

Algorithm&Data Structure

— The road to NB

张海鹏 2017-04-03

算法(基础)体系

递归与分治：二分查找，合并排序，快速排序，堆排序

动态规划：矩阵连乘，最长公共子序列，0-1背包

贪心算法：最优装载，哈夫曼编码，单源最短路径，MST

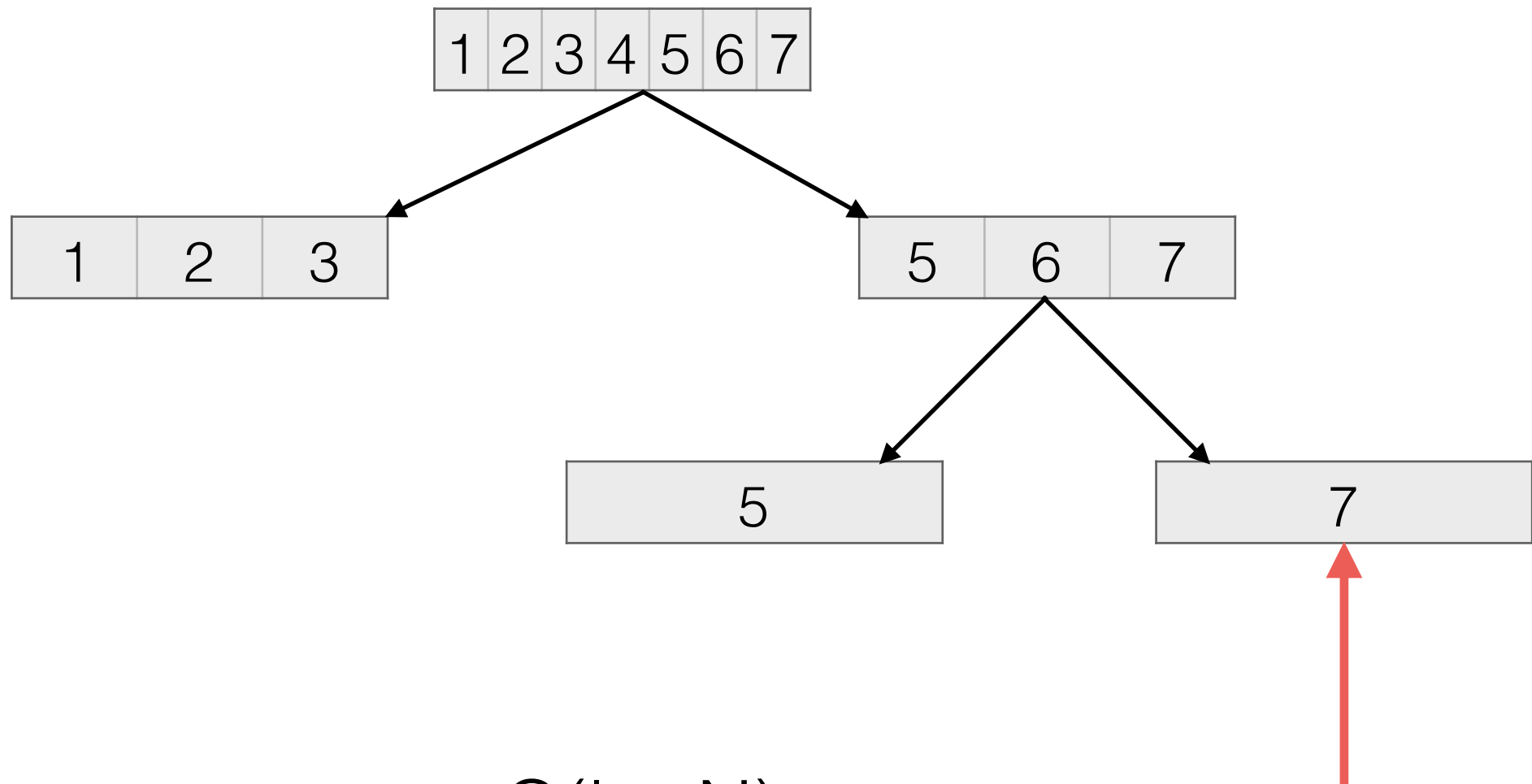
回溯法：旅行售货员

分支限界法：单源最短路径，0-1背包

随机化：Sherwood，Las Vegas，Monte Carlo

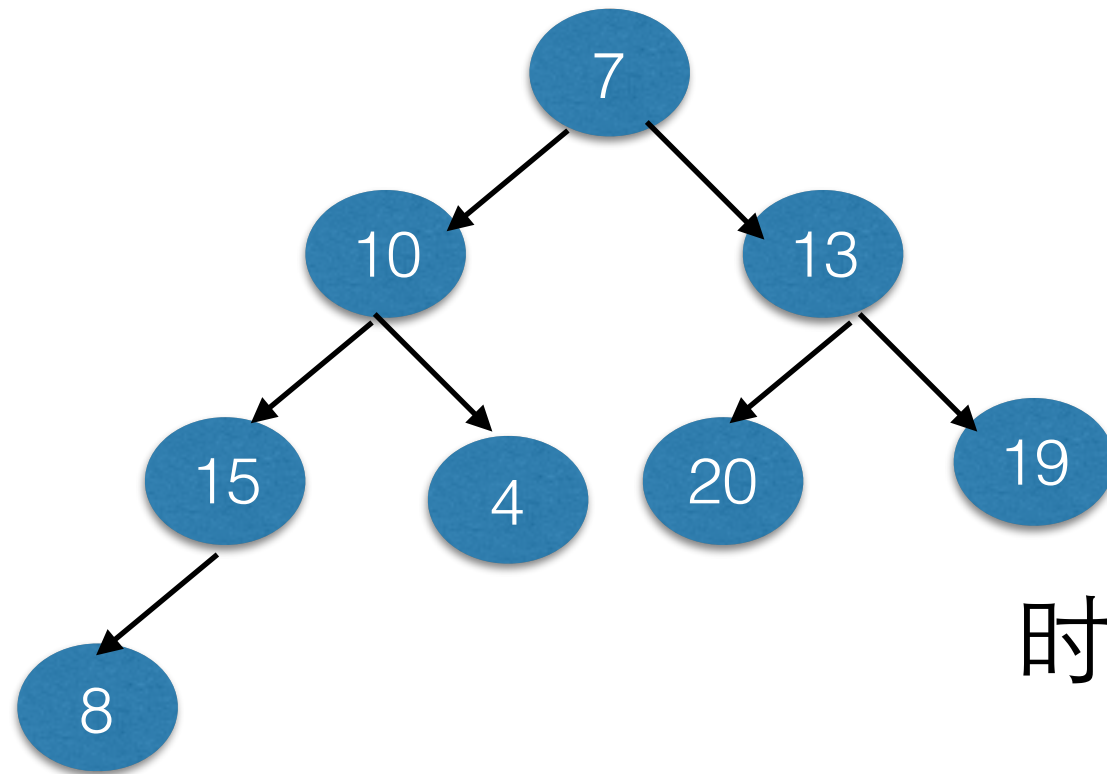
线性规划与网络流：线性规划与单纯形，最大网络流

二分查找



$O(\log N)$

堆排序



7	10	13	15	4	20	19	8
---	----	----	----	---	----	----	---

时空复杂度: $O(N \log N)$ + $O(1)$

稳定性: 不稳定


树

- 二叉树的基本形态，满/完全,性质证明
 - 二叉树的遍历(前，中，后)
 - 树与森林
 - 平衡树
-

二叉树

满二叉树：左右子树存在，所有叶子节点同层

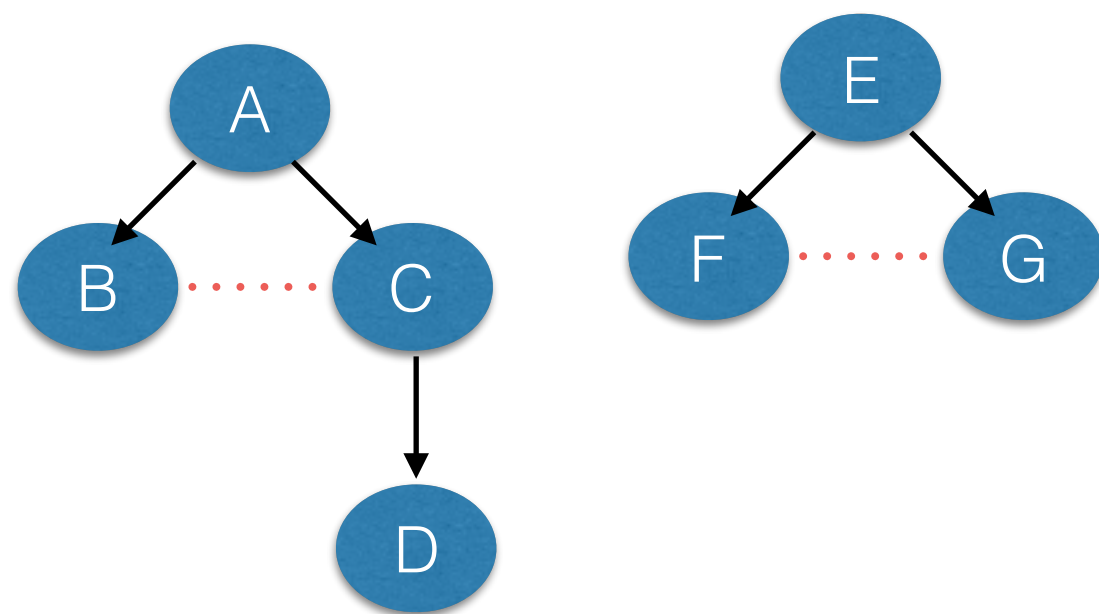
完全二叉树：与满二叉树编号相同的节点在二叉树中的位置相同

关系： 满二叉树  完全二叉树

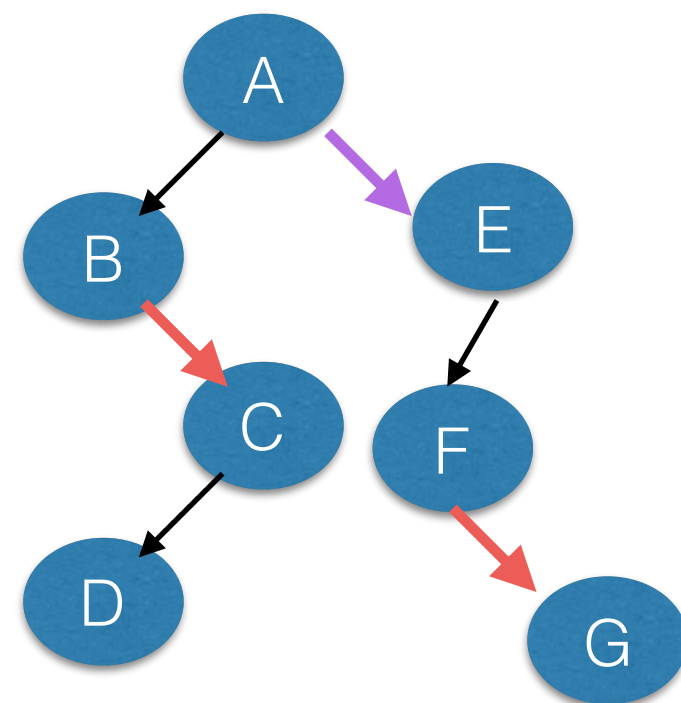
二叉树

```
void InOrder(BiTree bt){  
    if(bt != NULL){  
        Inorder(bt->lchild);  
        disp(bt->data);  
        Inorder(bt->rchild);  
    }  
}  
  
typedef struct BiTNode{  
    char data;  
    struct BiTNode *lchild,*rchild;  
}BiTNode,*BiTree;
```

森林



森林



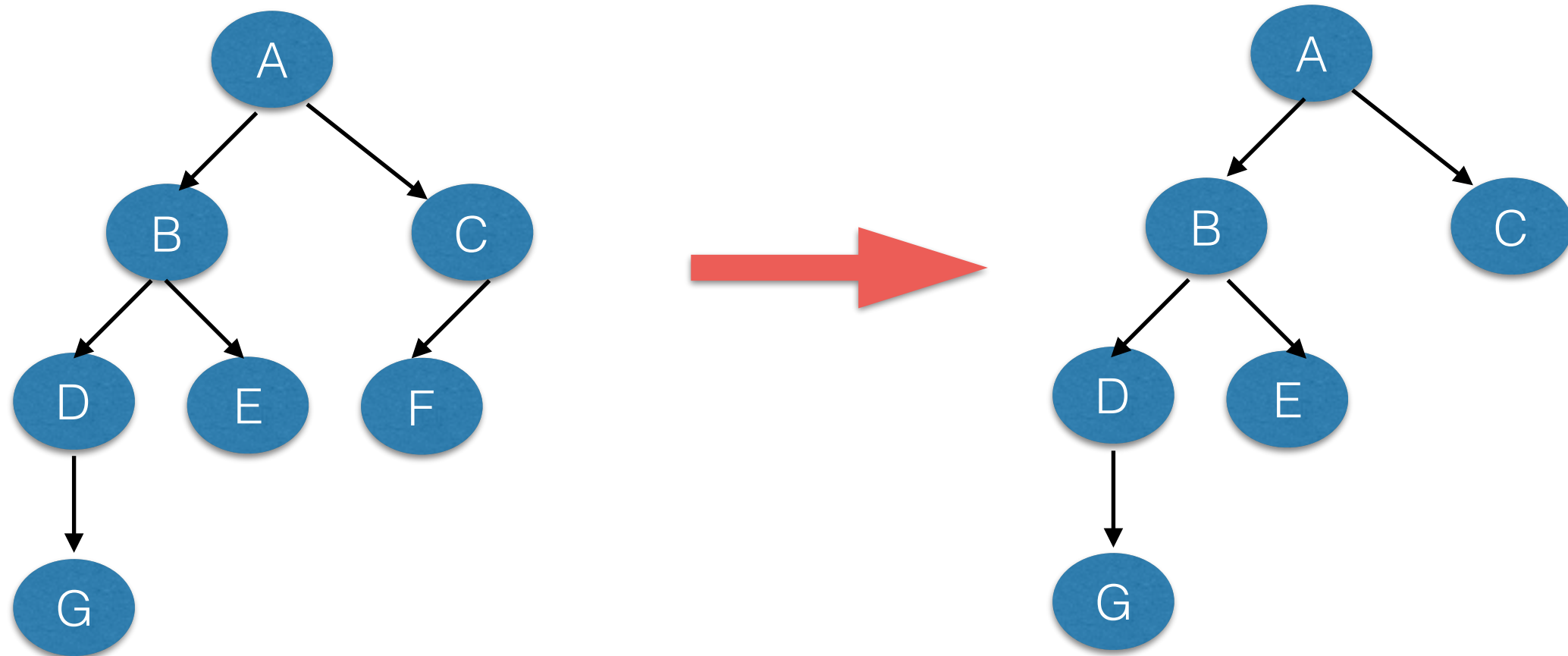
二叉树

平衡树(AVL)

intuition: 在二叉排序树中，如果插入元素的顺序接近有序，则二叉排序树退化为链表，导致二叉排序树的查找效率降低。

definition: 一棵空树或者是具有下列性质的二叉树：
左右子树都是平衡二叉树，且左子树和右子树的高度之差的绝对值不超过1

AVL



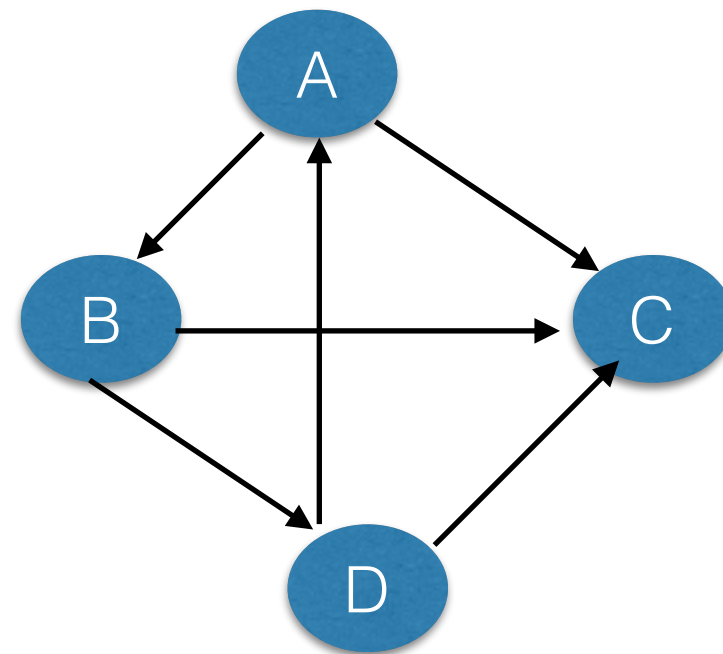
图

- 图的存储
 - 图的遍历(DFS,BFS)
 - 最小生成树MST
 - 最短路径
-

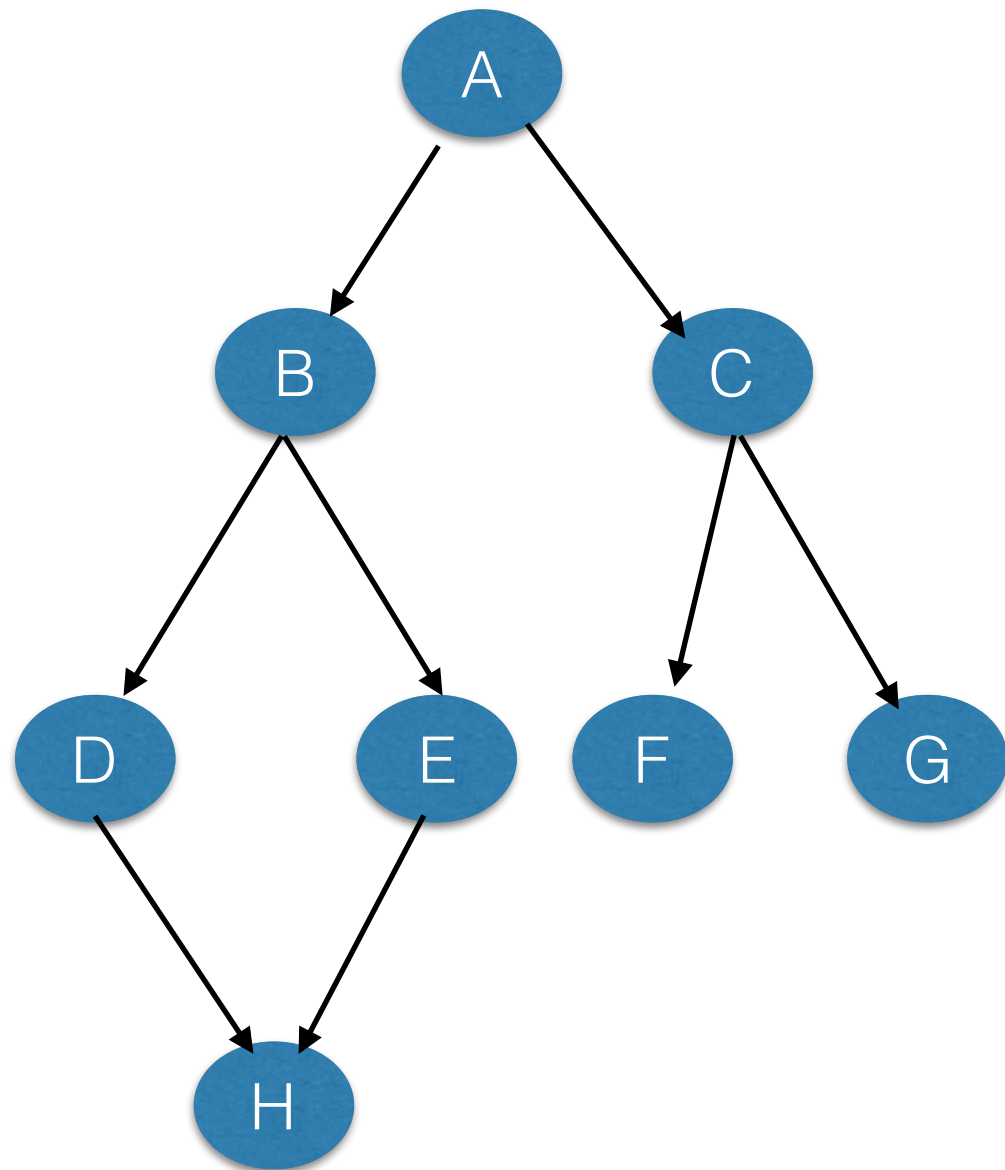
图的存储

邻接矩阵: 二维数组

邻接表: 两个单链表



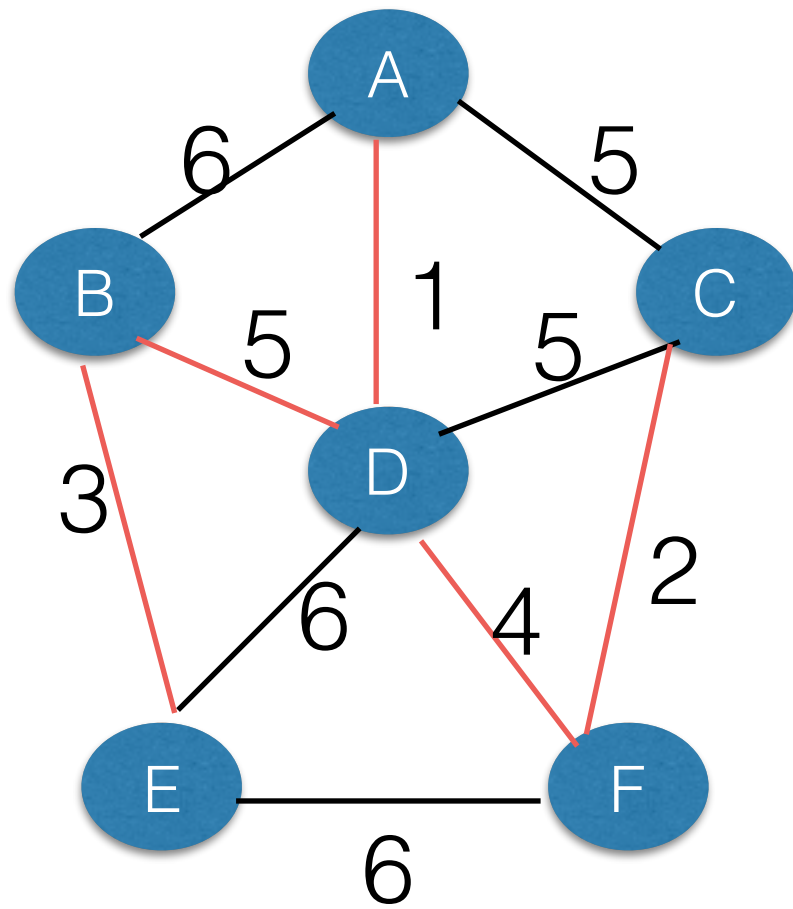
图的遍历



DFS: 栈

BFS: 队列

MST



生成树：任意两个顶点之间有且仅有一条路径，如果再增加一条边就会出现环路，如果去掉一条边就会变成非连通图

Prim算法

Kruskal算法

查找

类别	平均时间复杂度	查找条件	算法描述
顺序查找	$O(N)$	元素有序(无序)	顺序比较
二分查找	$O(\log N)$	有序数组	折半查找
分块查找	$O(\log N)$	按块有序	二分+顺序
二插排序树	$O(\log N)$	树型结构	按树查找
哈希	$O(1)$	哈希表	KV查找

排序

各种常用排序算法						
类别	排序方法	时间复杂度			空间复杂度	稳定性
		平均情况	最好情况	最坏情况	辅助存储	
插入排序	直接插入	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	稳定
	shell排序	$O(n^{1.3})$	$O(n)$	$O(n^2)$	$O(1)$	不稳定
选择排序	直接选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
	堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	不稳定
交换排序	冒泡排序	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$	稳定
	快速排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n^2)$	$O(n\log_2 n)$	不稳定
归并排序		$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	稳定
基数排序		$O(d(r+n))$	$O(d(n+rd))$	$O(d(r+n))$	$O(rd+n)$	稳定
注：基数排序的复杂度中，r代表关键字的基数，d代表长度，n代表关键字的个数						

图片来源:未知

参考

《剑指Offer》 何海涛

《算法竞赛入门经典》 刘汝佳

《算法艺术与信息学竞赛》 刘汝佳

《算法导论》 T.H Cormen etc.

楼天成@ACRush

July@CSDN

leetcode

hdu etc.
