Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

BIL 205 Veri Yapıları

Bütünleme Sınavı, 2.2.2012, 10:00

Süre : **90** Dakika

CEVAPLAR

```
if(p->left == NULL && p->right != NULL)
{
    if(parent->left == p)
        parent->left  = p->right;
    else
        parent->right = p->right;

    p->right = NULL;

    delete p;
}

if( p->left != NULL && p->right == NULL)
{
    if(parent->left == p)
        parent->left  = p->left;
    else
        parent->right  = p->left;

    p->left = NULL;

    delete p;
}
```

1. Yukarıdaki kodda boş bırakılan yerleri doldurunuz. (20P)

2. Dairesel (circularly) bağlı liste, kuyruk (queue) olarak kullanılırken veri silme kodu neden aynıdır? (30P)

Dairesel bağlı liste kuyruk olarak kullanıldığında eleman eklendikçe cursor advance() yapılarak hep son eklenen elemana işaret eder. Son elemanın next'i de ilk elemandır.

Dairesel bağlı listede cursor->next silindiğinden dairesel bağlı liste kuyruk olarak kullanıldığında da kuyruk mantığına uygun olarak ilk eklenen eleman silinecektir.

```
void addFront(const string& e)
{
   add(header->next, e);
}

void addBack(const string& e)
{
   add(trailer, e);
}

void add(DoublyNode* v, const string& e)
{
   DoublyNode* u    = new DoublyNode;
   u->elem          = e;
   u->next          = v;
   u->prev          = v->prev;
   v->prev->next    = u;
   v->prev          = u;
}
```

3. Çift yönlü bağlı listeye veri ekleyen yukarıda kodda addFront() fonksiyonu add()'i **header->next** ile çağırdığı halde addBack() neden **trailer->prev** değil de sadece **trailer** ile çağırmıştır? (20P)

add(), e'yi v'den önce ekler. Listenin başına eklerken header->next ile ilk elemanı parametre olarak veririz ki ondan önce eklesin. Listenin sonuna eklerken trailer dememizin nedeni **trailerdan önce** yani **son elemandan sonra** eklediğimiz içindir. Eğer trailer->prev derseydik son elemandan **önce** eklerdi.

8 7 6 5 4 3 2 1 9

4. Yukarıdaki verileri splay ağacına yerleştiriniz? (30P)