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）

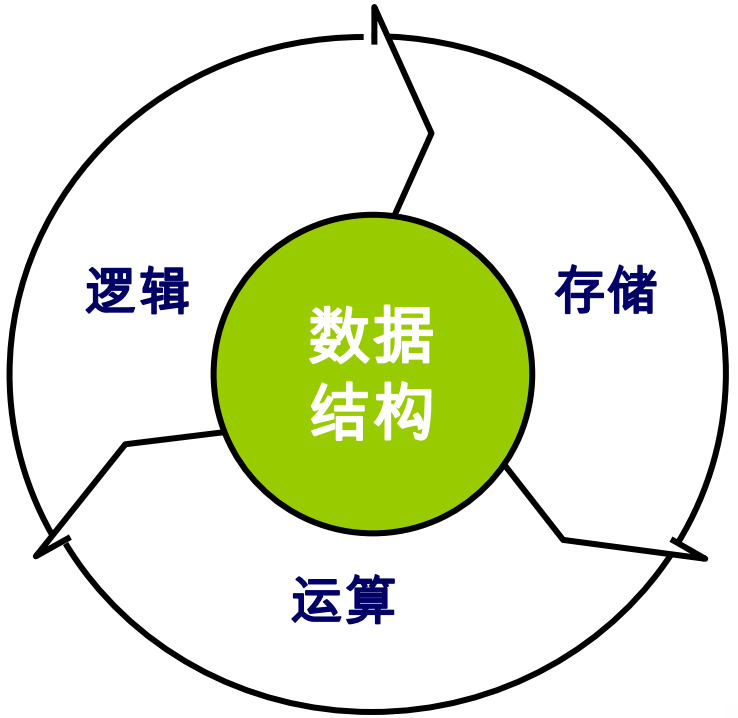
http://www.jpk.pku.edu.cn/pkujpk/course/sjjg

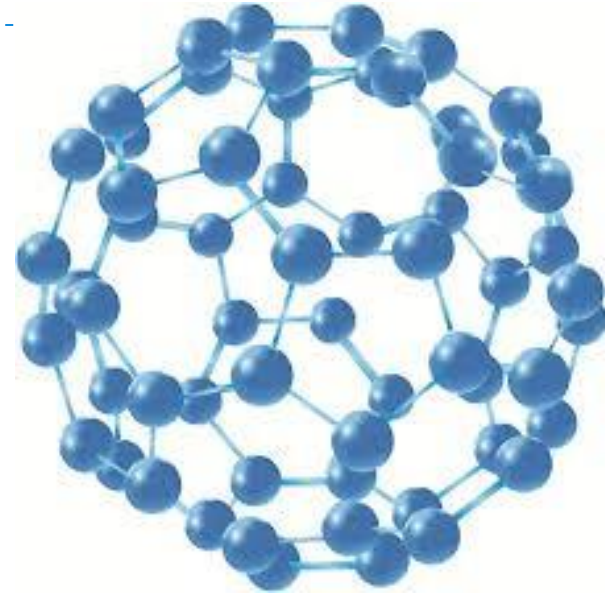
第1章　概论

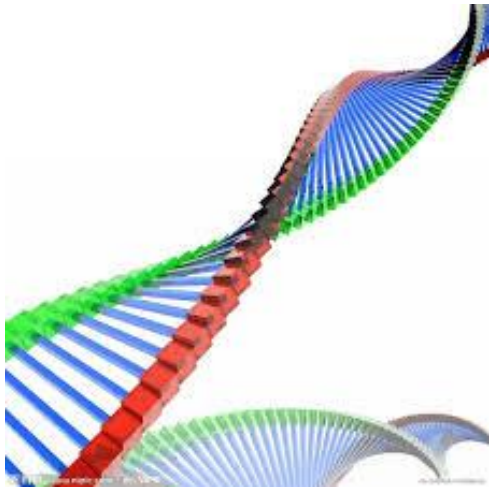
- 问题求解
- 数据结构及抽象数据类型
- 算法的特性及分类
- 算法的效率度量
- 数据结构的选择和评价

1.2 什么是数据结构

- **结构: 实体 + 关系**
- 数据结构:
  - 按照逻辑关系组织起来的一批数据,
  - 按一定的存储方法把它存储在计算机中
  - 在这些数据上定义了一个运算的集合

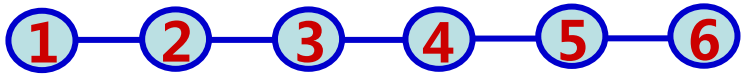




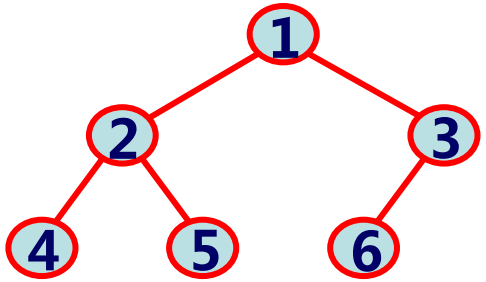


数据结构的逻辑组织

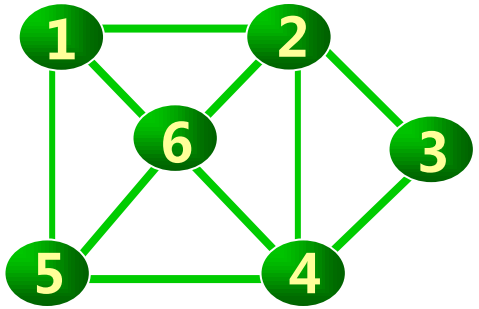
- **线性结构**
  - 线性表（表，栈，队列，串等）



- **非线性结构**
  - 树（二叉树，Huffman树，二叉检索树等）
  - 图（有向图，无向图等）

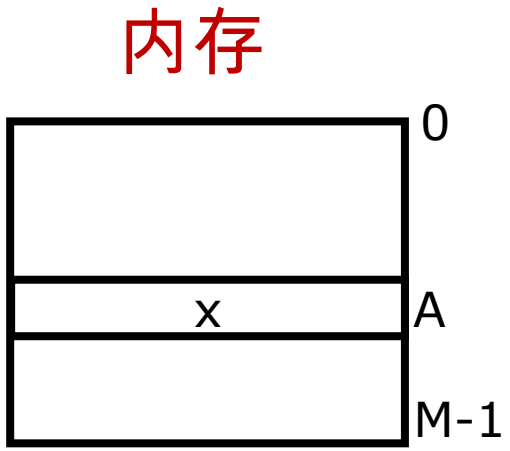


- **图 ⊇ 树 ⊇ 二叉树 ⊇ 线性表**



数据的存储结构

- **逻辑结构到物理存储空间的映射**



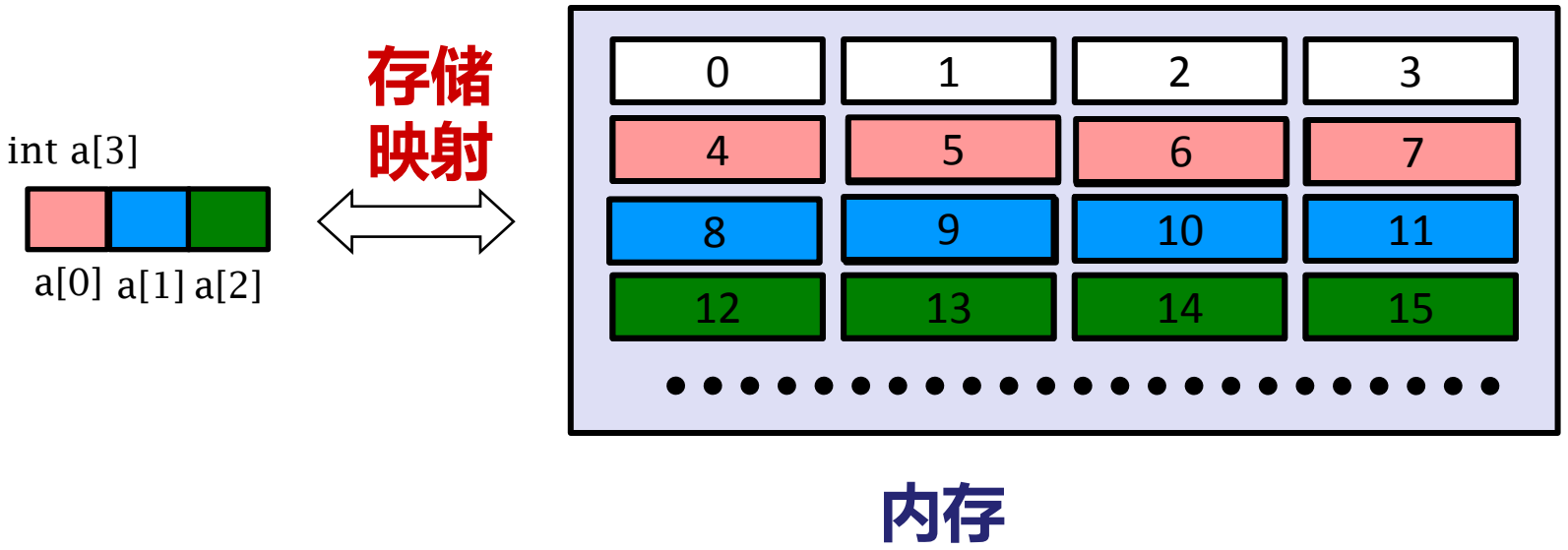
计算机主存储器（内存）

- **非负整数**地址编码，**相邻单元**的集合
  - 基本单位是字节
  - 访问不同地址所需时间基本相同（即随机访问）

1.2 什么是数据结构

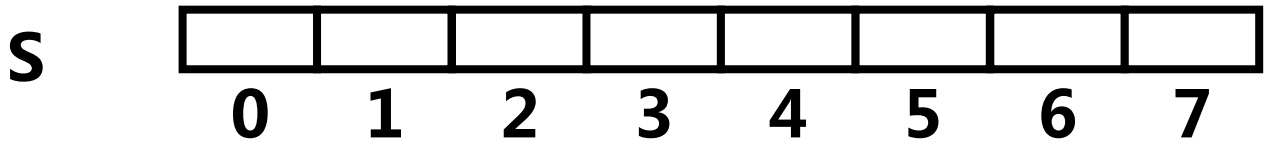
数据的存储结构

- 对逻辑结构（K ，r），其中r∈R
  - 对结点集 K 建立一个从 K 到存储器 M 的单元的映射：K→M，对于每一个结点 j∈K 都对应一个**唯一**的**连续存储**区域c ∈ M

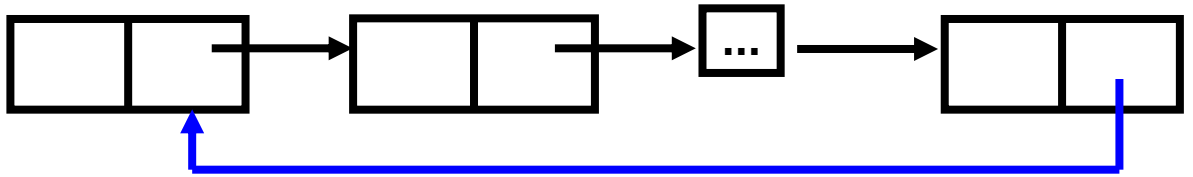


数据的存储结构

- 关系元组 $(j_1, j_2) \in r$
  (其中j_1，$j_2 \in$ K 是结点)
  - 顺序：存储单元的顺序地址



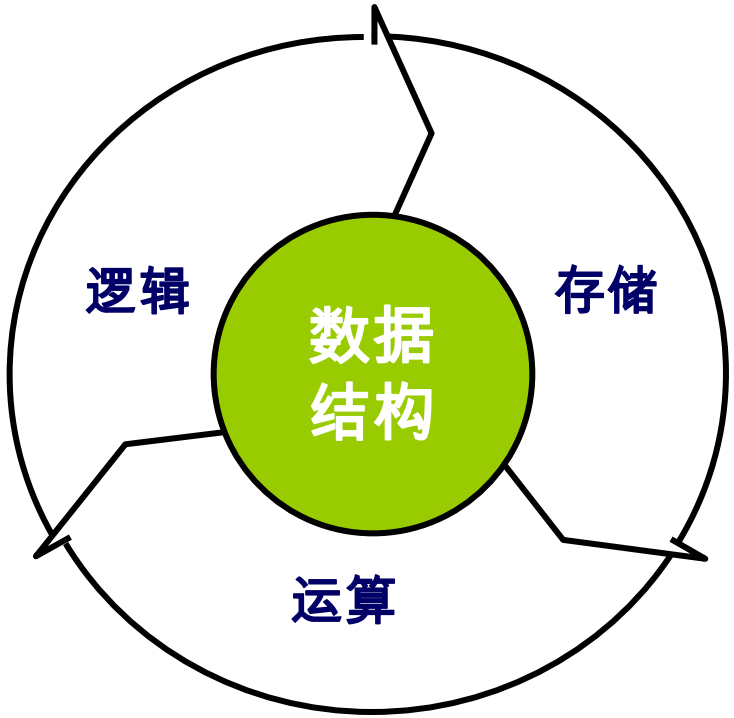
  - 链接：指针的地址指向关系



- 四类：**顺序、链接、索引、散列**

抽象数据类型

- 简称**ADT** (Abstract Data Type)
  - 定义了一组运算的数学模型
  - 与物理存储结构无关
  - 使软件系统建立在数据之上(面向对象)



- **模块化**的思想的发展
  - 隐藏运算实现的细节和内部数据结构
  - 软件复用

ADT 不关心存储细节

—— 例，C++版本括号匹配算法

```
void BracketMatch(char *str) {
  Stack<char> S; int i; char ch;
  // 栈可以是顺序或链式的，都一样引用
  for(i=0; str[i]!='\0'; i++)  {
      switch(str[i])  {
        case '(':   case '[':   case '{':
              S.Push(str[i]); break;
        case ')':  case ']':   case '}':
           if (S.IsEmpty( )) {
            cout<<"右括号多余!";
               return;
           }
           else {
              ch = S.GetTop( );
              if (Match(ch,str[i]))
                  ch = S.Pop( );
              else {
                  cout << "括号不匹配!";
                  return;
               }
             } /*else*/
          }/*switch*/
        }/*for*/
        if (S.IsEmpty( ))
            cout<<"括号匹配!";
        else cout<<"左括号多余" ;
    }
```

C 的顺序栈括号匹配算法 (与链式略不同)

```
void BracketMatch(char *str) {
  SeqStack S; int i; char ch;
  InitStack(&S);
  for(i=0; str[i]!='\0'; i++)  {
      switch(str[i])  {

       case '(':    case '[':    case '{':
            Push(&S,str[i]); break;

       case ')':   case ']':    case '}':
            if (IsEmpty(&S)) {
                printf("\n右括号多余!");
                return;
            }
            else {
                GetTop (&S,&ch);
                if (Match(ch,str[i]))
                    Pop(&S,&ch);
                else {
                    printf("\n括号不匹配!");
                    return;
                }
              } /*else*/
           }/*switch*/
         }/*for*/
         if (IsEmpty(&S))
             printf("\n括号匹配!");
         else printf("\n左括号多余" );
      }
```

C 的链式栈括号匹配算法 (与顺序栈不同)

```
void BracketMatch(char *str) {
  LinkStack S; int i; char ch;
  InitStack(/*&*/S);
  for(i=0; str[i]!='\0'; i++)  {
      switch(str[i])  {

       case '(':    case '[':    case '{':
            Push(/*&*/S, str[i]);
            break;

       case ')':   case ']':    case '}':
            if (IsEmpty(S)) {
                printf("\n右括号多余!");
                return;
            }
            else {
                GetTop (/*&*/S,&ch);
                if (Match(ch,str[i]))
                    Pop(/*&*/S,&ch);
                else {
                    printf("\n括号不匹配!");
                    return;
                 }
              } /*else*/
            }/*switch*/
          }/*for*/
          if (IsEmpty(/*&*/S))
              printf("\n括号匹配!");
          else printf("\n左括号多余" );
      }
```

抽象数据类型ADT

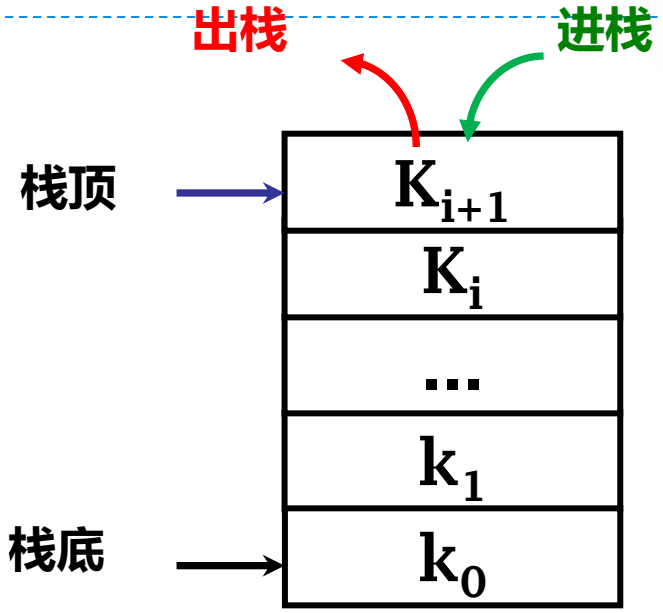
- 抽象数据结构二元组

  <数据对象D，数据操作P>

- 先定义逻辑结构，再定义运算
  - **逻辑结构**：数据对象及其关系
  - **运算**：数据操作

1.2 什么是数据结构

例：栈的抽象数据类型ADT



- 逻辑结构：线性表
- 操作特点：限制访问端口
  - 只允许在一端进行插入、删除操作
  - 入栈（ push ）、出栈（ pop ）、取栈顶（ top ）
    判栈空（ isEmpty ）

```
template <class T>                  // 栈的元素类型为 T
class Stack {
public:                             // 栈的运算集
    void clear();                   // 变为空栈
    bool push(const T item);        // item入栈，成功返回真，否则假
    bool pop(T & item);             // 弹栈顶，成功返回真，否则返回假
    bool top(T& item);              // 读栈顶但不弹出，成功真，否则假
    bool isEmpty(;                  // 若栈已空返回真
    bool isFull();                  // 若栈已满返回真
};
```

思考：关于抽象数据类型ADT

- 怎么体现逻辑结构？
- 抽象数据类型等价于类定义 ？
- 不用模板来定义可以描述 ADT 吗？

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