

1-5 Data Structures

魏恒峰

hfwei@nju.edu.cn

2019 年 11 月 14 日

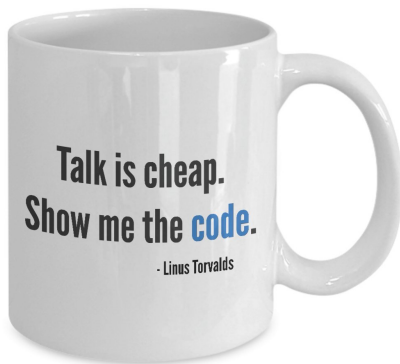




Generating All Permutations

```
1: procedure PERMS( $A[], l$ )
2:   if  $l = A.size - 1$  then
3:     print  $A$ 
4:   else
5:     for  $i \leftarrow l$  to  $A.size - 1$  do
6:       SWAP( $A[i], A[l]$ )
7:       PERMS( $A, l + 1$ )
```

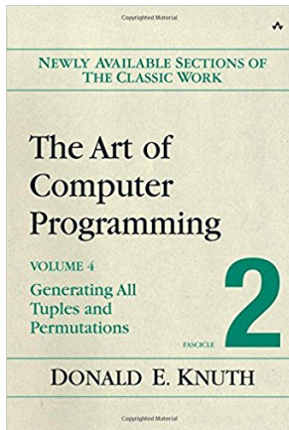
```
1: procedure PERMS( $A[], l$ )
2:   if  $l = A.size - 1$  then
3:     print  $A$ 
4:   else
5:     for  $i \leftarrow l$  to  $A.size - 1$  do
6:       SWAP( $A[i], A[l]$ )
7:       PERMS( $A, l + 1$ )
8:       SWAP( $A[i], A[l]$ )
```



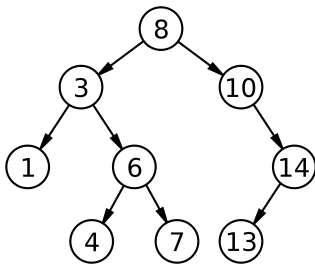


Iteration Version of PERMS

For more about “Generating All Permutations”:



Treesort Algorithm on BST



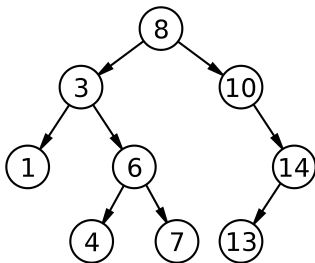
```
1: procedure BUILDBST(eles)
2:   Node root(eles[0])

3:   for all  $e \in \text{eles}[1 \dots]$  do
4:     INSERT(root, e)
```

```
1: procedure INSERT(T, e)
2:   if  $e < T.val$  then
3:     if  $T.left = \Lambda$  then
4:        $T.left = \text{new Node}(e)$ 
5:     else
6:       INSERT( $T.left$ , e)
7:   else
8:     if  $T.right = \Lambda$  then
9:        $T.right = \text{new Node}(e)$ 
10:    else
11:      INSERT( $T.right$ , e)
```

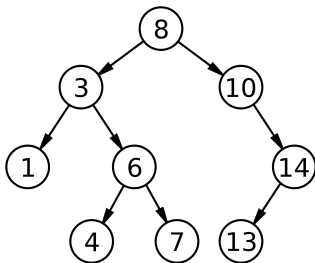
DH 2.16: Treesort

(ii) right; val; left



DH 2.16: Treesort

(ii) right; val; left



14, 13, 10, 8, 7, 6, 4, 3, 1

Thank
You!