, sayac = 0;
 int main()
   void ekle( int );
   int sil( void );
   int i, j, k;
   for(j = 0; j < DS; j++)
    DUG[i] = -1;
  i = 3;
   DUG[i] = 1;
   while (i!=-1)
     printf( " %d ", i + 1 );
     for(k = 0; k < DS; k++)
       if(KOM[i][k] != 0 && DUG[k] == -1)
         ekle(k);
         DUG[k] = 0;
     i = sil();
     DUG[i] = 1;
```

```
/************/
void ekle( int k )
{
    KUY[ arka ] = k;
    arka = ++arka % MXEL;
    ++sayac;
}
/***************
int sil()
{
    int j;
    if( sayac == 0 )
        return(-1 );
    j = KUY[ on ];
    on = ++on % MXEL;
    --sayac;
    return(j);
}
```

Barrane Colon Ruser Scried