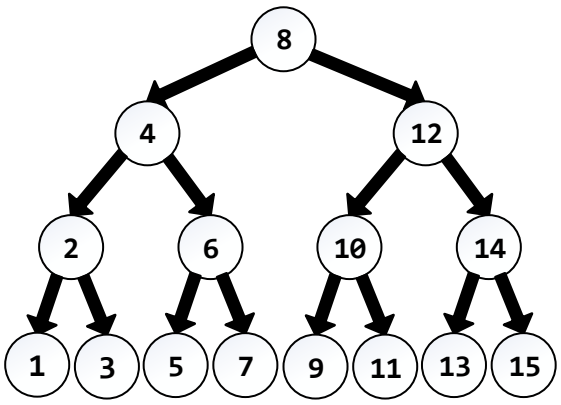
Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

BIL 2001 Veri Yapıları
Bütünleme Sınavı, 27.01.2017, 15:00, D-2
Süre : **100** Dakika

CEVAPLAR

```
void traverse1(Node* v)
{
  if (v->left != NULL)       traverse1(v->left);
  else                       cout << v->elt << " ";
  if (v->right != NULL)      traverse1(v->right);
}

void traverse2(Node* v)
{
  if (v->left != NULL)       traverse2(v->left);
  if (v->right != NULL)      traverse2(v->right);
  else                       cout << v->elt << " ";
}

void traverse3(Node* v)
{
  if (v->left == NULL)       cout << v->elt << " ";
  else                       traverse3(v->left);
  if (v->right != NULL)      traverse3(v->right);
}

void traverse4(Node* v)
{
  if (v->left != NULL)       traverse4(v->left);
  if (v->right == NULL)      cout << v->elt << " ";
  else                       traverse4(v->right);
}
```

1. **main()**'de aşağıdaki ağacın **root**u ile çağrıldığı varsayılan yukarıdaki fonksiyonların çıktıları nelerdir? **(20P)**



traverse1: 1 3 5 7 9 11 13 15

traverse2: 1 3 5 7 9 11 13 15

traverse3: 1 3 5 7 9 11 13 15

traverse4: 1 3 5 7 9 11 13 15

```
void traverse(Node* v)
{
   stack<Node*> stl_stack;
   Node* current = v;

   while (true)
   {
       if (current != NULL)
       {
              stl_stack.push(current);
              current = current->left;
       }
       else
       {
              if (stl_stack.empty())
              {
                   return;
              }
              else
              {
                   current = stl_stack.top();
                   cout << current->elt << " ";
                   stl_stack.pop();
                   current = current->right;
              }
       }
   }
}
```

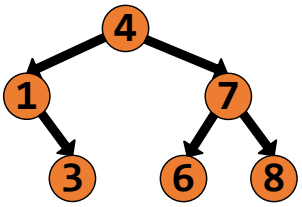
2. **main()**'de soldaki ağacın **root**u ile çağrıldığı varsayılan **traverse()** fonksiyonunun çıktısı nedir? **(30P)**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```
void removeOrdered(const string& e, const int& i)
{
  if(head == NULL)
  {
    cout << "List is empty !" << endl;  return;
  }

  if ((strcmp((char*)&head->elem, (char*)&e))
                       && (head->score == i))
  {
    SinglyNode* temp = head;
    head = head->next;
    delete temp;
    return;
  }

  SinglyNode* current = head;

  while (current->next != NULL)
  {
    if ((strcmp((char*)&current->next->elem,(char*)&e))
                       && (current->next->score == i))
    {
       SinglyNode* temp = current->next;
       current->next = current->next->next;
       delete temp;

       return;
    }

    current = current->next;
  }

  if (current->next == NULL)
   cout << "\n" << e << " is not found" << endl;
}
```

3. **removeOrdered()** fonksiyonunu tamamlayınız. **(25P)**
İpucu → Silinecek eleman **current->next** 'tir.

Zig	(X:Sol)
Zig-Zig	(X:Sağ, P:Sağ)
Zig-Zig	(X:Sol, P:Sol)
Zig	(X:Sol)
Zig-Zig	(X:Sağ, P:Sağ)
Zig-Zag	(X:Sol, P:Sağ)
Zig-Zag	(X:Sağ, P:Sol)
Zig-Zag	(X:Sağ, P:Sol)



4. Yukarıdaki işlemlerle oluşturulan Splay Ağacına verilerin hangi sırada eklendiğini bulunuz. **(25P)**

→

6	3	8	1	7	4