

6. 图

邻接矩阵

构思

邓俊辉

deng@tsinghua.edu.cn

Graph模板类

```
❖ template <typename Tv, typename Te> class Graph { //顶点类型、边类型
private:
    void reset() { //所有顶点、边的辅助信息复位
        for ( int i = 0; i < n; i++ ) { //顶点
            status(i) = UNDISCOVERED; dTime(i) = fTime(i) = -1;
            parent(i) = -1; priority(i) = INT_MAX;
            for ( int j = 0; j < n; j++ ) //边
                if ( exists(i, j) ) type(i, j) = UNDETERMINED;
        }
    }

public:    /* ... 顶点操作、边操作、图算法： 无论如何实现，接口必须统一 ... */
} //Graph
```

邻接矩阵 + 关联矩阵

❖ **adjacency matrix** : 用二维矩阵记录**顶点**之间的**邻接**关系

——对应：矩阵元素 \Leftrightarrow 图中可能存在的边

$A(i, j) = 1$ 若顶点*i*与*j*之间存在一条边
 $= 0$ 否则

既然只考察简单图，对角线统一设置为0

空间复杂度为 $\Theta(n^2)$ ，与图中实际的边数无关

❖ **incidence matrix** : 用二维矩阵记录**顶点与边**之间的**关联**关系

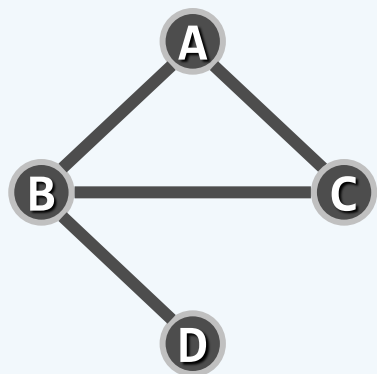
空间复杂度为 $\Theta(n \cdot e) = O(n^3)$

空间利用率 $< 2/n$

解决某些问题时十分有效

实例

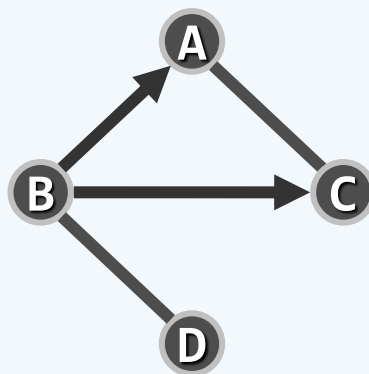
(a) undigraph



\emptyset	A	B	C	D
A		1	1	
B	1		1	1
C	1	1		
D		1		

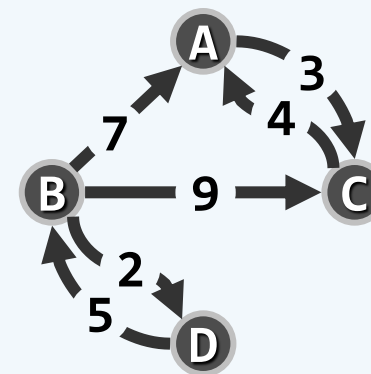
↖ redundancy

(b) digraph



\emptyset	A	B	C	D
A			1	
B	1		1	1
C	1			
D		1		

(c) network



∞	A	B	C	D
A			3	
B	7		9	2
C	4			
D		5		