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

BIL 2001 Veri Yapıları
Bütünleme, 24.01.2020, 13:20
Süre : 61 Dakika

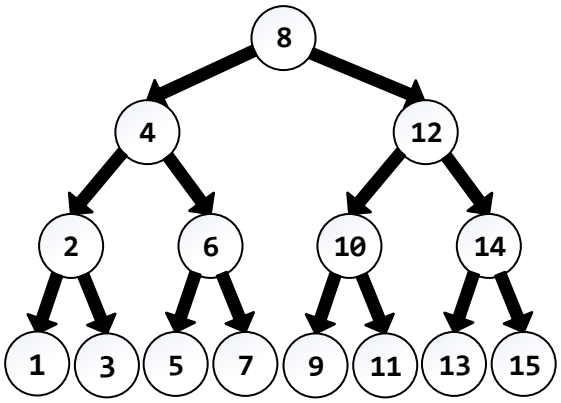
CEVAPLAR

```
void traverse(TreeNode* v)
{
     if (v->left != NULL)
     {
          traverse(v->left);
          cout << v->elem << " ";
     }

     if (v->right != NULL)
     {
          cout << v->elem << " ";
          traverse(v->right);
     }
}
```

1. **main()**'de aşağıdaki ağacın **root**u ile çağrıldığı varsayılan **traverse()** fonksiyonunun çıktısı nedir? **(40P)**

2	2	4	4	6	6	8	8	10	10	12	12	14	14



8 7 6 5 4 3 2 1

2. Yukarıdaki verilerin ikili ağaca eklendiği varsayılsın. Bu ağacın `inorder`, `preorder` ve `postorder` gezinme çıktılarından hangisi diğer ikisinden farklıdır? **(30P)**

*Yanlış cevaptan **5P** kırılacaktır.*

(A) inorder

(B) preorder

(C) postorder

```
void insertOrdered(string& e, int& i)
{
  DoublyNode* newNode      = new DoublyNode;
  newNode->elem            = e;
  newNode->score           = i;

  DoublyNode* current      = header;

  while (current->next != trailer)
  {
     if (newNode->score >= current->next->score)
       current = current->next;
     else
       break;
  }

  newNode->next     = current->next;
  newNode->prev     = current;
  ..............    = ..............;
  ..............    = ..............;
}
```

3. **`insertOrdered()`** fonksiyonunu tamamlayınız. **(30P)**

*Yanlış cevaptan **5P** kırılacaktır.*

(A)
```
newNode->next->prev     = newNode;
current->next->prev     = newNode;
```

(B)
```
newNode->prev->next     = newNode;
current->next           = newNode;
```

(C)
```
current->next->prev     = newNode;
newNode->prev->next     = newNode;
```

(D)
```
current->next           = newNode;
current->next->prev     = newNode;
```

(E)
```
newNode->prev->next     = newNode;
current->next->prev     = newNode;
```