

计算机问题求解 – 论题1-5

- 数据与数据结构

课程研讨

- DH第2章第3、4单元

数据结构

- 向量/列表 (vector/list)
- 数组/表格 (array/table)
- 队列 (queue)
- 栈 (stack)
- 树 (tree)

抽象数据类型

- 一个数学模型以及定义在此数学模型上的一组操作
- 抽象数据类型需要通过固有数据类型来实现

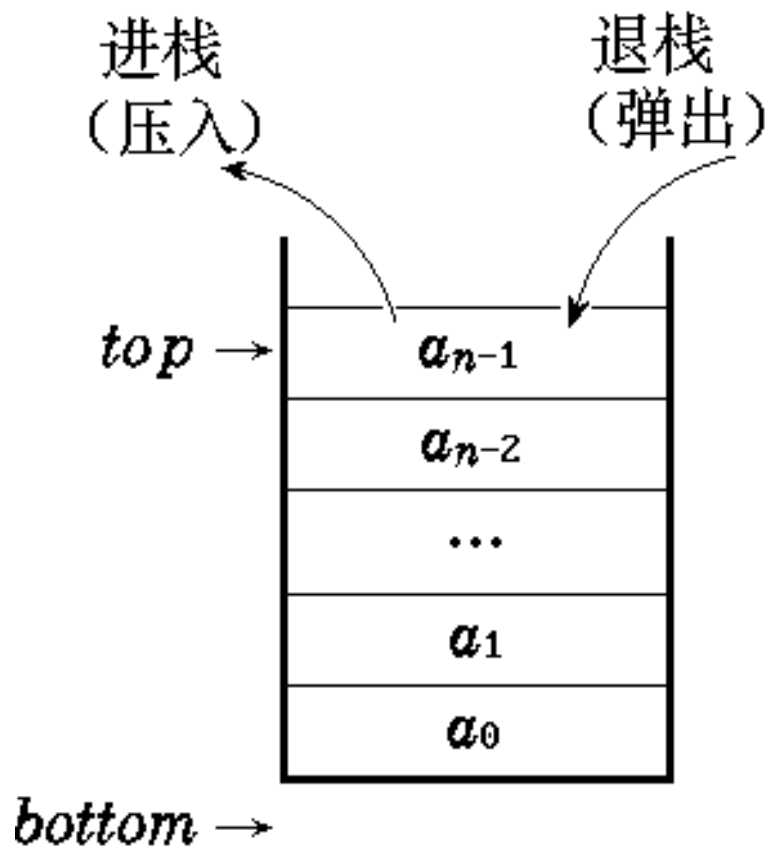
$+$
 \times

$-$
 \div

0 1 2 3
4 5
6 7 8 9

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P
Q	R	S	T
U	V	W	X
Y	Z		

栈 (Stack)



数据:

```
struct {  
    Datatype  
    int  
} stack;
```

```
data[stacksize];  
top;
```

操作:

1. Push
 2. Pop
 3. Empty
-

元素进栈
元素退栈
栈是否空

栈

- 设将整数以1、2、3、4依次进栈，但只要出栈时栈非空，则可将出栈操作按任何次序夹入其中，请回答下有问题：
 - (1) 若入栈次序为push(1), pop(), push(2), push(3), pop(), pop(), push(4), pop(), 则出栈的数字序列为什么？
 - (2) 请分析1、2、3、4的24种排列中，哪些序列可以通过相应的入出栈得到。

队列：先进先出

- 数组 $A[0..MAX-1]$
- MAX 最大允许的元素数
- 初始 $Head=0; Tail=0$;
- 新元素 e 入队 $A[Tail]=e$
- 队首元素出队 $e=A[Head]$

数据:

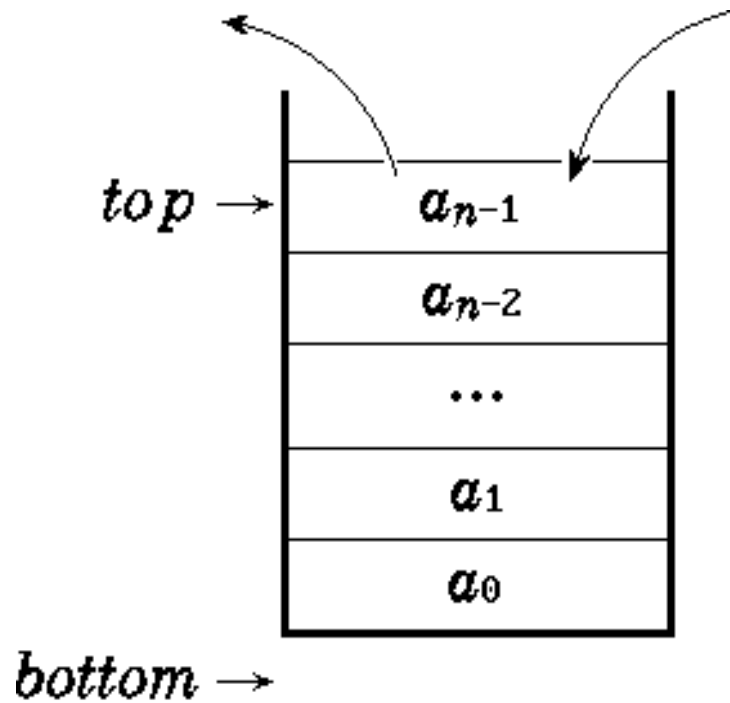
```
struct {  
    Datatype    data[size];  
    int         front;  
    int         rear;  
} stack;
```

操作:

- | | |
|------------|------|
| 1. enqueue | 元素入队 |
| 2. dequeue | 元素出队 |
| 3. Empty | 队是否空 |
| | |

栈与队列

- 如何用栈来实现队列？

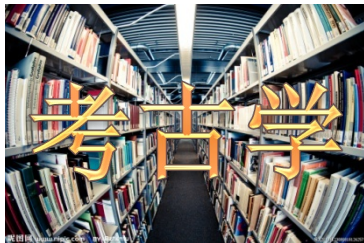
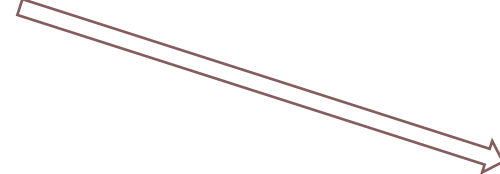
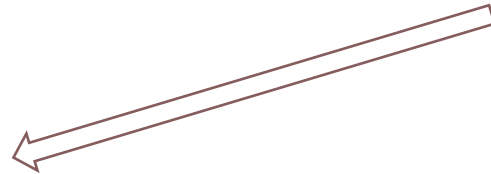
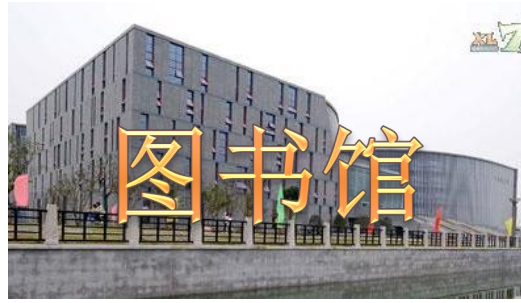


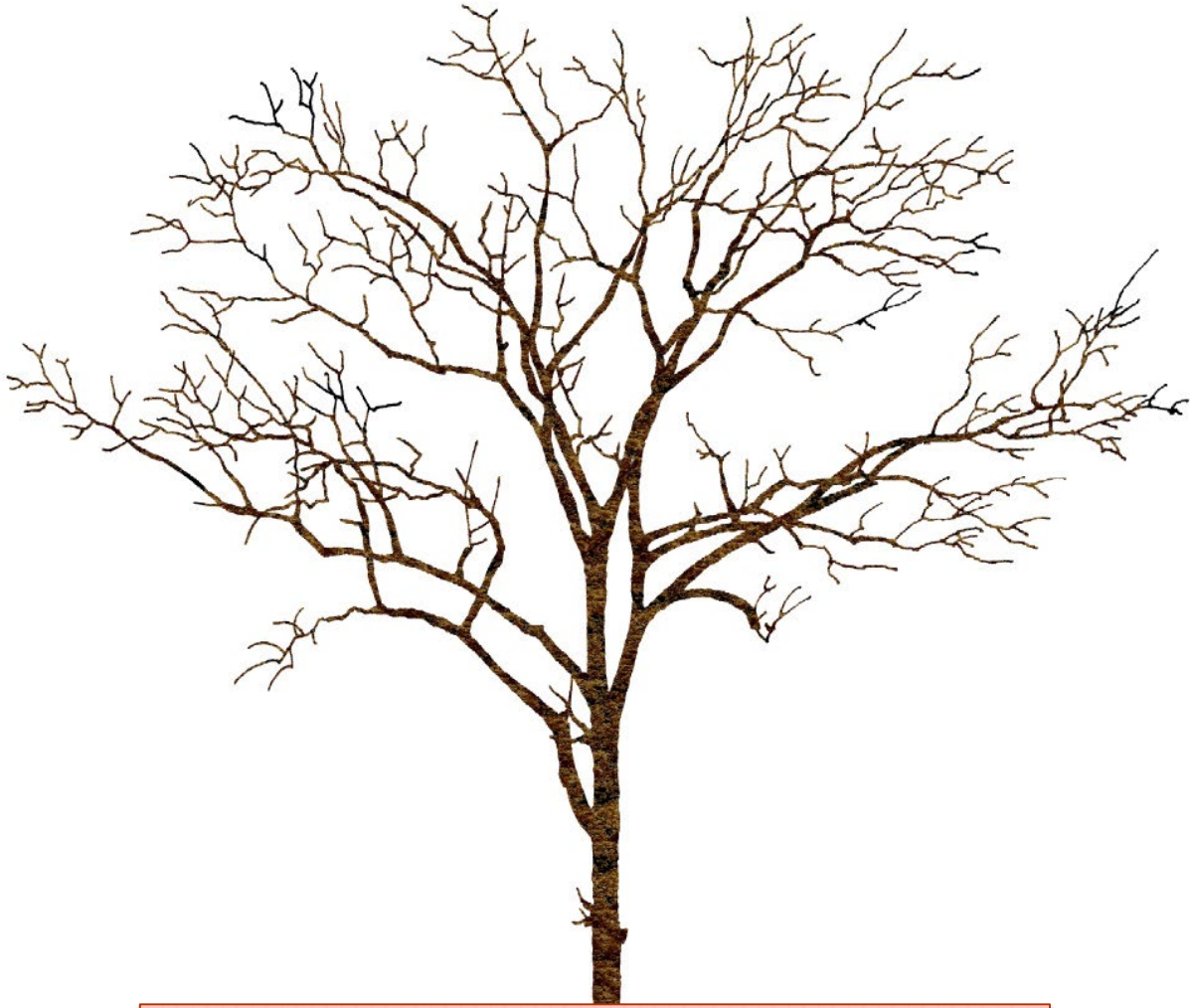
循环队列

- 循环队列的优点是什么？如何判断它的空和满？
- 用第二种方法，即少用一个元素空间的方法来区别循环队列的队空和队满，试设计置空队、判队空、判队满、出队、入队及取队头元素等六个基本操作。

思考：你想在图书馆找一些
关于中国近代史的资料

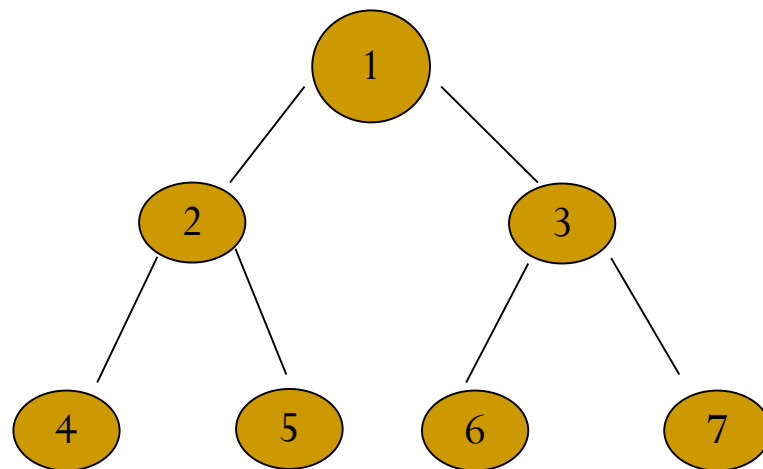
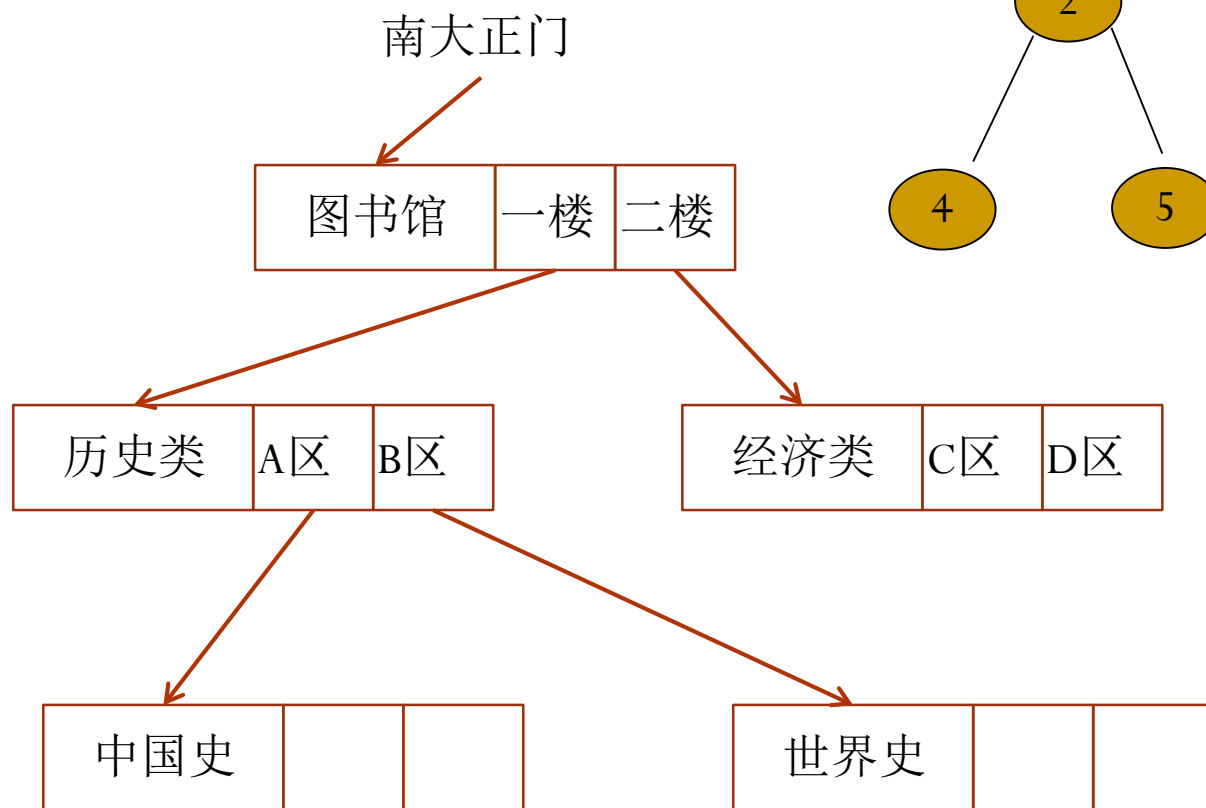




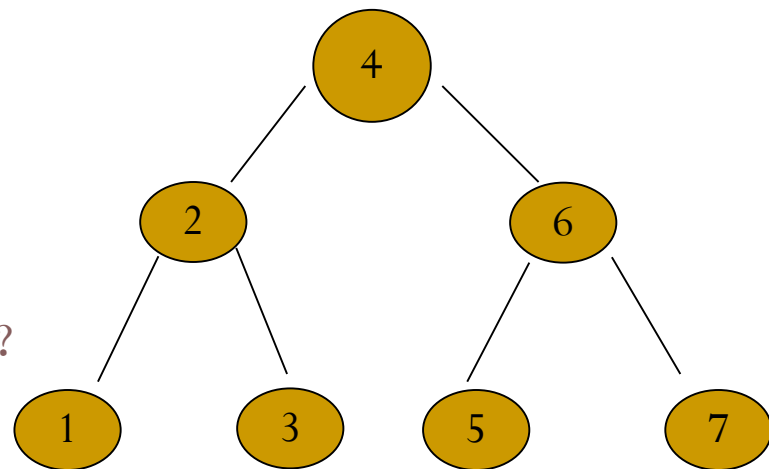
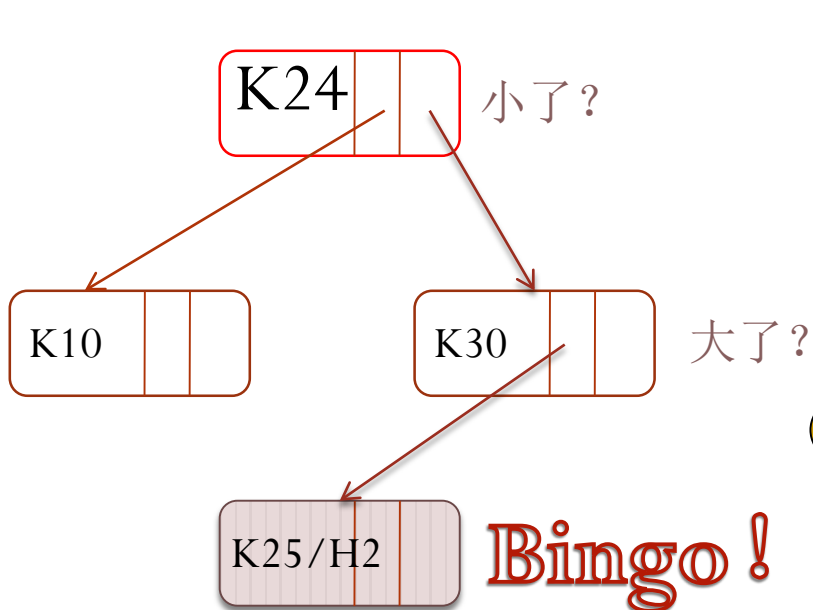
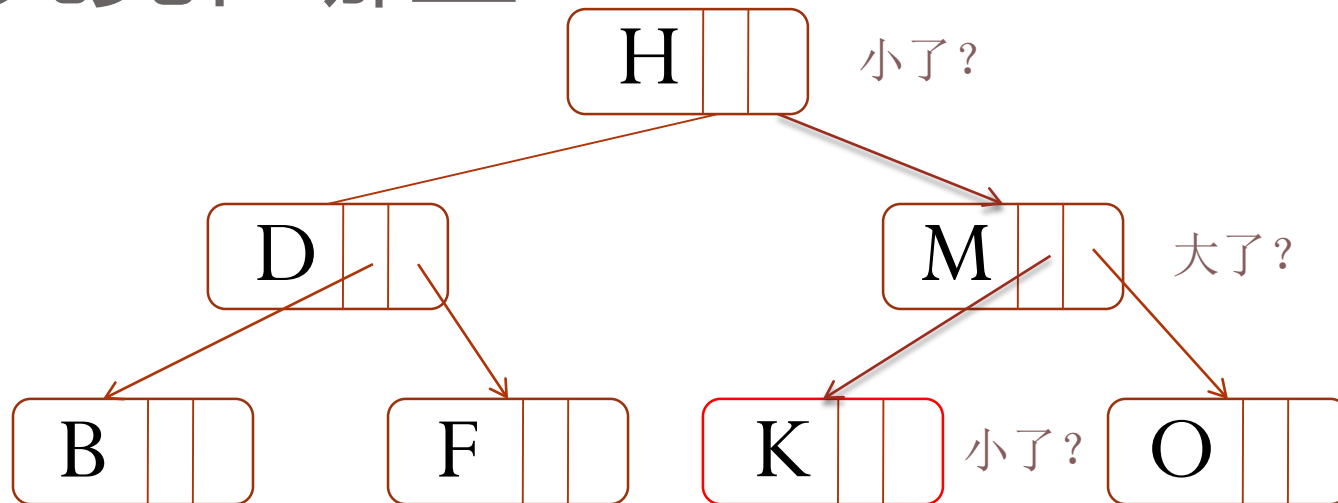


一棵倒着的树

树的数学模型



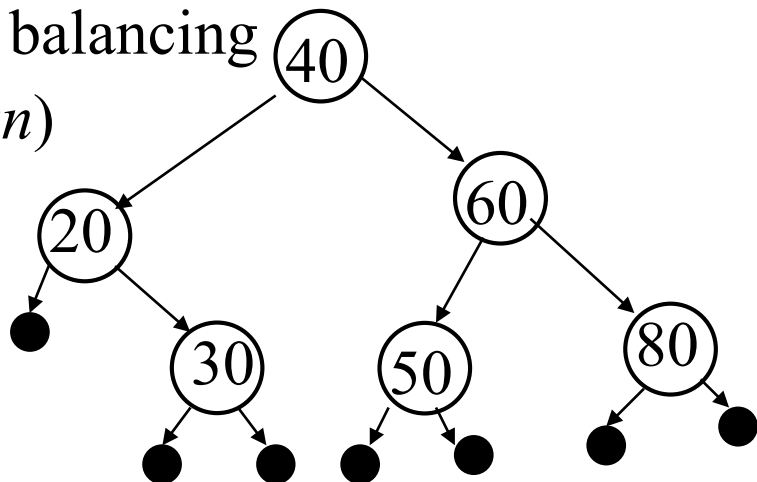
究竟在哪里? K25/H2



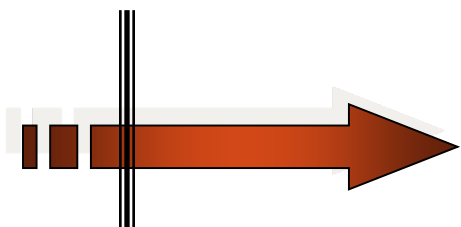
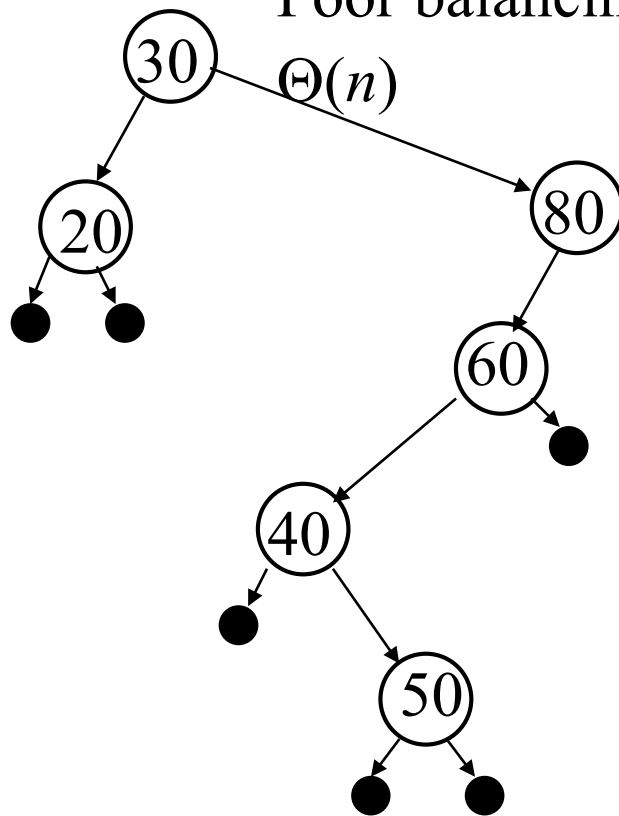
二叉搜索树

二叉搜索树

Good balancing
 $\Theta(\log n)$

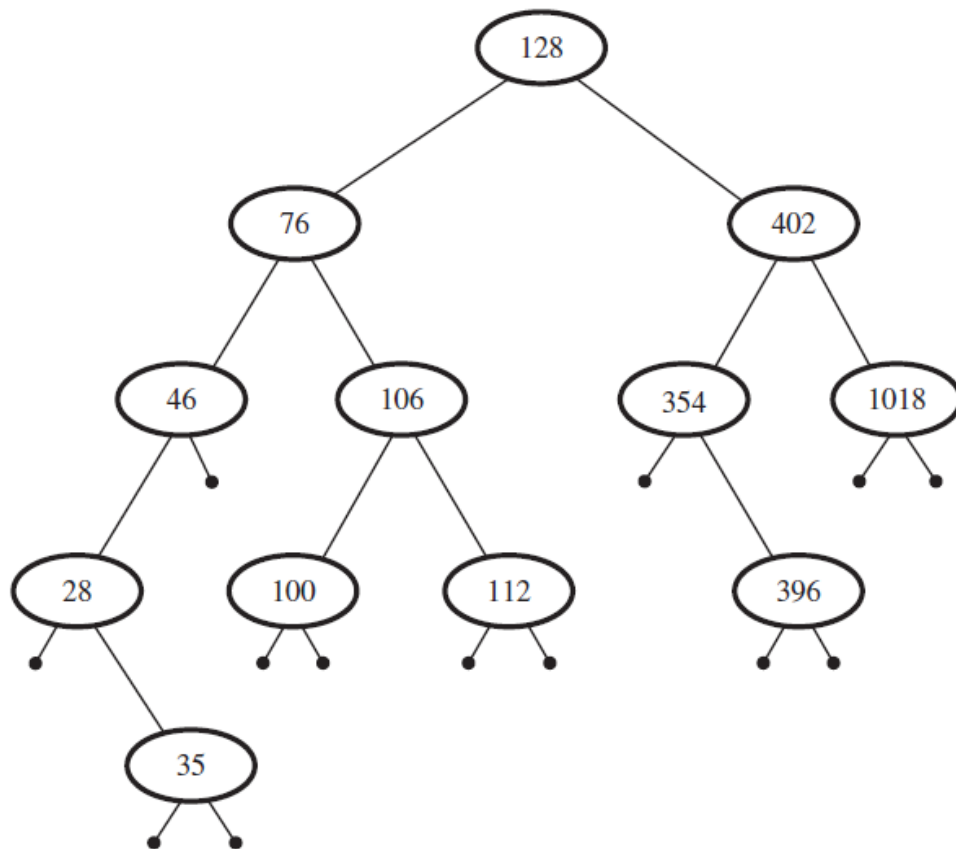


Poor balancing
 $\Theta(n)$

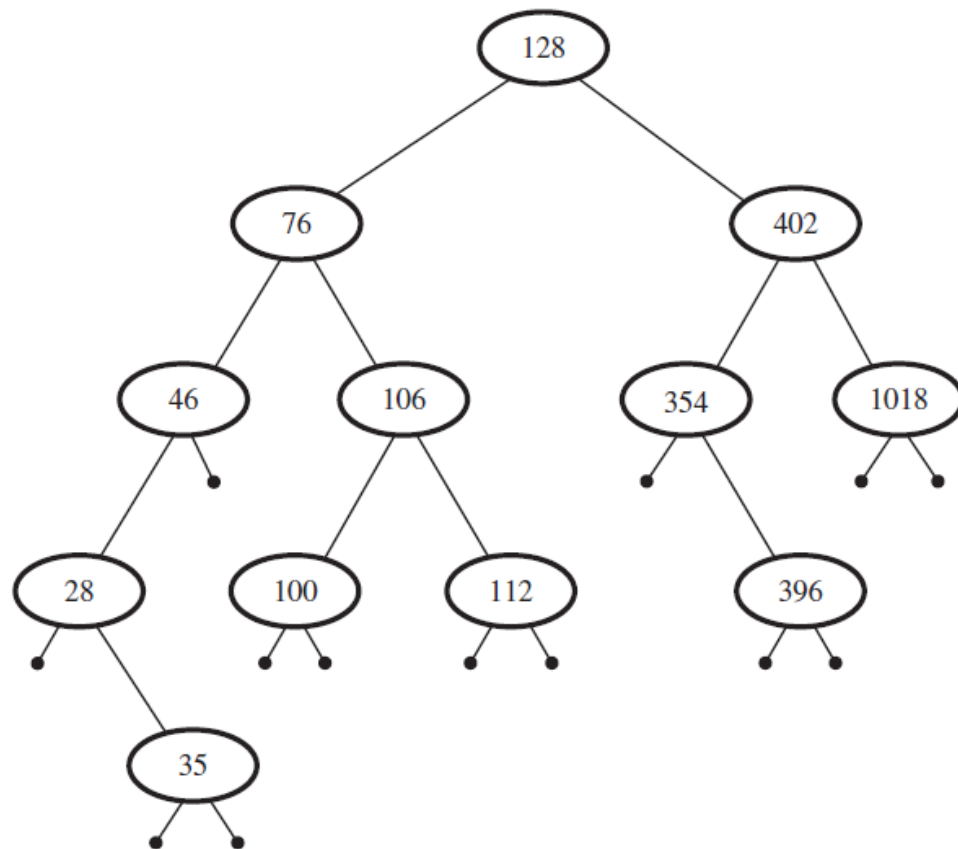


一个画的合适的树，
从左向右得到从小到大
大有序的序列

如何实现逐层输出？

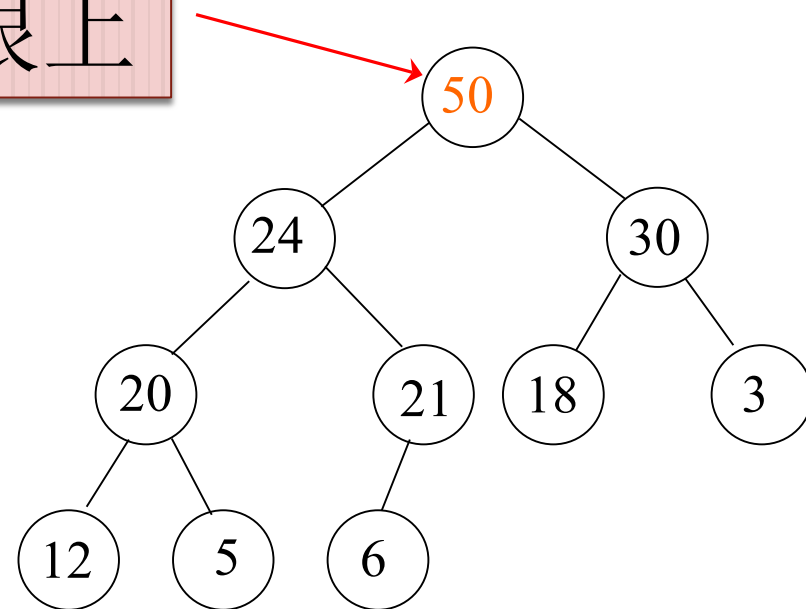
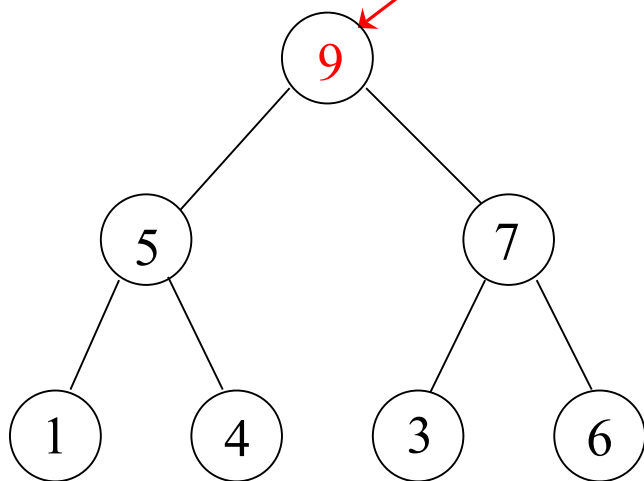


如何不用递归实现second-visit-traversal?



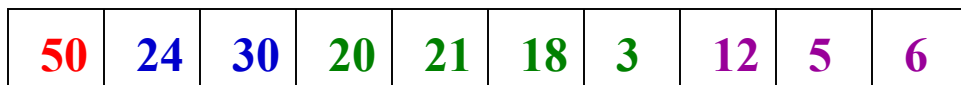
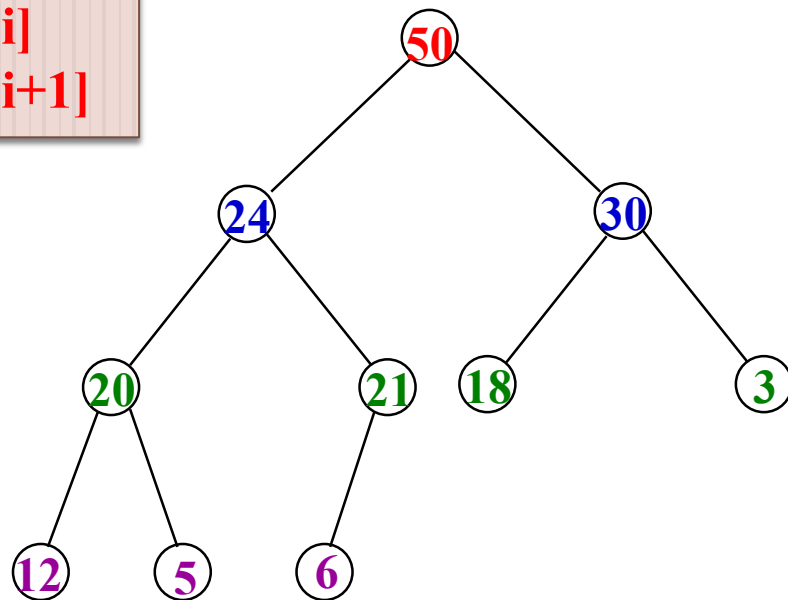
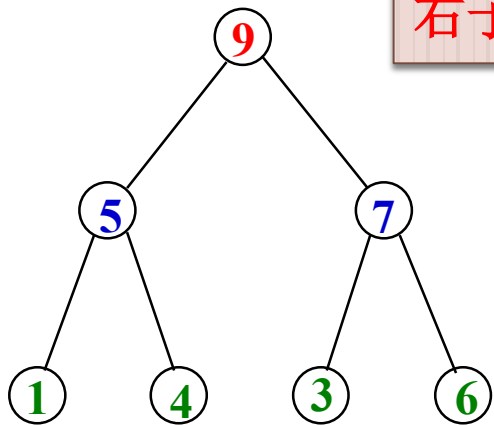
(最大) 堆

最大值总是在根上

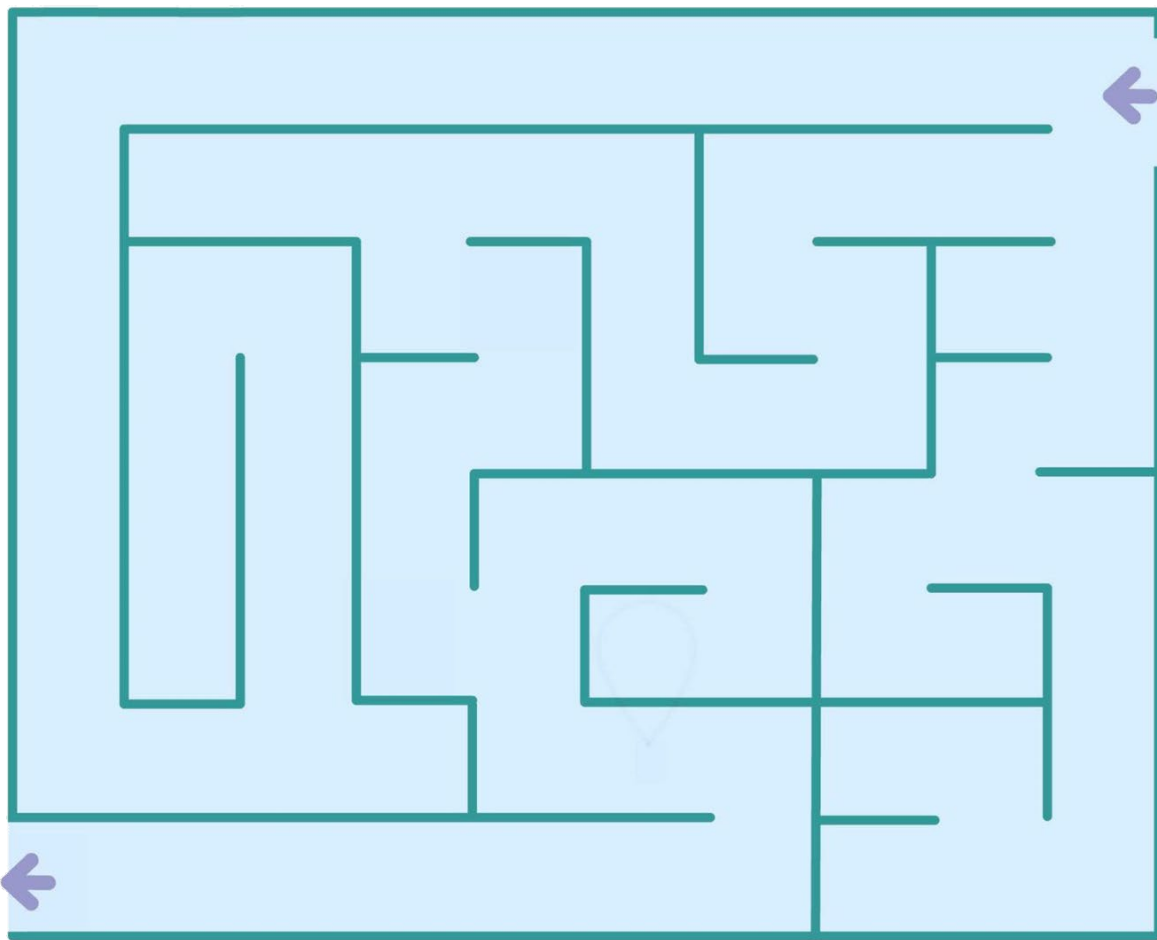


通过数组实现堆

对于元素 $E[i]$:
左子堆: $E[2i]$
右子堆: $E[2i+1]$



如何走迷宫？



如何对战Tic-tac-toe?

