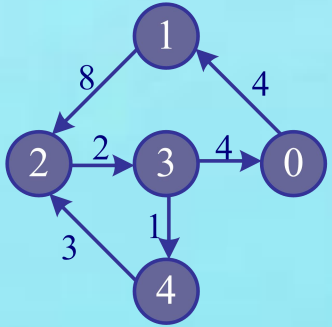
本节主题：

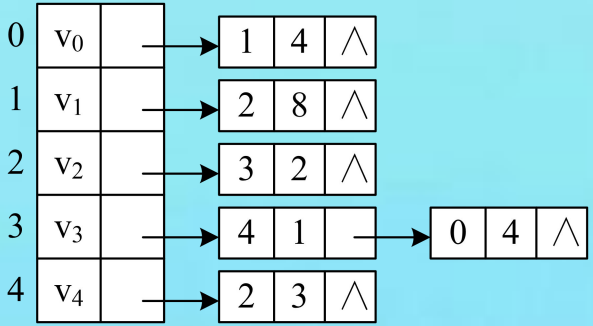
图的邻接表存储结构

图的存储结构

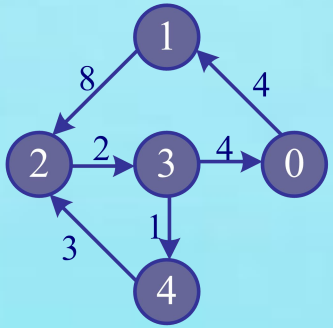
- 邻接矩阵存储方法
- 邻接表存储方法

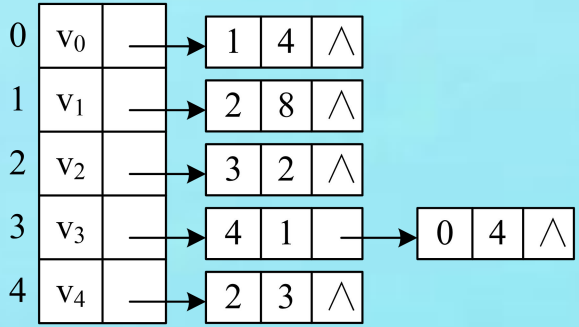


$$A=\begin{bmatrix} 0 & 4 & \infty & \infty & \infty \\ \infty & 0 & 8 & \infty & \infty \\ \infty & \infty & 0 & 2 & \infty \\ 4 & \infty & \infty & 0 & 1 \\ \infty & \infty & 3 & \infty & 0 \end{bmatrix}$$



邻接表存储方法





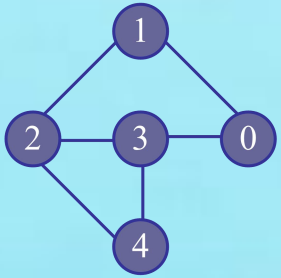
表头节点 | data | firstarc

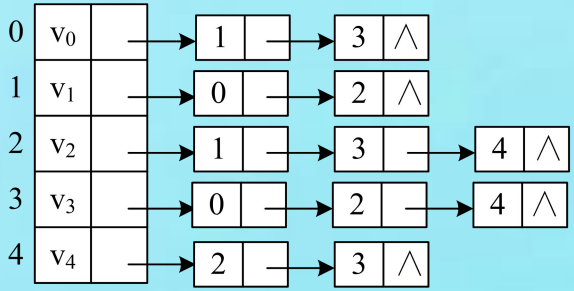
边表节点 | adjvex | nextar | info

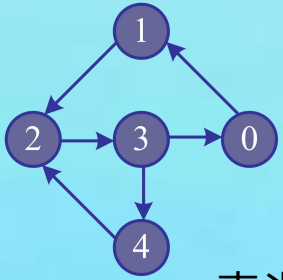
- 顺序分配与链式分配相结合
- 每个顶点建立一个单链表
- 第j个单链表中的节点表示依附于顶点j的边
  - 每个单链表上附设一个表头节点。
  - 对有向图，是以顶点j为尾的边。

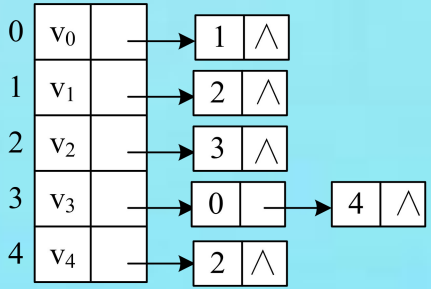
```
typedef struct ANode
{
    int adjvex;
    struct ANode *nextarc;
    InfoType info;
} ArcNode; //边表节点类型
typedef struct Vnode
{
    Vertex data;
    ArcNode *firstarc;
} VNode;   //表头节点类型
typedef VNode AdjList[MAXV];
typedef struct
{
    AdjList adjlist;
    int n,e;
} ALGraph; //完整的图邻接表类型
```

邻接表存储无权图









表头节点 | data | firstarc

边表节点 | adjvex | nextar | info

邻接表的特点：

- 邻接表表示不唯一：在每个顶点对应的单链表中，各边节点的链接次序可以是任意的。
- 可能空间耗费大：对于有n个顶点和e条边的无向图，其邻接表有n个顶点节点和2e个边节点。
- 对于无向图，邻接表的顶点i对应的第i个链表的边节点数目正好是顶点i的度。
- 对于有向图，邻接表的顶点i对应的第i个链表的边节点数目仅仅是顶点i的出度；其入度为邻接表中所有adjvex域值为i的边节点数目。

算法：将邻接矩阵转换为邻接表

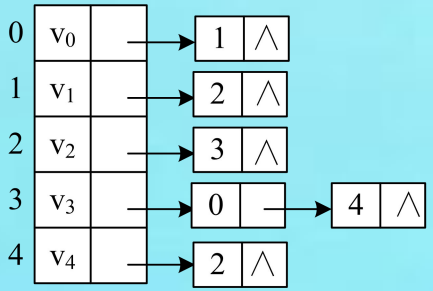
```
void MatToList(MGraph g，ALGraph *&G)
{
   int i, j;
   ArcNode *p;
   G=(ALGraph *)malloc(sizeof(ALGraph));
   //给所有头节点的指针域置初值
   //根据邻接矩阵建立邻接表中节点

   G->n=n;
   G->e=g.e;
}
```

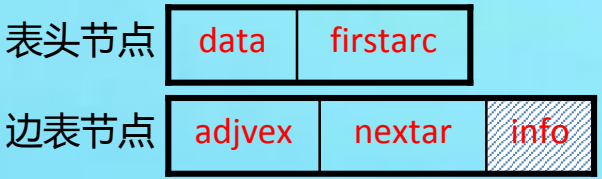
```
for (i=0; i<g.n; i++)
   G->adjlist[i].firstarc=NULL;
```

```
for (i=0; i<g.n; i++)
   for (j=g.n-1; j>=0; j--)
      if (g.edges[i][j]!=0)
      {
         p=(ArcNode *)malloc(sizeof(ArcNode));
         p->adjvex=j;
         p->nextarc=G->adjlist[i].firstarc;
         G->adjlist[i].firstarc=p;
      }
```

时间复杂度
$O(n^2)$

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



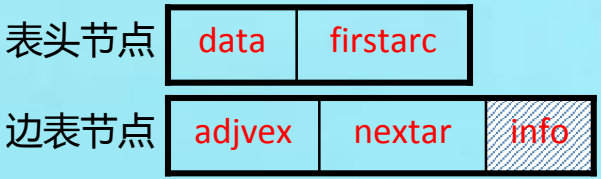


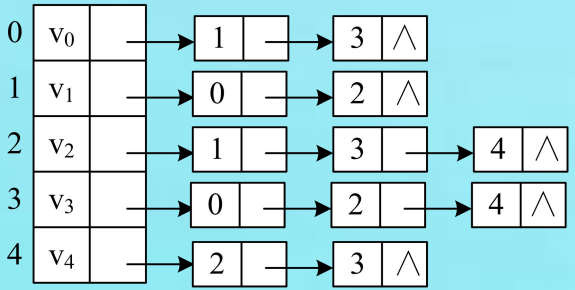
算法：将邻接表转换为邻接矩阵

```
void ListToMat(ALGraph *G , MGraph &g)
{
    int i , j;
    ArcNode *p;
    for (i=0; i<G->n; i++)
    {
        p=G->adjlist[i].firstarc;
        while (p!=NULL)
        {
            g.edges[i][p->adjvex]=1;
            p=p->nextarc;
        }
    }
    g.n=G->n;
    g.e=G->e;
}
```

要求g的实参调用前已经初始化为全0

时间复杂度
O(n+e)





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

思考题

（1）对于有n个顶点e条边的无向图，邻接表表示时有多少个表头节点，多少个表节点？

（2）对于有n个顶点e条边的有向图，邻接表表示时有多少个表头节点，多少个表节点？