

# 第15讲 复习与练习

# 声明

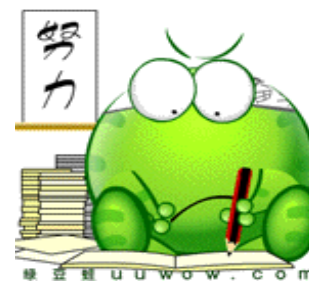
本文档用于帮助梳理课程所学习的内容，进一步强化所学习的知识点，但  
**绝非考试范围和考察重点！**

# 教学内容

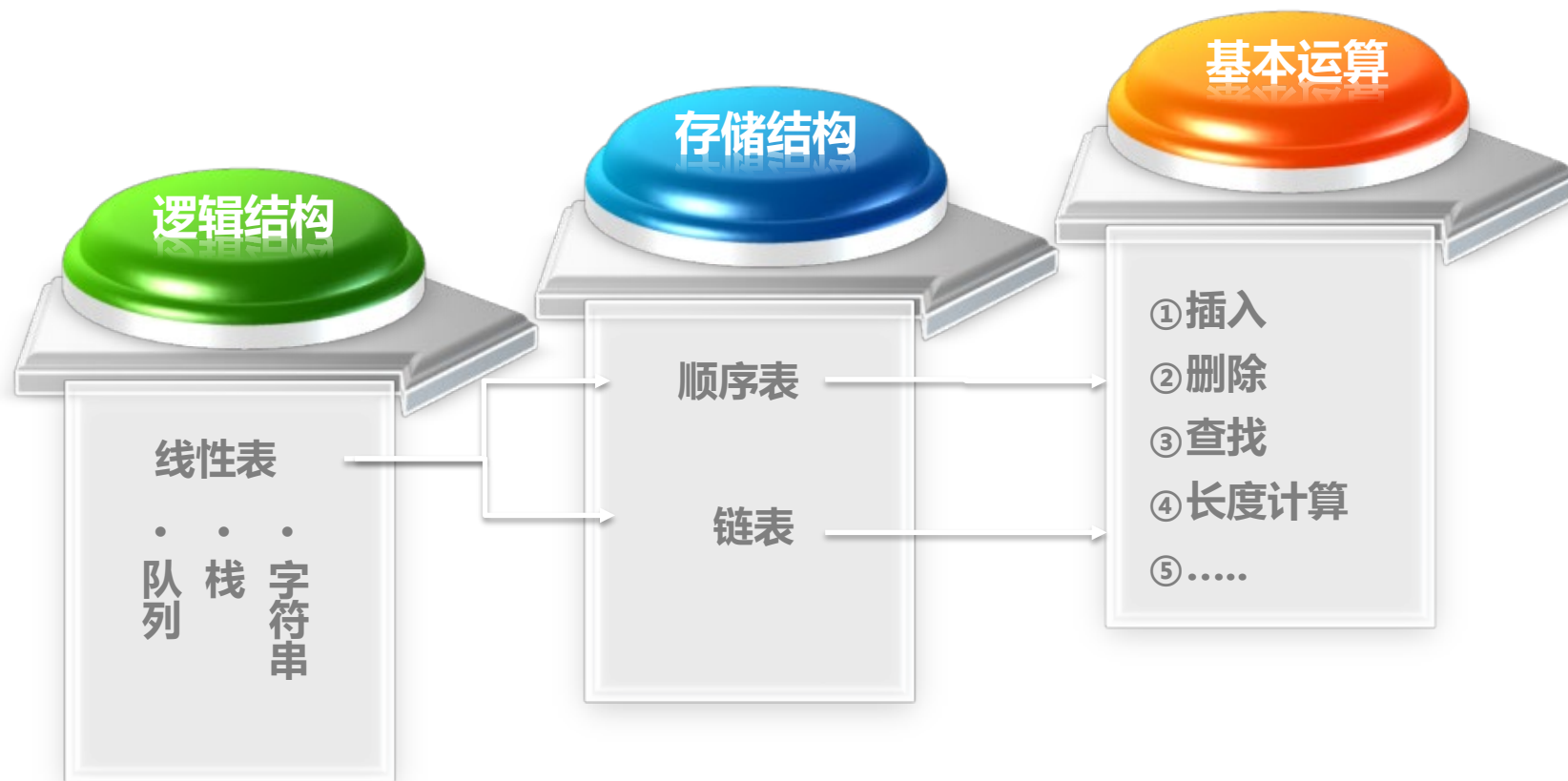
数据结构	算法
<b>线性表</b> 顺序表、链表、循环链表、字符串、队列、栈	排序（插入、选择、交换、归并、分配） 算法（分治、贪心、动划、穷举、回溯） 线性表的查找、插入、删除、合并等
<b>树结构</b> 二叉树、树、树林、二叉检索树、Huffman树、AVL树、字典	字典的散列存储和索引存储、 二叉树周游（四种）、Huffman建树、 树的周游等，AVL树的调整*
<b>图结构</b> 邻接表、逆邻接表、邻接矩阵	图的周游（DFS, BFS）、 最短路径(两种)、拓扑排序、 最小生成树(两种)、关键路径*等

# 课程学习后应达到的能力

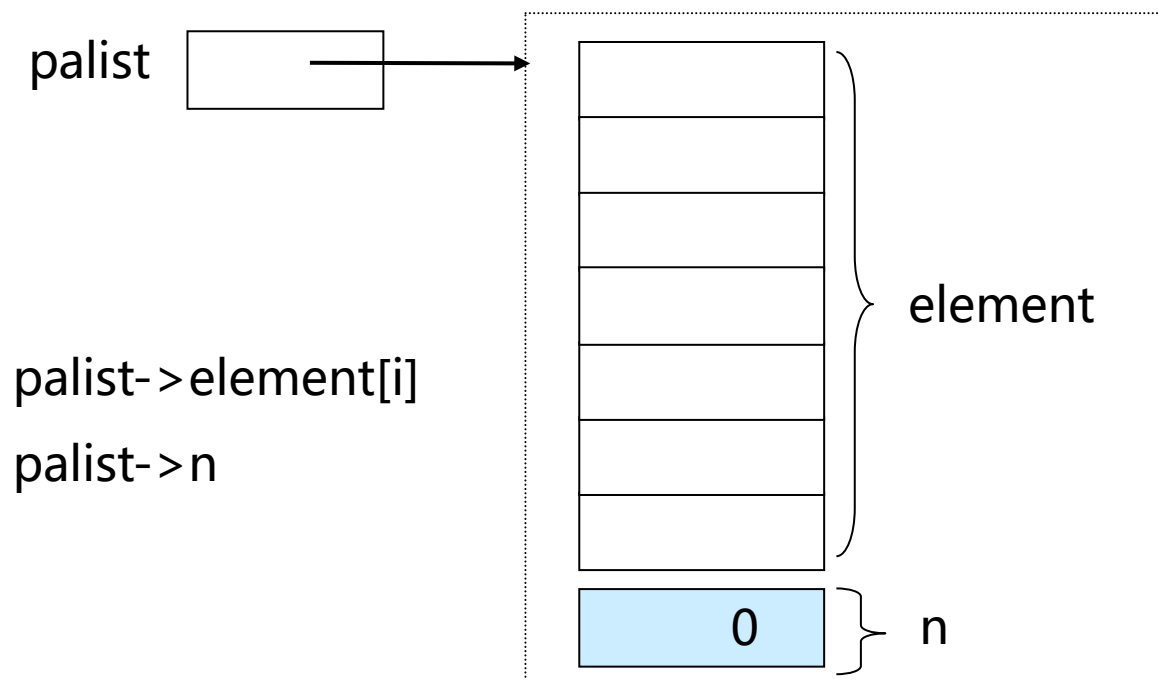
- ① 从逻辑结构、存储结构和相应的一组基本运算三个方面去掌握常用的数据结构。
- ② 对算法的时间和空间复杂性有一定的分析能力。
- ③ 针对简单的应用问题，应能选择合适的数据结构及设计有效的算法解决之。



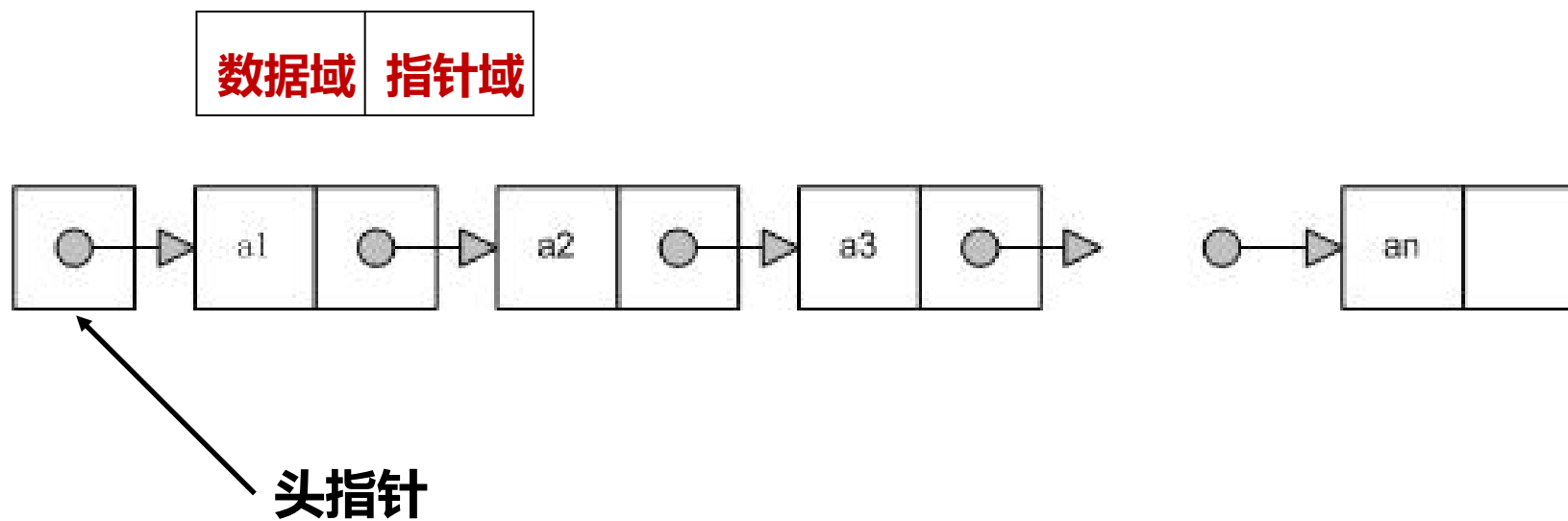
# 1、线性表（线性结构）



# 顺序表



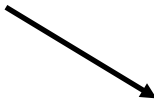
# 链表



# 链表的模拟

姓名	序号	后继
LI	002	3
WANG	004	5
ZHANG	001	0
XIA	003	1
ZOU	006	6
FENG	005	4
WANG	007	7
ZHAO	008	-1

头指针





# 特殊的线性结构

- **字符串：** 内容特殊的线性表
  - 常用的操作： **<string.h>** 中有关字符串的函数

串复制 `char *strcpy(char *destin, char *source);`

串拼接 `char *strcat(char *destin, char *source);`

串比较 `int strcmp(char *str1, char *str2);`

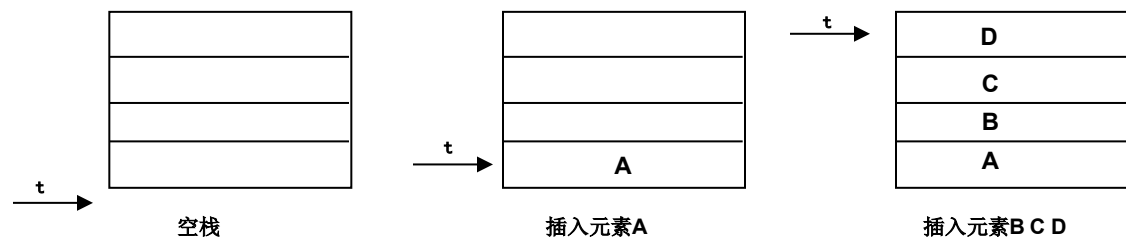
找子串 `char *strstr(const char *str1, const char *str2);`

.....

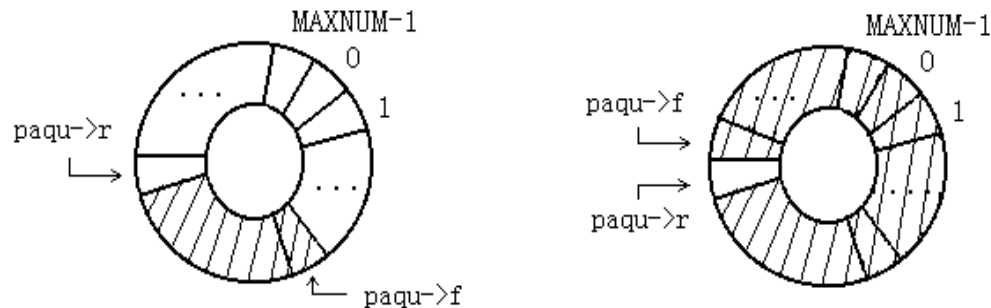
# 特殊的线性结构

- **栈和队列**: 操作特殊的线性表

- 顺序栈的定义和使用



- 循环队列的定义和使用



- 用作算法的辅助结构

**深度优先=》栈；      宽度优先=》队列**

# 基本的排序算法

- 排序的基本概念和排序算法
  - 插入排序 => 直接插入排序/ 折半插入排序/ Shell排序
  - 选择排序 => 直接选择排序/ 堆排序
  - 交换排序 => 起泡排序/ 快速排序
  - 分配排序 => 基数排序
  - 归并排序 => 归并排序
- 排序算法的比较和选择



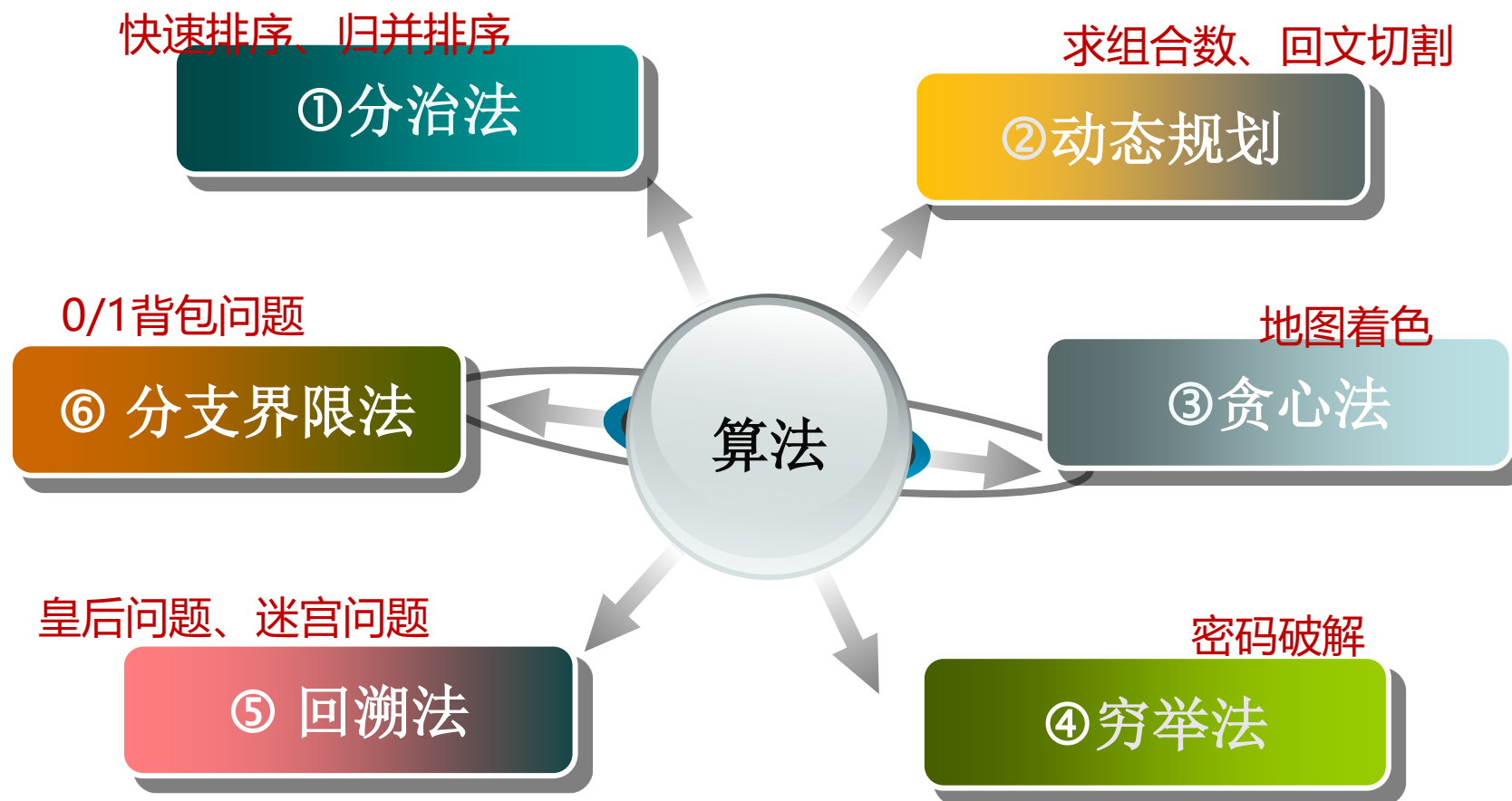
# 各种排序算法的比较

分类	算法	最大时间	平均时间	最小时间	空间代价	稳定性
插入排序	直接插入	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n)$	$\Theta(1)$	稳定
	二分插入	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n \log n)$	$\Theta(1)$	稳定
	Shell 排序	$\Theta(n^{3/2})$	$\Theta(n^{3/2})$	$\Theta(n^{3/2})$	$\Theta(1)$	不稳定
选择排序	直接选择	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(1)$	不稳定
	堆排序	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(1)$	不稳定

# 各种排序算法的比较

分类	算法	最大时间	平均时间	最小时间	空间代价	稳定性
交换排序	冒泡排序	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(1)$	稳定
	快速排序	$\Theta(n^2)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(\log n)$	不稳定
分配排序	桶式排序	$\Theta(n+m)$	$\Theta(n+m)$	$\Theta(n+m)$	$\Theta(m)$	稳定
	基数排序	$\Theta(d \cdot (n+r))$	$\Theta(d \cdot (n+r))$	$\Theta(d \cdot (n+r))$	$\Theta(n+r)$	稳定
归并排序	二路归并	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n)$	稳定

# 算法的基本策略



# 简单练习 (一)

# 判断题

1. 如果某种算法是不稳定的,则该方法没有实际应用价值。
2. 对于n个记录的集合进行冒泡排序, 所需要的平均时间是 $O(n)$  。
3. 对于n个记录的集合进行快速排序, 在最坏情况下所需要的平均时间是 $O(n^2)$  。
4. 对n个元素的序列进行起泡排序时最小的比较次数是 $n-1$ 。

N N Y Y



# 选择题

- 对n个不同的排序码的元素进行不减序冒泡排序，在（A）情况下比较的次数最多，其比较次数为（ B ）。

A:     1. 从大到小排列好的;    2. 从小到大排列好的  
         3. 元素无序;                4. 元素基本有序

-----

B:     1.  $n+1$  ;        2.  $n$   
         3.  $n-1$  ;        4.  $n(n-1)/2$

A: 1    B: 4

# 选择题

- 在内排序的过程中，通常需要对待排序的关键码集合进行多便扫描。采取不同排序方法会产生不同的排序中间结果。设要将序列{Q, H, C, Y, P, A, M, S, R, D, F, X}中的关键码按字母序的升序重新排列，则（ ）是冒泡排序的一次扫描结果。

- (1) F, H, C, D, P, A, M, Q, R, S, Y, X
- (2) P, A, C, S, Q, D, F, X, R, H, M, Y
- (3) A, D, C, R, F, Q, M, S, Y, P, H, X
- (4) H, C, Q, P, A, M, S, R, D, F, X, Y

(4)

# 选择题

- 下述几种排序方法中，要求内存量（空间复杂性）最大的是( )。

A.插入排序

B.选择排序

C.快速排序

D.归并排序

D

# 选择题

- 以下排序方法中，关键字比较的次数与记录的初始排列次序无关的是（ ）。

- A shell排序
- B 起泡排序
- C 直接插入排序
- D 直接选择排序

D

# 简单练习 (二)

# 判断题

1. 在线性表的链式存储结构中，逻辑上相邻的元素在物理位置上不一定相邻;
2. 在线性表的顺序存储结构中，插入和删除时，移动元素的个数与该元素的位置有关;
3. 线性表的链式存储结构的特点是用一组任意的存储单元存储线性表的数据元素;
4. 若两个串含有相同的字符，则说它们相等;
5. KMP算法的最大特点是指主串的指针不需回溯;
6. 在顺序表中取出第 $i$ 个元素所花费的时间与 $i$ 成正比。

Y Y Y N Y N

# 填空题

1. 若已知一个栈的入栈序列是1, 2, 3, ... n, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$ , 若 $p_1=n$ , 则 $p_i$ 为  $n-i+1$ 。
2. 假设以数组A[m]存放循环队列的元素, 其头指针是front, 当前队列有k个元素, 则队列的尾指针为  $(front+k) \% m$ 。
3. 数据结构是研究数据的 逻辑结构 和 存储结构, 以及这个结构具有的行为特征, 定义相应的运算, 设计出相应的 算法。

# 选择题

- 链表不具有的特点是（      ）。
- A. 可随机访问任意元素
- B. 插入和删除不需要移动元素
- C. 不必事先估计存储空间
- D. 所需空间与线性表长度成正比

A



# 选择题

- 在双向链表中，删除p所指的结点时需( )。

A.  $(p \rightarrow \text{llink}) \rightarrow \text{rlink} = p \rightarrow \text{rlink};$

$(p \rightarrow \text{rlink}) \rightarrow \text{llink} = p \rightarrow \text{llink};$

B.  $p \rightarrow \text{link} = (p \rightarrow \text{llink}) \rightarrow \text{link};$

$((p \rightarrow \text{llink}) \rightarrow \text{llink}) \rightarrow \text{rlink} = p;$

C.  $((p \rightarrow \text{llink}) \rightarrow \text{llink}) \rightarrow \text{rlink} = p;$

$p \rightarrow \text{llink} = (p \rightarrow \text{llink}) \rightarrow \text{llink};$

A

# 选择题

- 设环形队列中数组长度为  $n$ ，头尾指针分别为  $f$  和  $r$ ，则队列中现有元素个数为：  
( )
  - A.  $r-f$
  - B.  $r-f+1$
  - C.  $(r-f+1) \bmod n$
  - D.  $(r-f+n) \bmod n$

D

# 算法设计

- 编写一函数从一给定数组A中删除元素值在X到Y( $X \leq Y$ )之间的所有元素，要求以较低的时间复杂性来实现

## 一种思路：

先将所有元素值在x到y之间的元素置成一个特殊的值，然后从最后向前依次扫描，对于该特殊值的元素便移动其后面的元素将其删除(这比每删除一个元素后立即移动其后元素效率要高些)。

## 另一种思路：

改进快速排序，将所有X到Y的元素移动到数组的右侧，然后一起删除。

# 教学内容

数据结构	算法
<b>线性表</b> 顺序表、链表、循环链表、字符串、队列、栈	排序（插入、选择、交换、归并、分配） 算法（分治、贪心、动划、穷举、回溯） 线性表的查找、插入、删除、合并等
<b>树结构</b> 二叉树、树、树林、二叉检索树、Huffman树、AVL树、字典	字典的散列存储和索引存储、 二叉树周游（四种）、Huffman建树、 树的周游等，AVL树的调整*
<b>图结构</b> 邻接表、逆邻接表、邻接矩阵	图的周游（DFS, BFS）、 最短路径(两种)、拓扑排序、 最小生成树(两种)、关键路径*等

## 2、树和二叉树

- 二叉树
  - 基本概念：术语、特殊二叉树（完全二叉树、满二叉树）
  - 基本性质：**第 $i$ 层至多有 $2^i$ 个结点( $i \geq 0$ )**;  **$n_0 = n_2 + 1$**
  - 基本运算：建立、周游、查找、检索、.....
- 哈夫曼树和二叉排序(搜索)树
  - 哈夫曼算法、编码和译码
  - 二叉排序树基本性质和应用
- 树与树林
  - 基本概念
  - 树的周游



# 顺序表示的二叉树

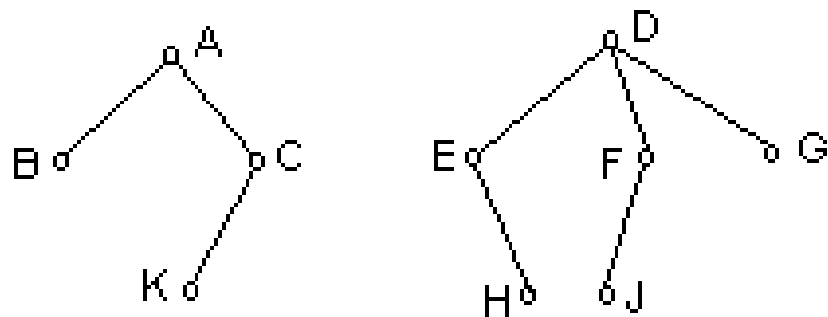
采用顺序存储表示的二叉树定义如下:

```
① #define MAXNODE 100      /* 定义二叉树中结点的最大个数 */
② struct SeqBTree          /* 顺序树类型定义 */
③ {
④     DataType nodelist[MAXNODE];
⑤     int      n;           /* 改造成完全二叉树后, 结点的个数 */
⑥ }
⑦ typedef struct SeqBTree *PSeqBTree;    /* 顺序树类型的指针类型 */
⑧ PSeqBTree pabtree;      /* pabtree是指向顺序树的一个指针变量 */
```

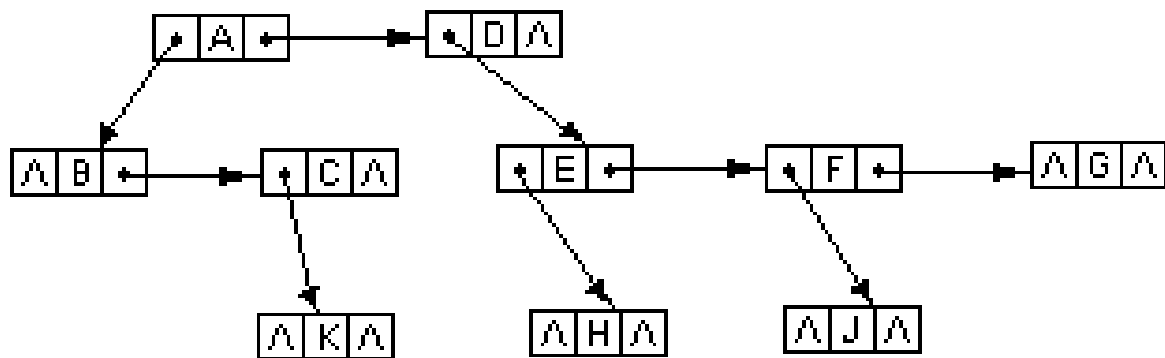
# 链式表示的二叉树

- ① **struct BinTreeNode**            **/\* 二叉树中结点 \*/**
- ② **typedef struct BinTreeNode \*PBinTreeNode;**
  
- ③ **struct BinTreeNode**
- ④ **{**
- ⑤        **DataType info;**            **/\* 数据域 \*/**
- ⑥        **PBinTreeNode llink;**        **/\* 指向左子女 \*/**
- ⑦        **PBinTreeNode rlink;**        **/\* 指向右子女 \*/**
- ⑧ **};**
- ⑨ **typedef struct BinTreeNode \*BinTree;**        **/\*二叉树定义\*/**
- ⑩ **typedef BinTree \*PBinTree;**            **/\* 二叉树类型的指针类型 \*/**
  
- ⑪ **PBinTree t;**            **/\* t是指向二叉树类型的指针变量 \*/**

# 树和树林的存储表示



- 长子-兄弟表示法





# 长子-兄弟表示法

- ① **//类型定义**
- ② **struct CSNode;**
- ③ **typedef struct CSNode \* PCSNode;**
- ④ **struct CSNode {**
- ⑤ **DataType        info;**
- ⑥ **PCSNode        lchild;**
- ⑦ **PCSNode        rsibling;**
- ⑧ **};**
- ⑨ **typedef struct CSNode \*CSTree;**
- ⑩ **typedef CSTree \*PCSTree;**
- ⑪ **//定义一个指向树/树林的指针变量**
- ⑫ **PCSTree t;**

# 字典和检索

- 顺序表表示
  - 顺序检索算法\二分法检索算法
- 散列构造的字典和检索
  - 散列函数
  - 碰撞处理
- 索引简介
  - 二叉排序树
  - AVL树（算法不要求）



# 简单练习 (三)

# 判断题

- 二叉树的先根遍历序列中，任意一个结点均处在其子女结点的前面。
- 由树转换成二叉树，其根结点的右子树总是空的。
- 哈夫曼树是带权路径长度最短的树，路径上权值较大的结点离根较近。

Y Y Y

# 选择题

- 已知某二叉树的后根周游序列是dabec，中根周游是debac，它的先根周游序列是（ ）。
  - A acbed
  - B decab
  - C deabc
  - D cedba

D

# 选择题

- 任何一棵二叉树的叶结点在先根、中根和后根周游序列中的相对次序 ( )
  - A 不发生改变
  - B 发生改变
  - C 不能确定
  - D 以上都不对

A

# 选择题

- 某二叉树的先根周游序列和后根周游序列正好相反，则该二叉树一定是（            ）二叉树。
  - A. 空或只有一结点
  - B. 树的高度等于其结点数减1
  - C. 任一结点都只有右子结点
  - D. 任一结点都只有左子结点

B

# 选择题

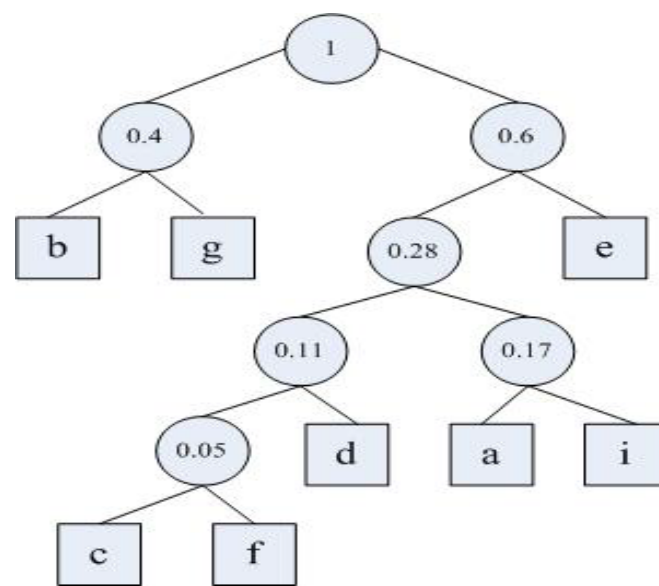
- 设高度为 $h$ 的二叉树上只有度为0和度为2的结点，则此类二叉树中所包含的结点数至少为( )。
  - A.  $2h$
  - B.  $2h-1$
  - C.  $2h + 1$
  - D.  $h+1$

C



# 解答题

- 假设通讯电文由8个字符：a、b、c、d、e、f、g、i组成，
  - 出现频率依次为0.07、0.19、0.02、0.06、0.32、0.03、0.21、0.10
  - 为这8个字母设计哈夫曼编码
  - 请画出哈夫曼树，并给出字符串“edfa”的编码。



“11 1001 10001 1010”  
或者  
“00 0110 01110 0101”

# 解答题

- 对于表达式  $(a-b)*d/(e+f)$ ,
  - 请画出它的中序二叉树,
  - 给出该表达式的前缀表达式形式和后缀表达式形式

# 算法设计

- 二叉树采用链式存储结构，设计一个按层次顺序（同一层自左向右）遍历二叉树的算法

## 解题思路：

采用一个队列q，先将二叉树根结点入队列，然后退队列，输出该结点，若它有左子树，便将左子树的根结点入队列，若它有右子树，便将右子树根结点入队列，如此直到队列为空为止。

因为队列是先进先出的，从而达到按层次顺序遍历二叉树的目的。

# 简单练习 (四)

# 判断题

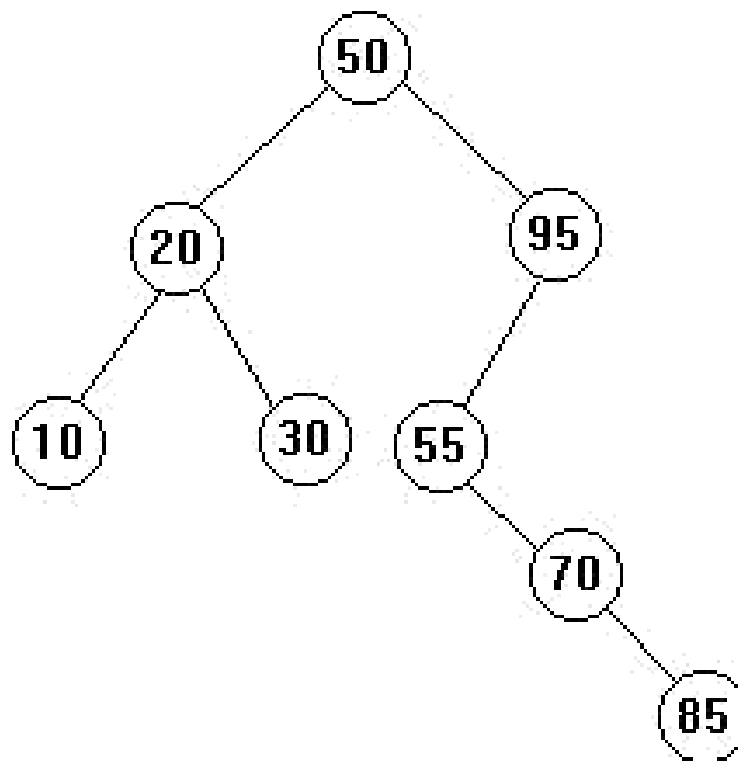
1. 二叉排序的查找和折半查找时间的性能相同
2. 散列法存储的基本思想是由关键码的值决定数据的存储地址
3. 任一二叉排序树的平均查找时间都小于用顺序查找同样结点的线形表的平均查找时间

N Y N

# 选择题

- 有一棵二叉树，如图所示该二叉树是( )。

- A. 二叉平衡树
- B. 二叉排序树
- C. 堆的形状



**B**

# 选择题

- 对线性表进行二分查找时,要求线性表必须( )
  - A.以顺序方式存储
  - B.以链接方式存储
  - C.以顺序方式存储,且结点按关键字有序排序
  - D.以链接方式存储,且结点按关键字有序排序

C

# 选择题

- 当有序表为

{ 1, 3, 9, 12, 32, 41, 45, 62, 75, 77, 82, 95, 100 },

则二分查找值为82的结点时, ( )次比较后成功。

- A. 1
- B. 2
- C. 4
- D. 8

C



# 算法设计

- 假设二叉排序树t各元素值均不相同，设计一个算法按递减次序打印各元素值。

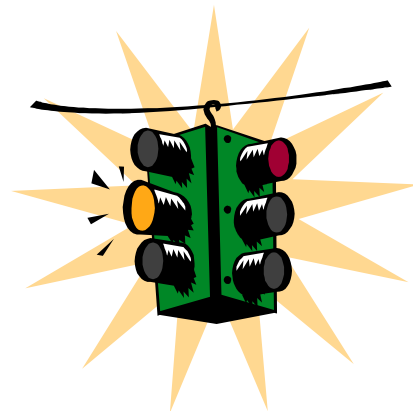
```
//二叉树中根序周游的修订
void destorder( bTree t)
{
    if (t!=NULL)
    {
        destorder(t->rlink);
        print (t->info);
        destorder(t->llink);
    }
}
```

# 教学内容

数据结构	算法
<b>线性表</b> 顺序表、链表、循环链表、字符串、队列、栈	排序（插入、选择、交换、归并、分配） 算法（分治、贪心、动划、穷举、回溯） 线性表的查找、插入、删除、合并等
<b>树结构</b> 二叉树、树、树林、二叉检索树、Huffman树、AVL树、字典	字典的散列存储和索引存储、 二叉树周游（四种）、Huffman建树、 树的周游等，AVL树的调整*
<b>图结构</b> 邻接表、逆邻接表、邻接矩阵	图的周游（DFS, BFS）、 最短路径(两种)、拓扑排序、 最小生成树(两种)、关键路径*等

# 图结构

- 图的基本概念
  - 图、子图、无向图，有向图、入/出度、路径、连通（图）、带权图、根、生成树 等
- 存储表示
  - 邻接矩阵表示法
  - 邻接表表示法
- 图的基本运算与周游算法
  - 计算顶点入度、出度计算
  - 计算周游序列, DFS(深度优先)/BFS(广度优先)
- 图的最小生成树，最短路径，拓扑排序



# 概念—完全图

- 图G的顶点数 $n$ 和边数 $e$ 满足以下关系：
  - 若G是有向图，则 $0 \leq e \leq n(n-1)$ 
    - 有向完全图：有 $n(n-1)$ 条边的有向图
  - 若G是无向图，则 $0 \leq e \leq n(n-1)/2$ 
    - 无向完全图：有 $n(n-1)/2$ 条边的无向图
- 在顶点个数相同的图中，完全图具有最多的边
  - 具有4个顶点的无向完全图，边数为 $4*(4-1)/2=6$

# 概念—无向图的连通

- **连通**：无向图 $G=(V, E)$ 中，若从 $v_i$ 到 $v_j$ 有一条路径(从 $v_j$ 到 $v_i$ 也一定有一条路径)，则称 $v_i$ 和 $v_j$ 是连通的。
- **连通图**：若 $V(G)$ 中任意两个不同的顶点 $v_i$ 和 $v_j$ 都是连通的(即有路径)，则称 $G$ 为连通图。
- **连通分量**：无向图 $G$ 的最大连通子图 $G'$ （即任意增加 $G$ 中结点和边到 $G'$ 所得到的子图都不再连通），称为 $G$ 的连通分量。
  - 连通图只有一个连通分量，就是其自身；
  - 非连通的无向图有多个连通分量。

# 概念—有向图的连通

- **强连通图**：有向图 $G=(V, E)$ 中，若 $V(G)$ 中任意两个不同的顶点 $v_i$ 和 $v_j$ 都存在从 $v_i$ 到 $v_j$ 以及从 $v_j$ 到 $v_i$ 的路径，则称图 $G$ 是强连通图
- **强连通分量**：有向图 $G$ 的最大强连通子图，称为图的强连通分量
  - 强连通图只有一个强连通分量，就是其自身；
  - 非强连通的有向图有多个强连通分量。

# 图的邻接矩阵表示法

```
① #define MAXVEX  常数1
② #define MAXEDGE 常数2
③ typedef char VexType;
④ typedef float AdjType;

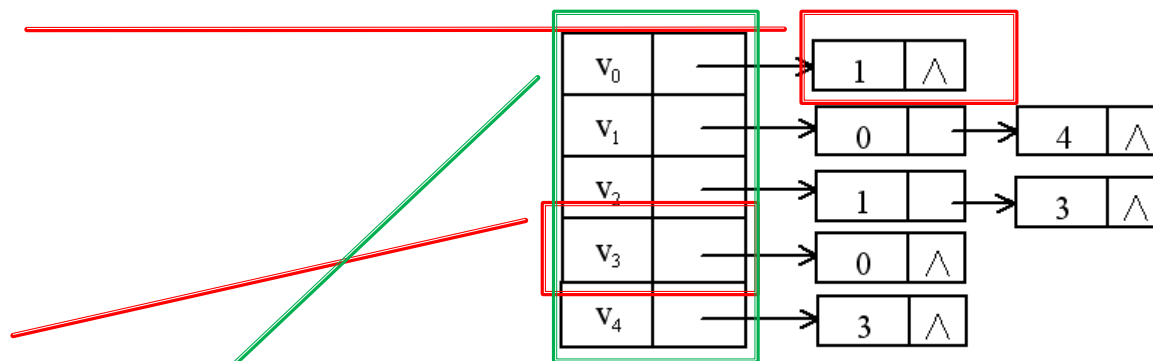
⑤ typedef struct
⑥ {
⑦     VexType vexs[MAXVEX];          /* 顶点信息 */
⑧     AdjType arcs[MAXVEX][MAXVEX]; /* 邻接矩阵 */
⑨     int n;      /* 图的顶点个数 */
⑩ } Graph;
```

$$A_1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} \infty & 3 & 5 & 8 & \infty \\ 3 & \infty & 6 & 4 & 11 \\ 5 & 6 & \infty & 2 & \infty \\ 8 & 4 & 2 & \infty & 10 \\ \infty & 11 & \infty & 10 & \infty \end{bmatrix}$$

# 图的邻接表表示法

- ① struct EdgeNode;
- ② typedef struct EdgeNode \* PEdgeNode;
- ③ typedef struct EdgeNode \* EdgeList;
- ④ struct EdgeNode  
{  
    int endvex;  
    AdjType weight;  
    PEdgeNode nextedge;  
}; /\* 边表中的结点\*/
- ⑤ typedef struct  
{  
    VexType vertex;  
    EdgeList edgelist;  
} VexNode; /\* 顶点表中的结点 \*/
- ⑥ typedef struct  
{  
    VexNode vexs[MAXVEX];  
    int n;  
} GraphList;





# 简单练习 (五)

# 判断题

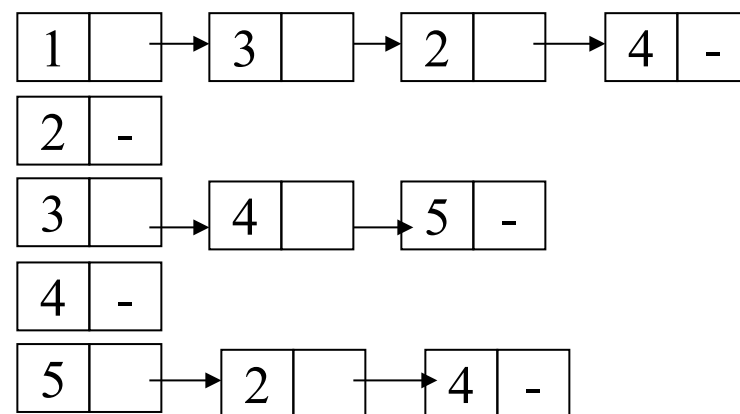
1. 用相邻接矩阵法存储一个图时，在不考虑压缩存储的情况下，所占用的存储空间大小只与图中结点个数有关，而与图的边数无关。
2. 任何有向图网络（AOV-网络）拓扑排序的结果是唯一的。
3. 强连通分量是有向图中的极大强连通子图。
4. 一个有向图的邻接表和逆邻接表中的结点个数一定相等。

Y N Y Y

# 选择题

- 已知一个有向图的邻接表存储结构如图所示，根据有向图的深度优先遍历算法,从顶点v1 出发，所得的顶点序列是(     )，按广度优先遍历算法得到序列(     )。

- A. v1,v2,v3,v5,v4
- B. v1,v2,v3,v4,v5
- C. v1,v3,v4,v5,v2
- D. v1,v4,v3,v5,v2
- E. v1,v3,v2, v4,v5



C     E

# 选择题

- 设无向图G中顶点数为n，则图最多有（ ）条边。

A、  $n(n+1)/2$

B、  $n(n-1)$

C、  $n(n-1)/2$

D、  $n(n+1)$

C

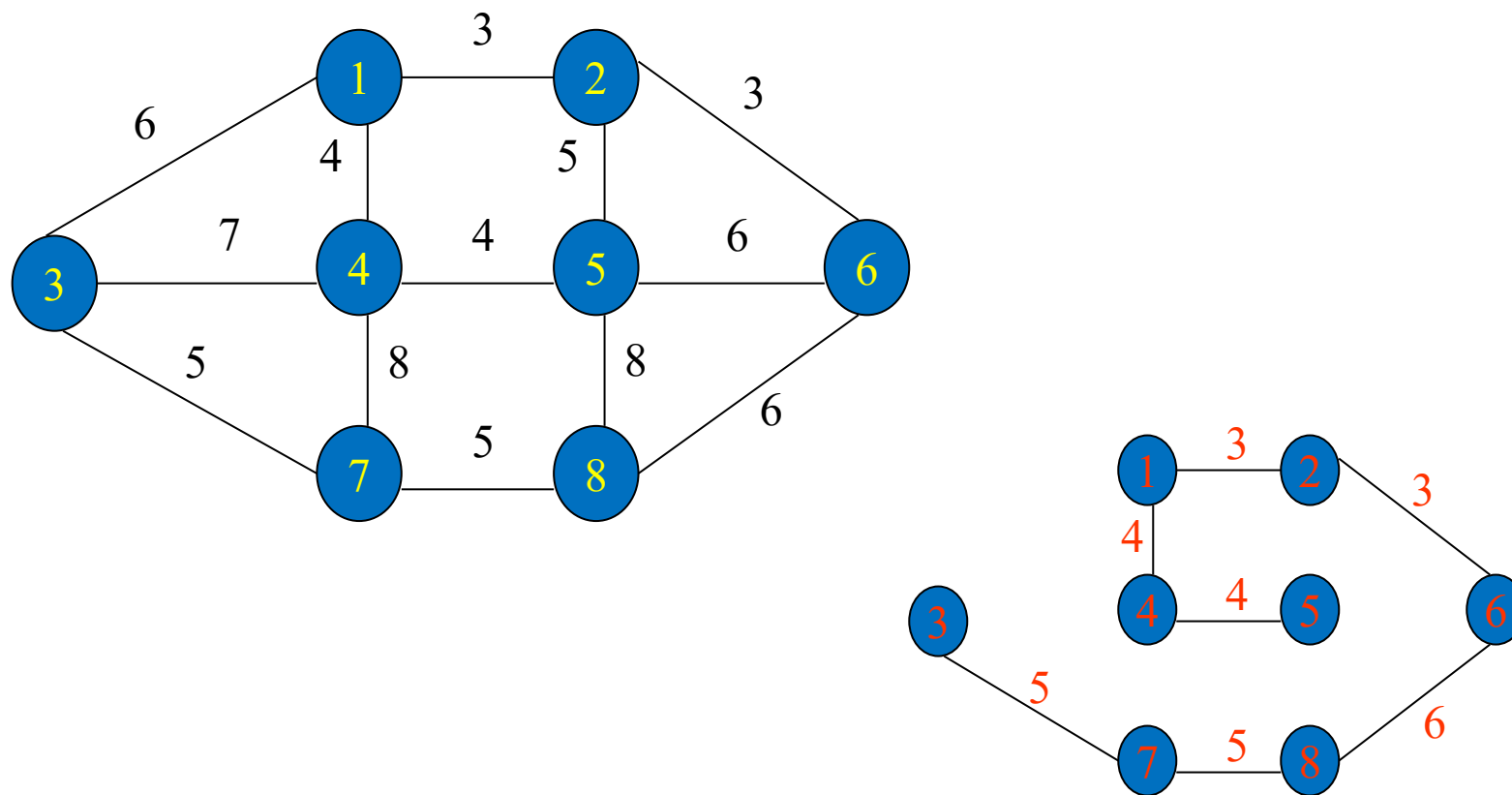
# 选择题

- 图的生成树（ ），一个连通图的生成树是该图的一个（ ）连通子图， $n$ 个顶点的生成树有（ ）条边。
  - A. 是唯一的、最大、 $n$
  - B. 不是唯一的、最小、 $n+1$
  - C. 唯一性不能确定、最大、 $n-1$
  - D. 唯一性不能确定、最小、 $n-1$

C

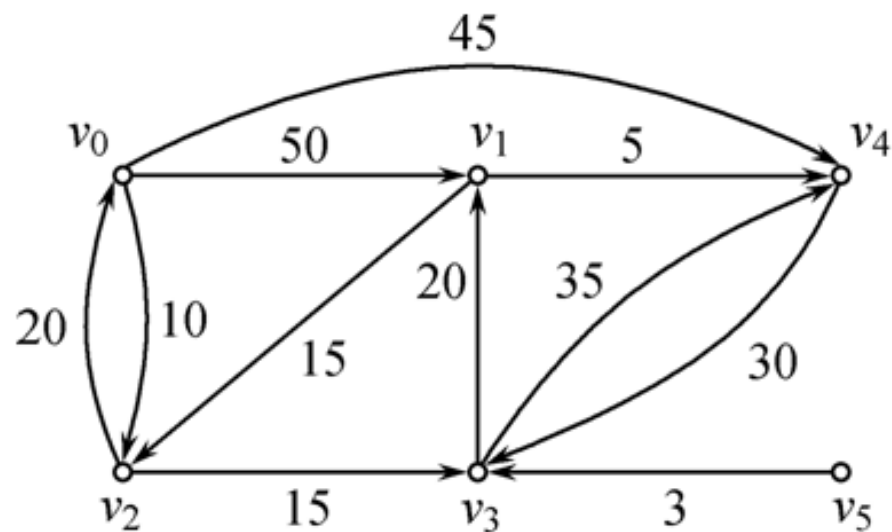
# 习题解答

- 由Kruskal算法，从V1点开始，求下图的一棵最小生成树。



# 解答题

- 下面是一个带权图，试给出该图的邻接矩阵/邻接表存储表示，试给出该图中由 $V_0$ 到 $V_4$ 的最短路径。



# 算法设计

- 设计算法判断无向图G是否是一棵树？若是树，则返回true；否则，返回false。

思路：

- ① G是一棵树的条件是G必须是无回路的连通图或者是有 $n-1$ 条边的连通图；
- ② 对于连通的判断可用图的遍历算法实现；
- ③ 对于回路的判断，根据回路定义，回路是除了第一顶点和最后一个顶点之外，其余顶点不重复出现的路径，因此遍历时设标记判断。



# 算法设计

- 判定无向图G是否连通，若不连通则求该图有多少连通分量。

思路：

- ① 对于连通的判断可用图的遍历算法实现；
- ② 对于回路的判断，根据回路定义，回路是除了第一顶点和最后一个顶点之外，其余顶点不重复出现的路径，因此遍历时设标记判断。

# 期末考试

## 二.常见违纪/作弊情况

- 对学生考试违纪/作弊行为的认定和处理依据“北京大学本科考试工作与学习纪律管理规定”

### 1. 违纪

第三十二条 学生在考试中有下列情形之一的，属违反考试纪律行为，监考人员当场给予口头警告并予以纠正：

- （一）未按考场规则隔位就座或未按监考教师指定座位就座的；
- （二）至监考人员分发试卷时尚未将书包或其他与考试无关的物品放在指定位置的；
- （三）已收起的物品中有未关闭的手机等电子设备，致使呼叫声影响考场秩序的；
- （四）未经允许携带自备草稿纸（空白）的；
- （五）未遵守监考人员指令，提前开始答题或不按时结束答题的；
- （六）在考场附近高声喧哗，交卷后仍在考场逗留或有其他影响考场秩序行为的；
- （七）其他违反考试纪律构成口头警告的情形。

# 期末考试

## 二.常见违纪/作弊情况

### 2. 对使用手机等电子设备作弊行为的规定

第三十四条 除开卷考试中教师另有说明外，学生在开始考试答题至交卷离场期间有下列情形之一的，属考试作弊行为，视情节轻重给予记过或留校察看处分，该门课程总成绩按零分处理：

（二）携带具有发送、接收信息功能或存储有与考试课程内容相关材料的电子设备（如手机、智能手表、非教师允许的计算器等）参加考试的；

（六）利用上厕所等暂时离开考场之机，在考场外看与考试课程相关的资料、与他人交流考试内容、使用手机等各种具有信息发送、接收、存储功能的设备的；

第三十五条 学生有下列情形之一的，属严重作弊行为，给予开除学籍处分，该门课程总成绩按零分处理：

（三）使用手机等具有通讯功能的设备、器材与他人串通作弊的；

3. 第三十四条第七款，在考试过程中“故意损坏试卷、答案、草稿纸等考试用纸的”也属考试作弊行为。

# 期末考试

## 二.常见违纪/作弊情况

### 3. 强调

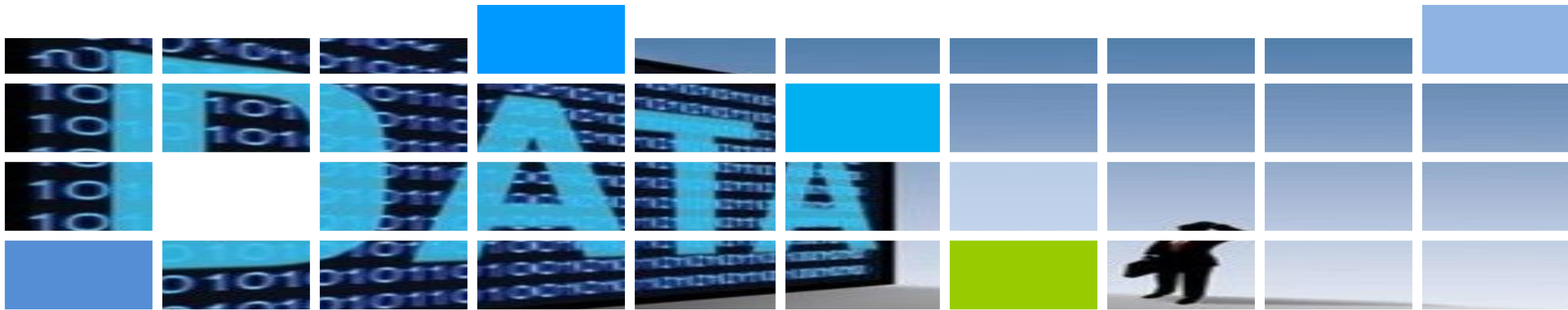
实际案例中，常见携带**手机、智能手表、与考试课程相关材料**等物品的情况：

（一）携带与考试课程内容相关的材料参加考试，或者在本人课桌、座位及旁边发现有与考试课程内容相关的材料的；

（二）携带具有发送、接收信息功能或存储有与考试课程内容相关材料的电子设备（如手机、智能手表、非教师允许的计算器等）参加考试的；

以上情形按规定属考试作弊行为，将给予记过或留校察看处分。请注意，**“携带”本身就违规了，按作弊处理，不是看了用了才算作弊。因此，监考人员请务必重视考前清理物品！！**





燕园认识诸君，三生有幸！  
祝愿前程似锦，功不唐捐！