数据结构与算法（一）

张铭 主讲

采用教材：张铭，王腾蛟，赵海燕 编写
高等教育出版社，2008. 6（“十一五”国家级规划教材）

http://www.jpk.pku.edu.cn/pkujpk/course/sjjg

第1章 概论

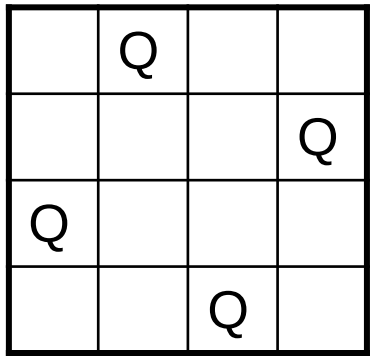
- 问题求解
- 数据结构及抽象数据类型
- **算法的特性及分类**
- 算法的效率度量

问题 —— 算法 —— 程序

目标：问题求解

- **问题（problem）**一个函数
  - 从输入到输出的一种映射
- **算法（algorithm）**一种方法
  - 对特定问题求解过程的描述，是指令的有限序列
- **程序（program）**
  - 是算法在计算机程序设计语言中的实现

算法的特性

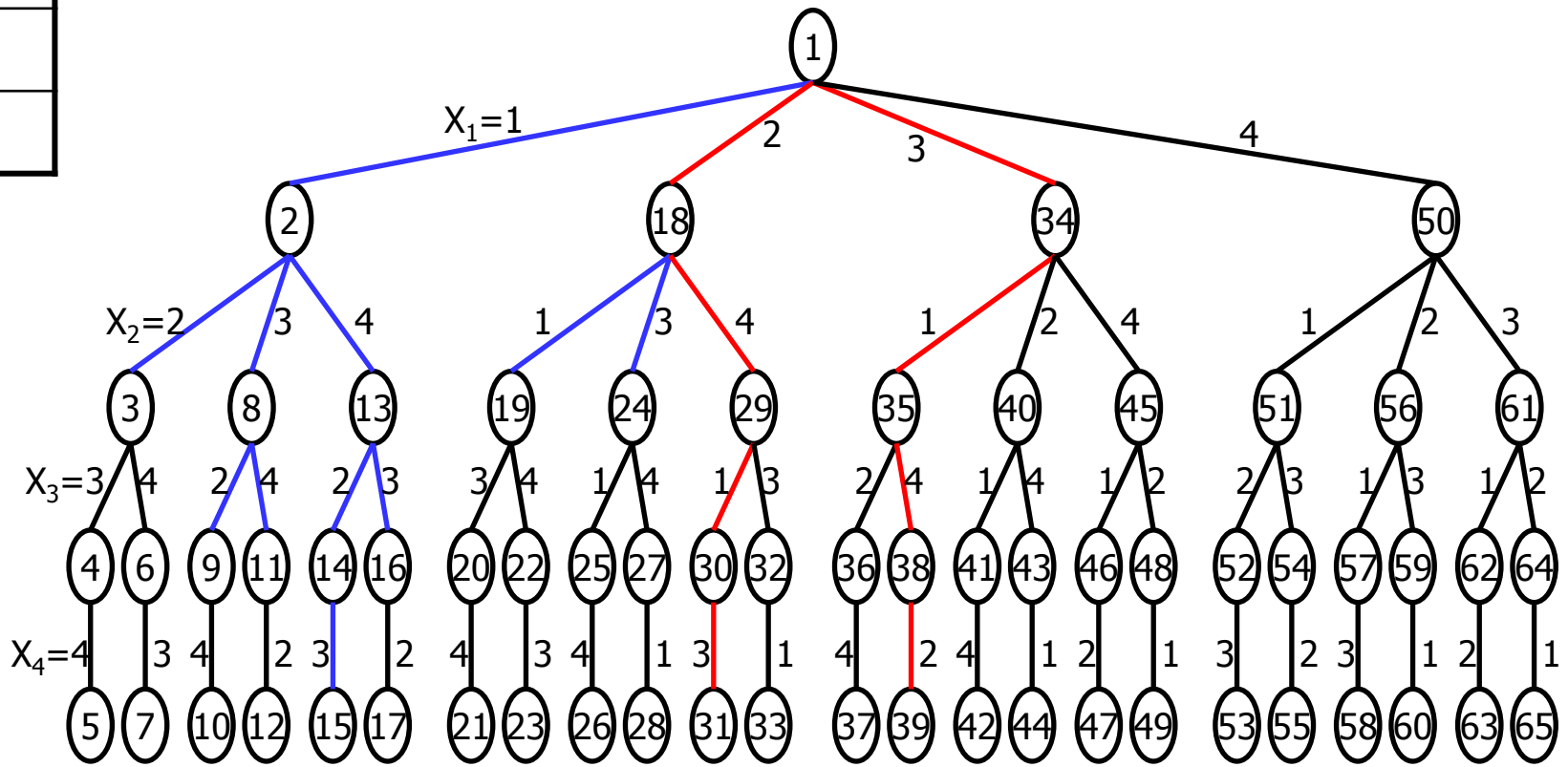


- 通用性
  - 对参数化输入进行问题求解
  - 保证计算结果的正确性
- 有效性
  - 算法是有限条指令组成的指令序列
  - 即由一系列具体步骤组成
- 确定性
  - 算法描述中的下一步应执行的步骤必须明确
- 有穷性
  - 算法的执行必须在有限步内结束
  - 换句话说，算法不能含有死循环

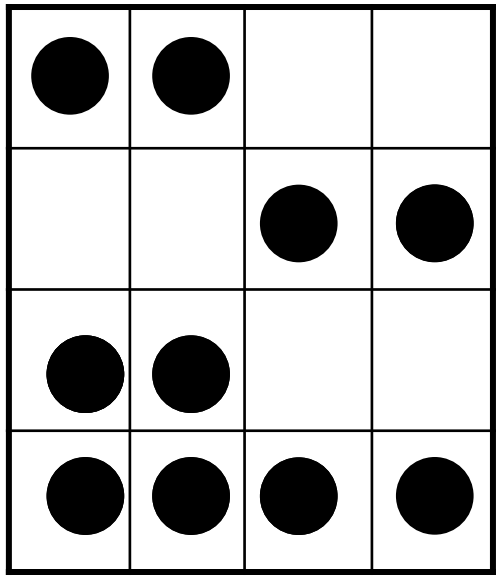
皇后问题（四皇后）

	Q		
			Q
Q			
		Q	

- 解<$x1,x2,x3,x4$>（放置列号）
- 搜索空间：4叉树（排列树）



基本算法分类



- **穷举法**
  - 顺序找 K 值
- **回溯、搜索**
  - 八皇后、树和图遍历
- **递归分治**
  - 二分找 K 值、快速排序、归并排序
- **贪心法**
  - Huffman 编码树、最短路 Dijkstra 算法、最小生成树 Prim 算法
- **动态规划**
  - 最短路 Floyd 算法

1.3 算法

顺序找k值

0	1	2	3	4	5	6	7	8
	17	35	22	18	93	60	88	52

```
template  <class Type>
class Item {
private:
    Type  key;                                  // 关键码域
                                                // 其它域
public:
    Item(Type value):key(value) {}
    Type getKey() {return key;}                 // 取关键码值
    void setKey(Type k){ key=k;}                // 置关键码
};
vector<Item<Type>*>  dataList;
template  <class Type>  int SeqSearch(vector<Item<Type>*>& dataList, int length,
Type k) {
    int i=length;
    dataList[0]->setKey (k);                    // 将第0个元素设为待检索值，设监视哨
    while(dataList[i]->getKey()!=k) i--;
    return i;                                   // 返回元素位置
}
```

二分法找 k 值

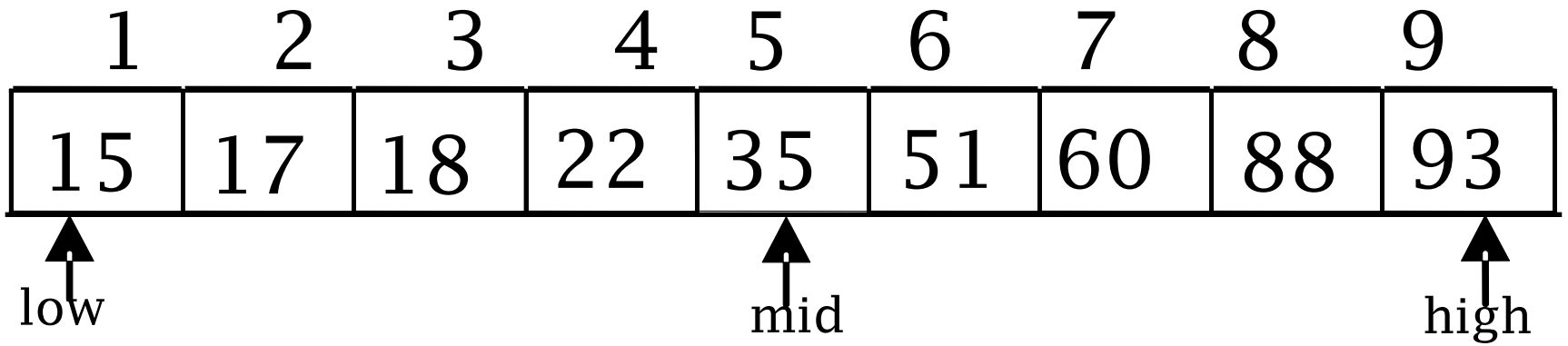
对于已排序顺序线性表

- 数组中间位置的元素值 k_{mid}
  - 如果 $k_{mid} = k$，那么检索工作就完成了
  - 当 $k_{mid} > k$ 时，检索继续在前半部分进行
  - 相反地，若 $k_{mid} < k$，就可以忽略 mid 以前的那部分，检索继续在后半部分进行
- 快速
  - $k_{mid} = k$ 结束
  - $K_{mid} \neq k$ 起码缩小了一半的检索范围

二分法找 k 值

```
template  <class Type> int BinSearch (vector<Item<Type>*>& dataList,
int length, Type k){
    int low=1, high=length, mid;
    while (low<=high)  {
        mid=(low+high)/2;
        if (k<dataList[mid]->getKey())
                high = mid-1;          // 右缩检索区间
        else if (k>dataList[mid]->getKey())
                low = mid+1;           // 左缩检索区间
        else return mid;               // 成功返回位置
     }
    return 0;                          // 检索失败，返回0
}
```

二分法检索图示



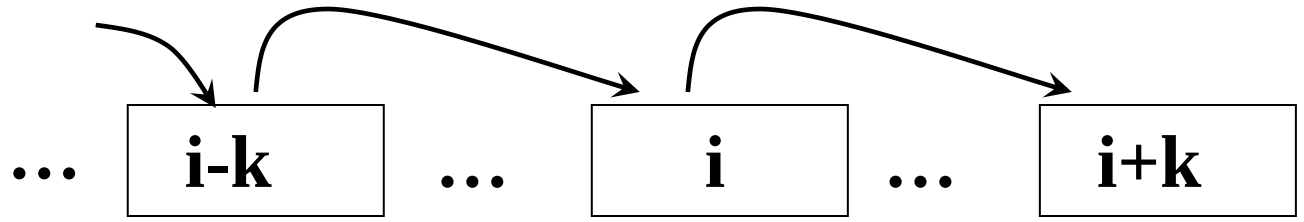
检索关键码18 low=1 high=9 K=18

第一次：mid=5; array[5]=35>18
high=4; (low=1)
第二次：mid=2; array[2]=17<18
low=3; (high=4)
第三次：mid=3; array[3]=18=18
mid=3 ; return 3

思考：算法的时空限制

设计一个算法，将数组 A(0..n-1) 中的元素循环右移 k 位，假设原数组序列为 $a_0, a_1, \ldots, a_{n-2}, a_{n-1}$；移动后的序列为 $a_{n-k}, a_{n-k+1}, \ldots, a_0, a_1, \ldots, a_{n-k-1}$。要求只用一个元素大小的附加存储，元素移动或交换次数与 n 线性相关。例如，n=10, k=3

原始数组： 0 1 2 3 4 5 6 7 8 9
右移后的： 7 8 9 0 1 2 3 4 5 6



数据结构与算法

谢谢聆听

国家精品课“数据结构与算法”
http://www.jpk.pku.edu.cn/pkujpk/course/sjjg/

张铭，王腾蛟，赵海燕
高等教育出版社，2008. 6。“十一五”国家级规划教材