3.1 6%(①~④ 1%，⑤2%)

① n==1
② return 0
③ n==2
④ return 1
⑤ return (n-1)*(fun(n-1)+fun(n-2))

3.2

A.6%（each line 1.5%）

```
if L[i]<L[i+1]:
    temp=L[i]
    L[i]=L[i+1]
    L[i+1]=temp
```

B.O（n²）4%

3.3

A. Print out the elements of all nodes on the tree(or subtree) with root being t based on depth first principle. 3%

B. 3%

8
3
1
2
5
2
3
1
2

C. ①D[x]=D.get(x,0)+1

②L.append((x,D[x]))

③DFSearch(t.left,e,x+1,D,L)

④DFSearch(t.right,e,x+1,D,L) 8%（each point 2%）