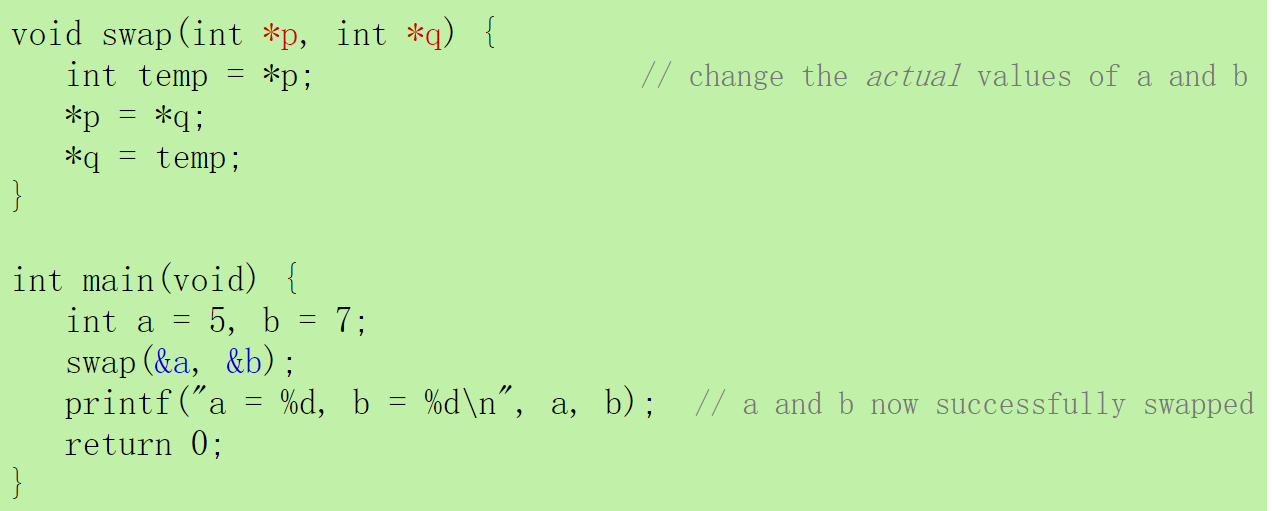


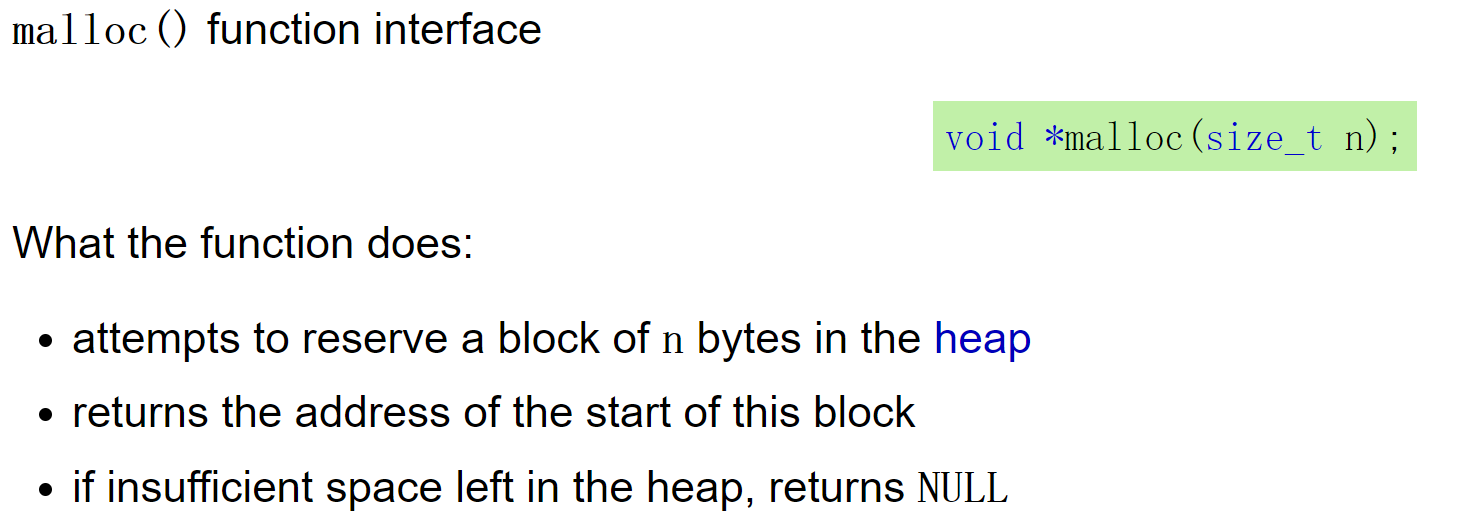
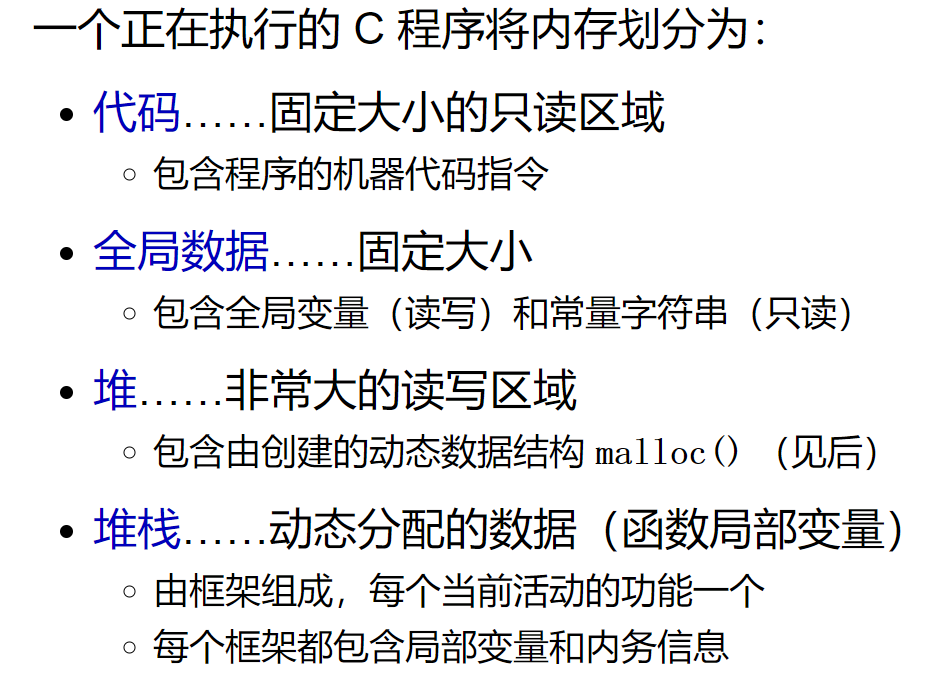
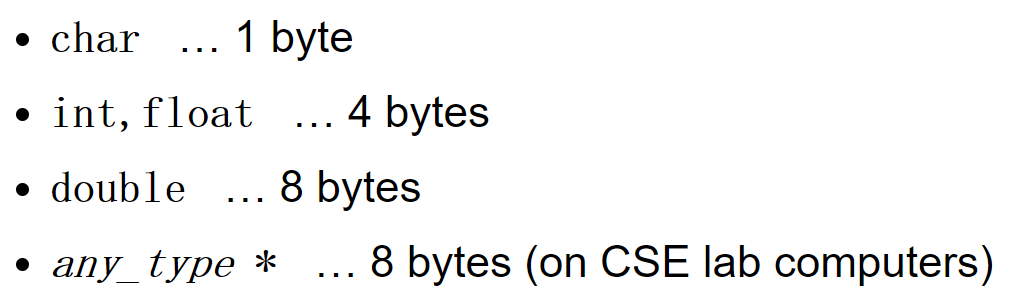
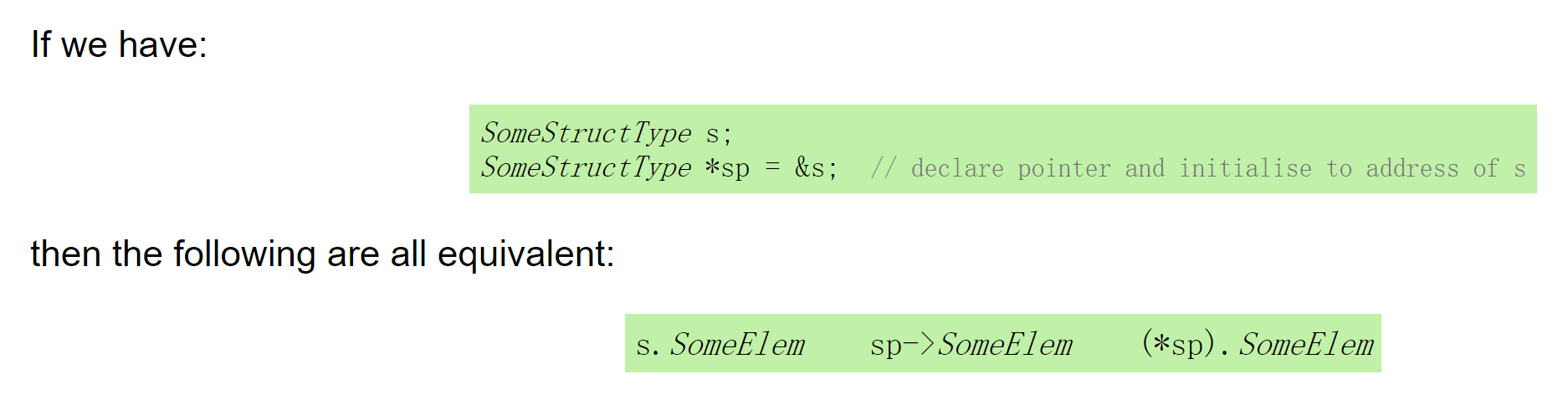
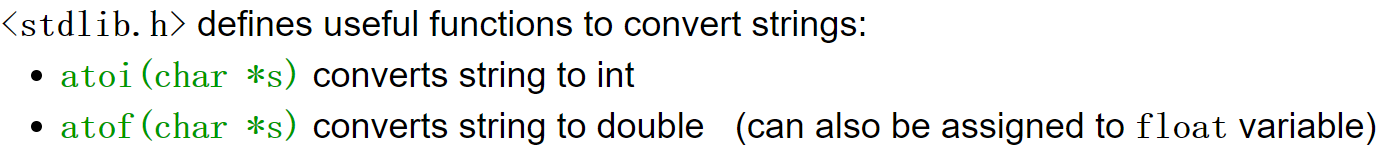
地址的占位符： %p

指针：特殊类型的变量，存储另一个变量的地址（内存位置）

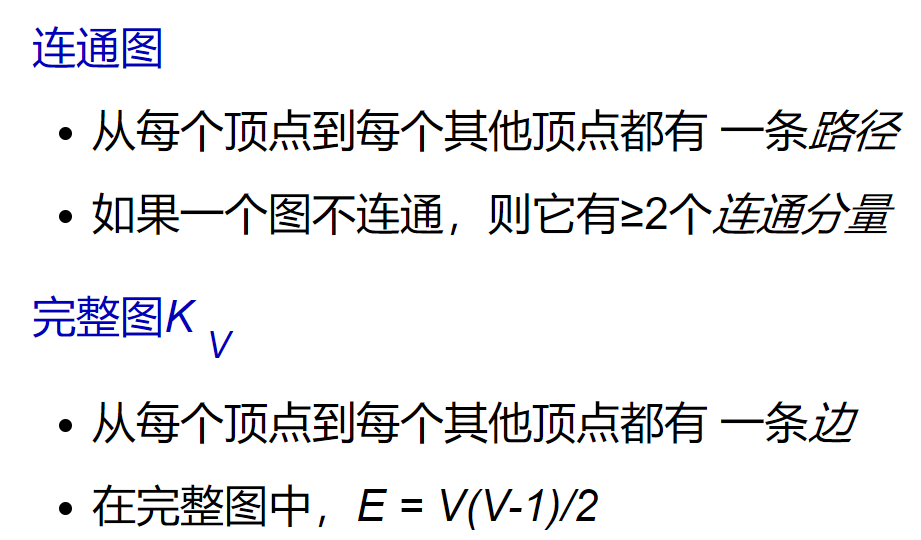
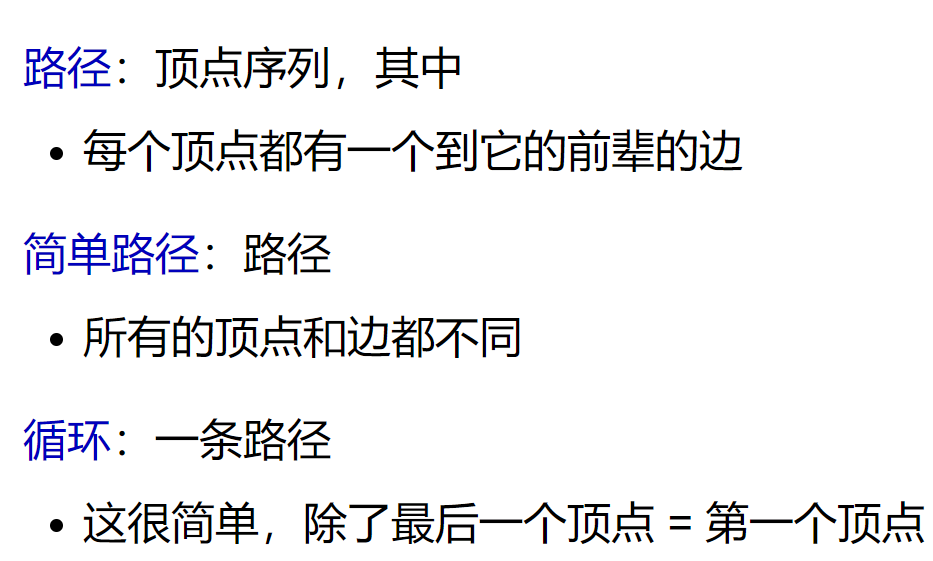
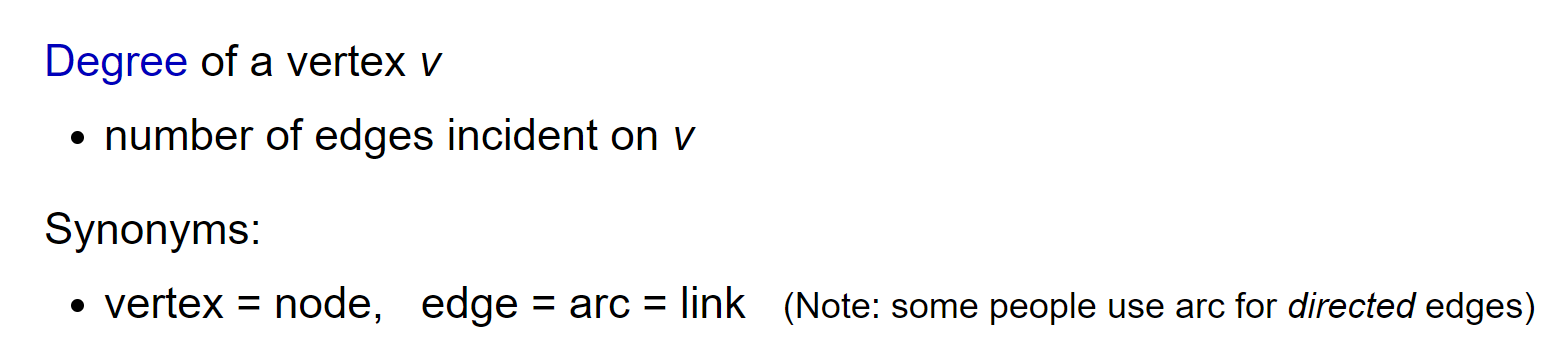
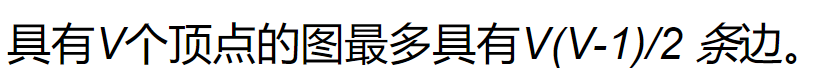
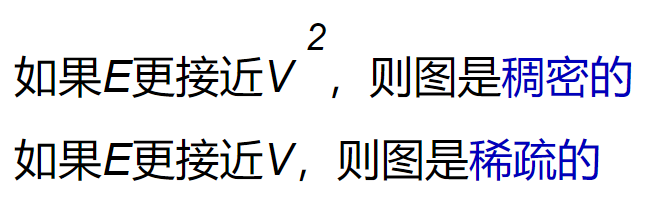
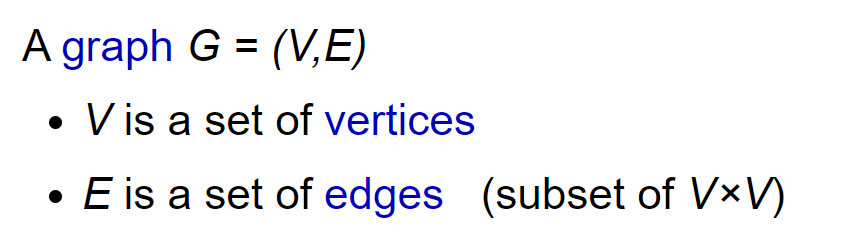


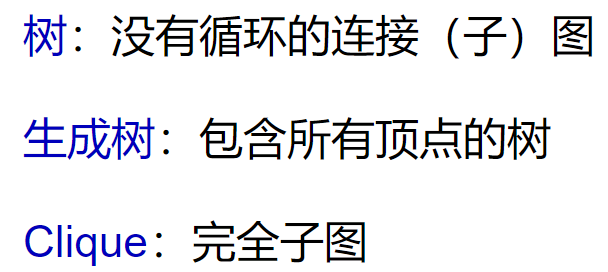
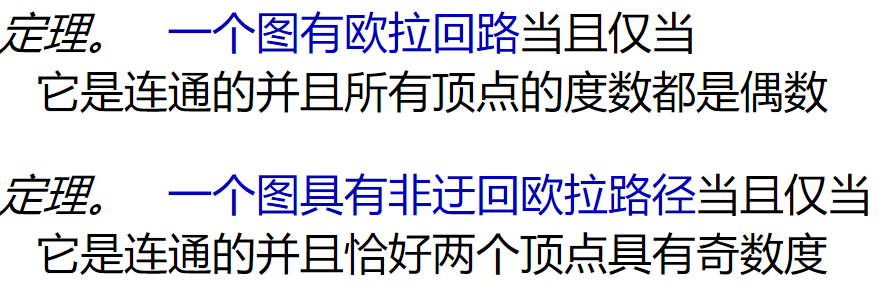
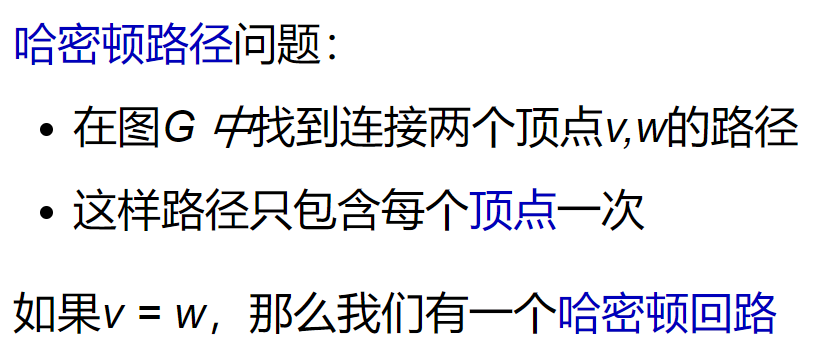
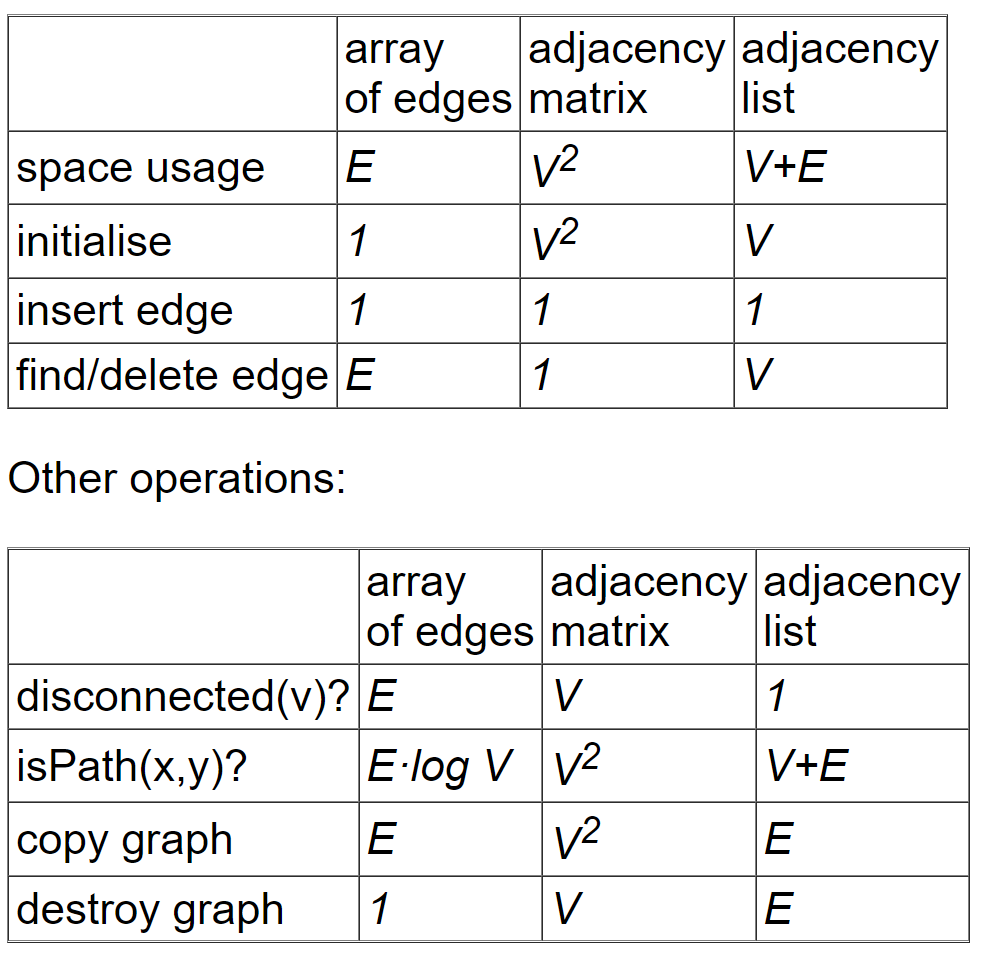
（交换两个数）

命令行参数：argc,argv[](前者为命令行的参数个数，后者为完整的参数数组)



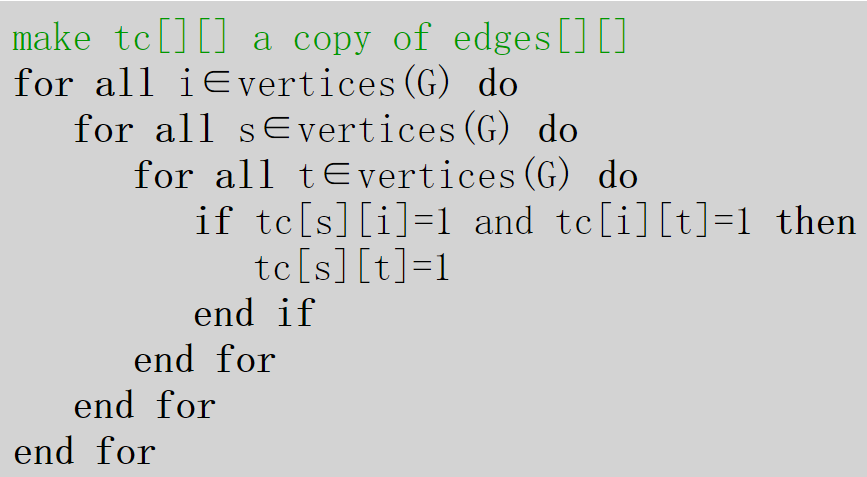
（记得用完要free()）



图可以由边数组，邻接矩阵和邻接表三种方式表示

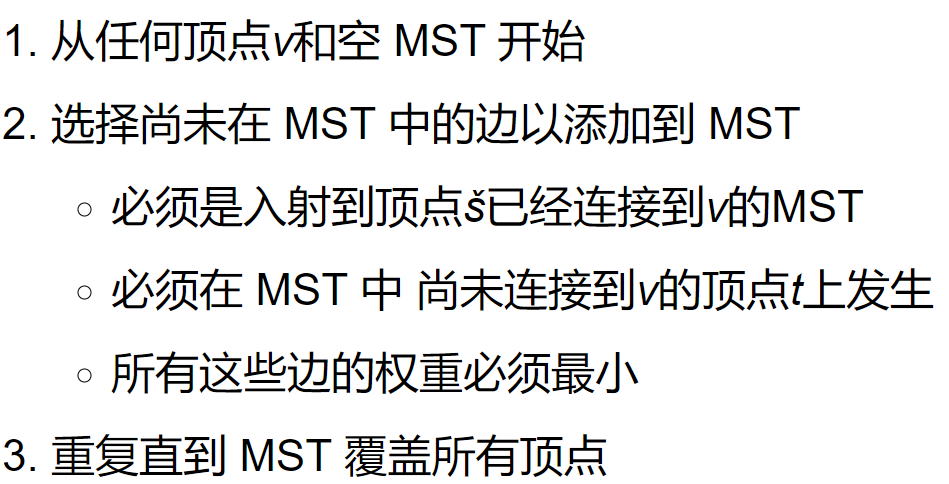
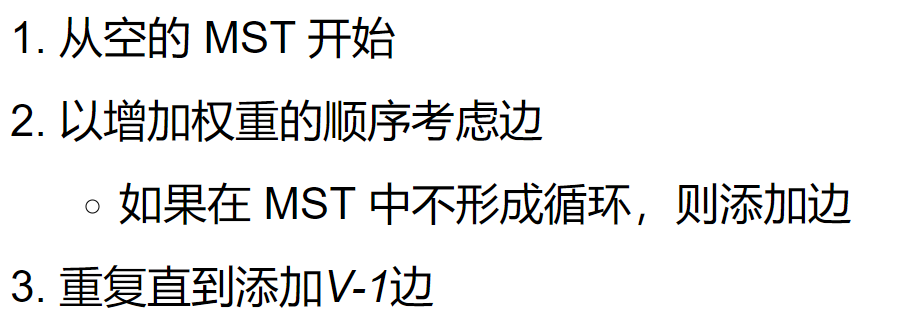
传递闭包（可达性）

Warshall's algorithm



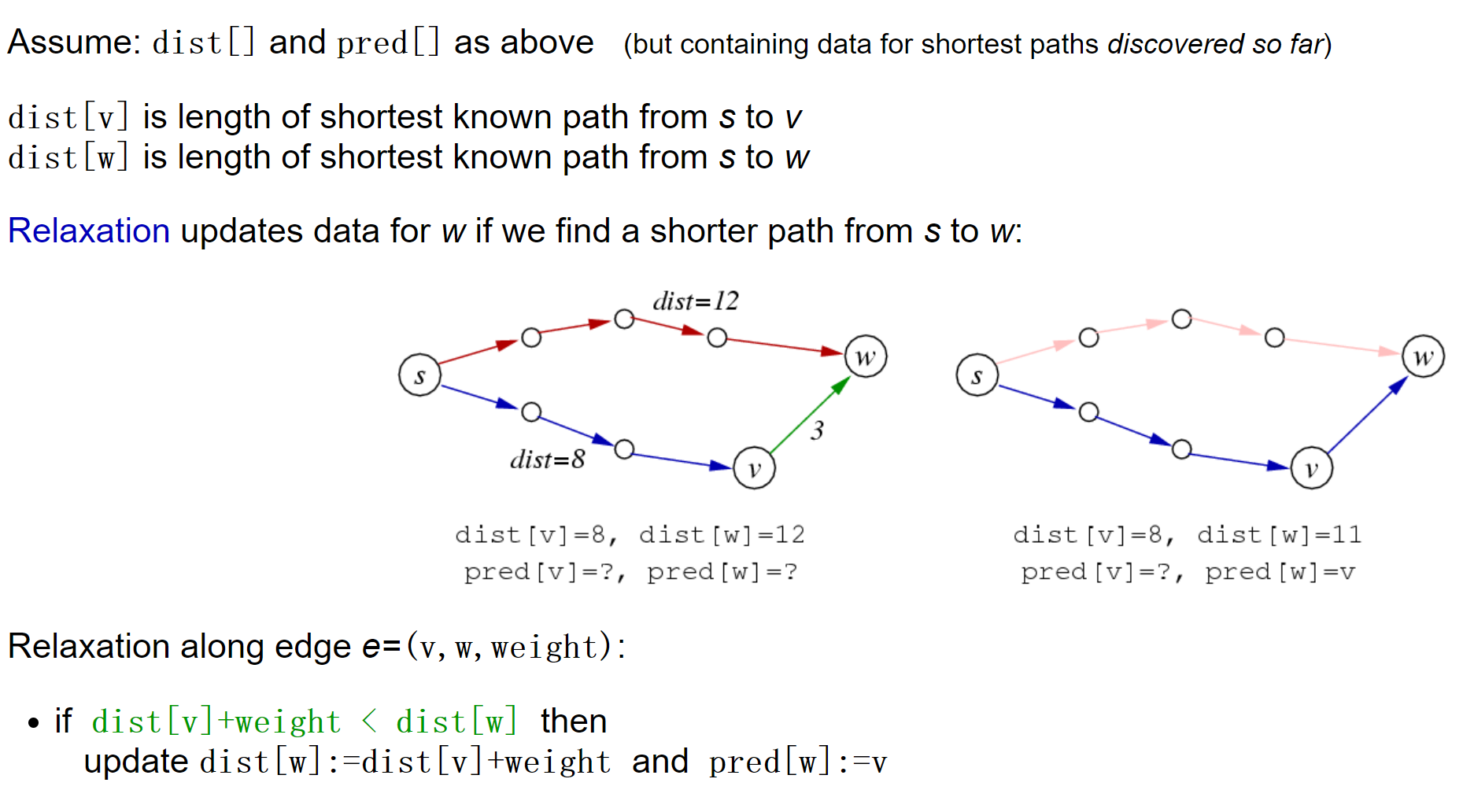
最小生成树问题（MST）

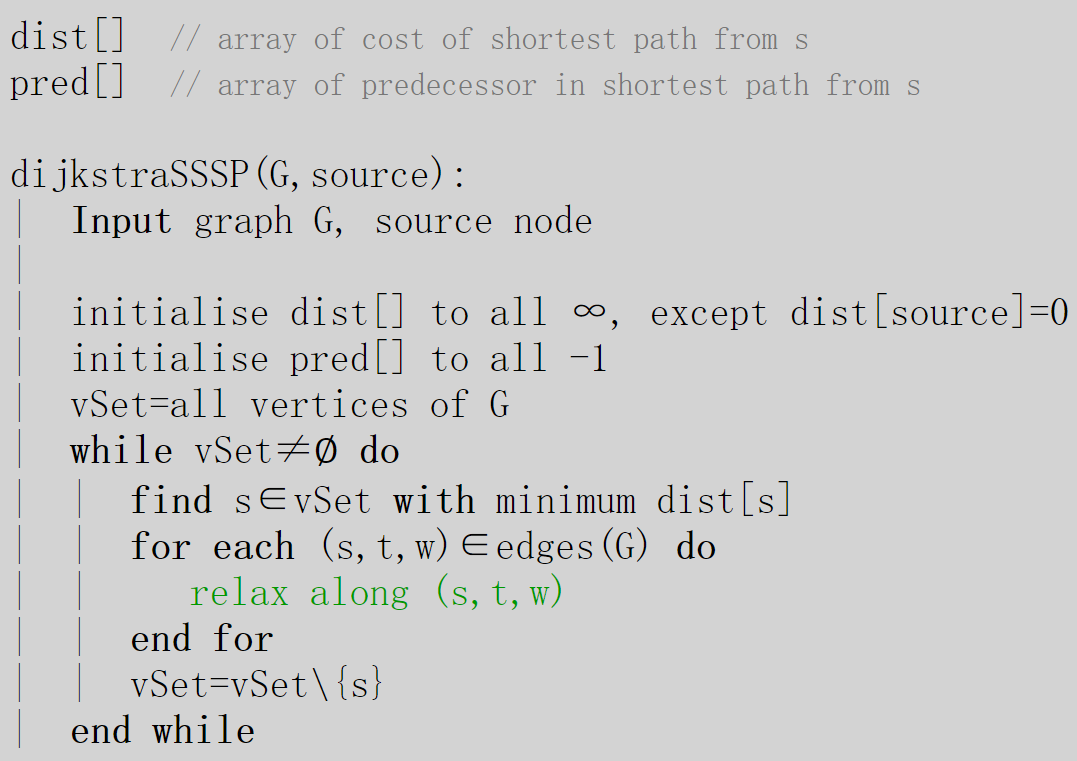
克鲁斯卡尔算法： 普里姆算法：



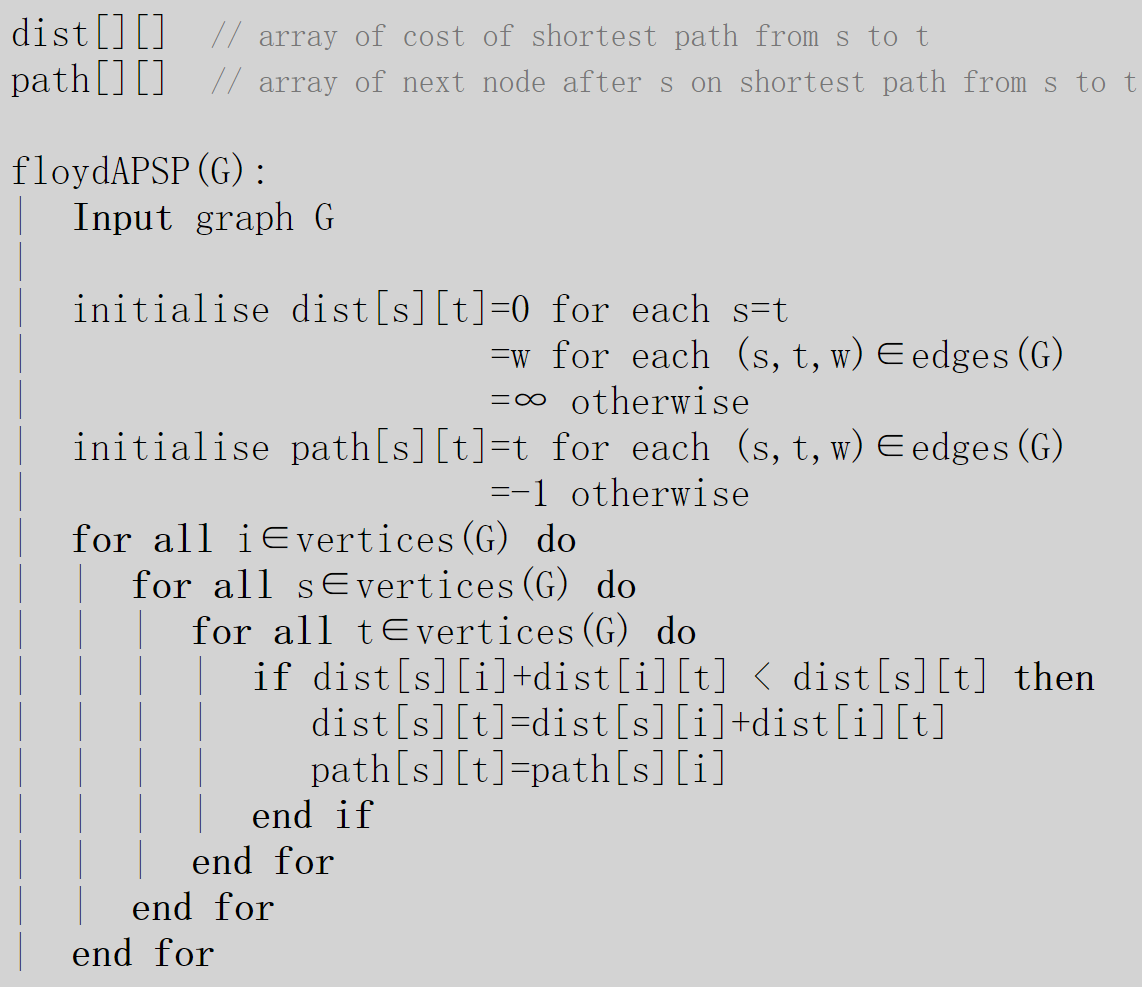
最短路径问题

Dijkstra 算法：



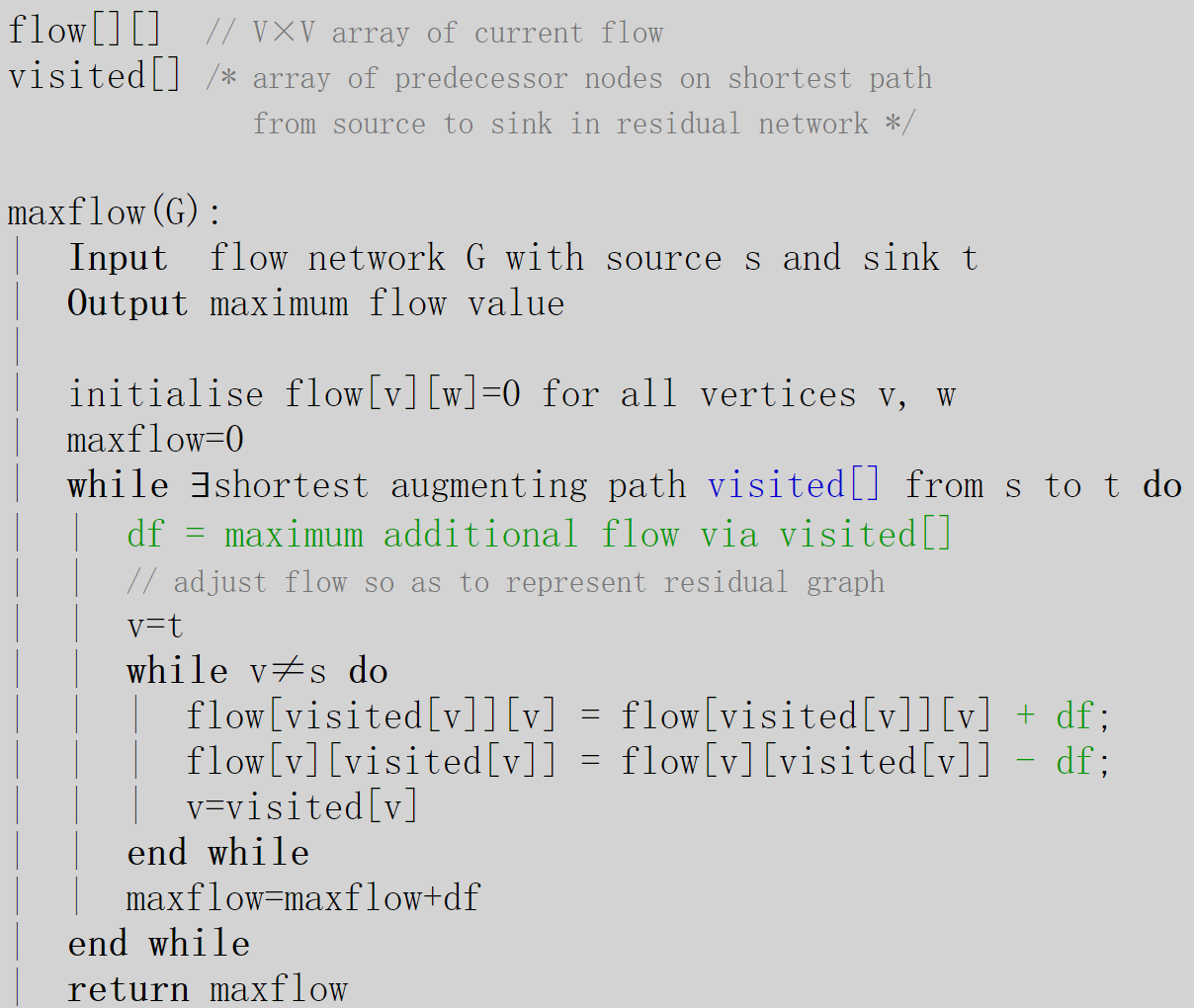


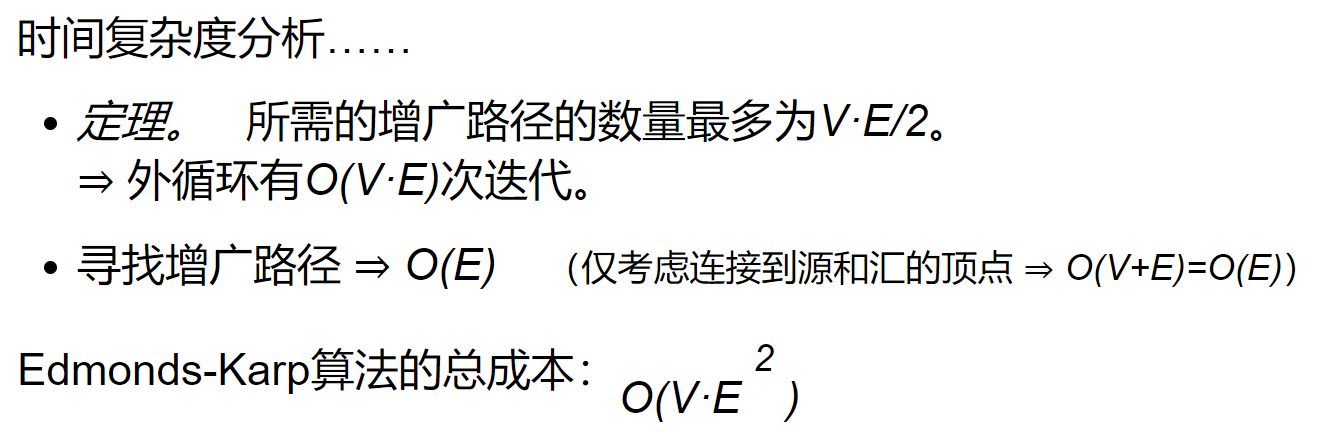
弗洛伊德算法：（解决所有对最短路径问题）



最大流量：

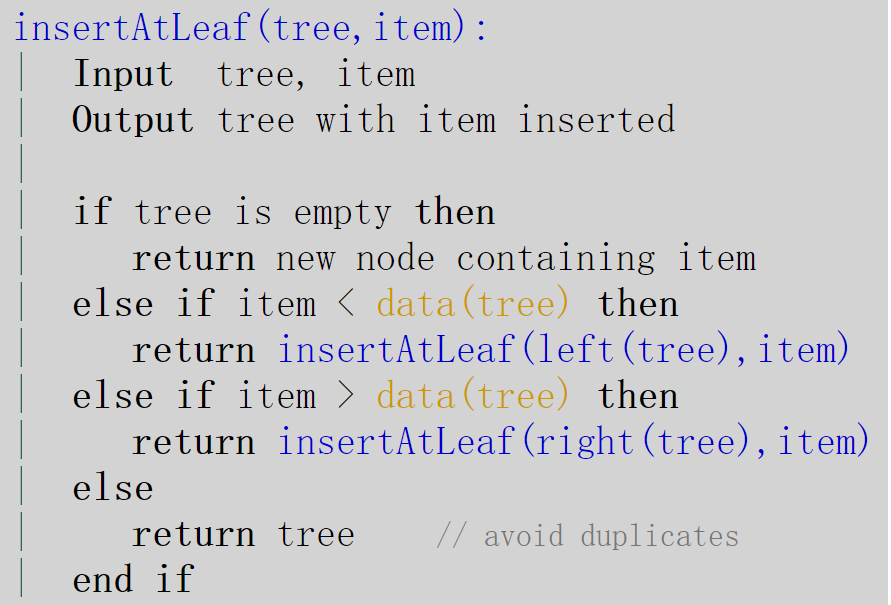
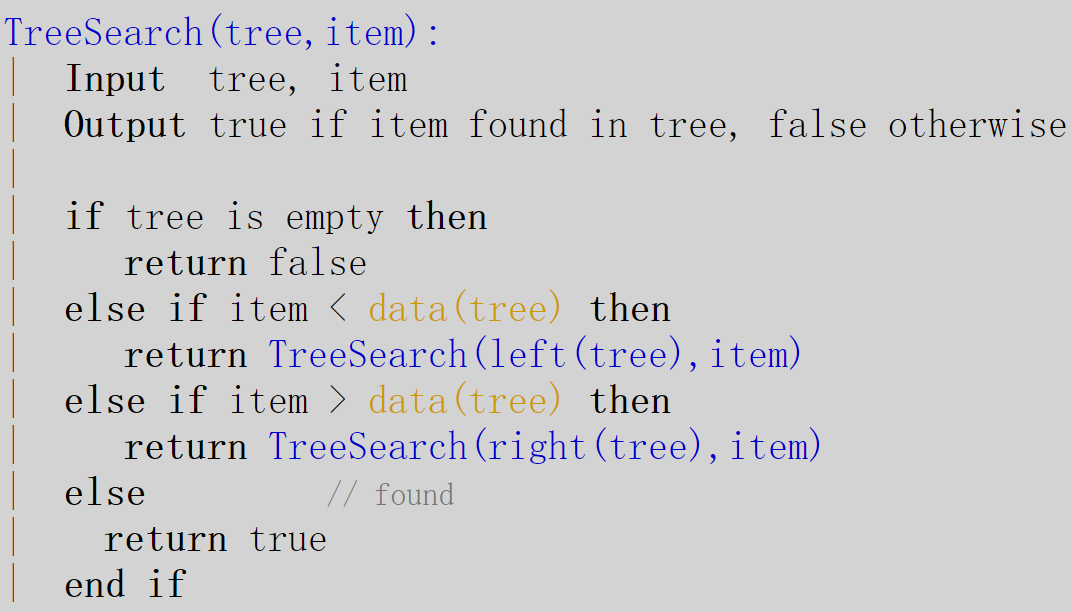
Edmonds-Karp 算法：



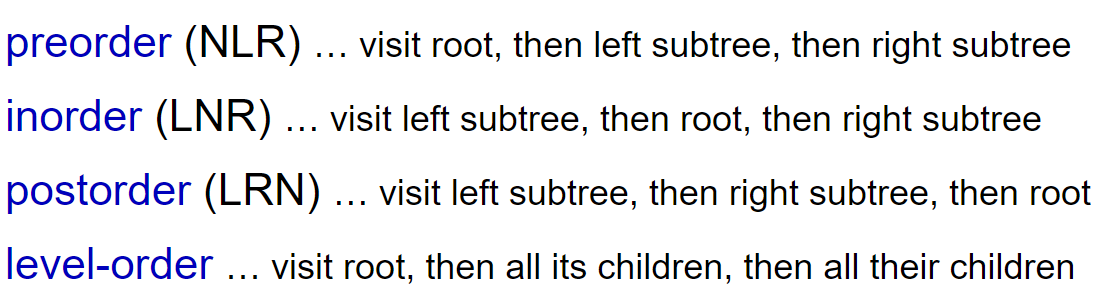




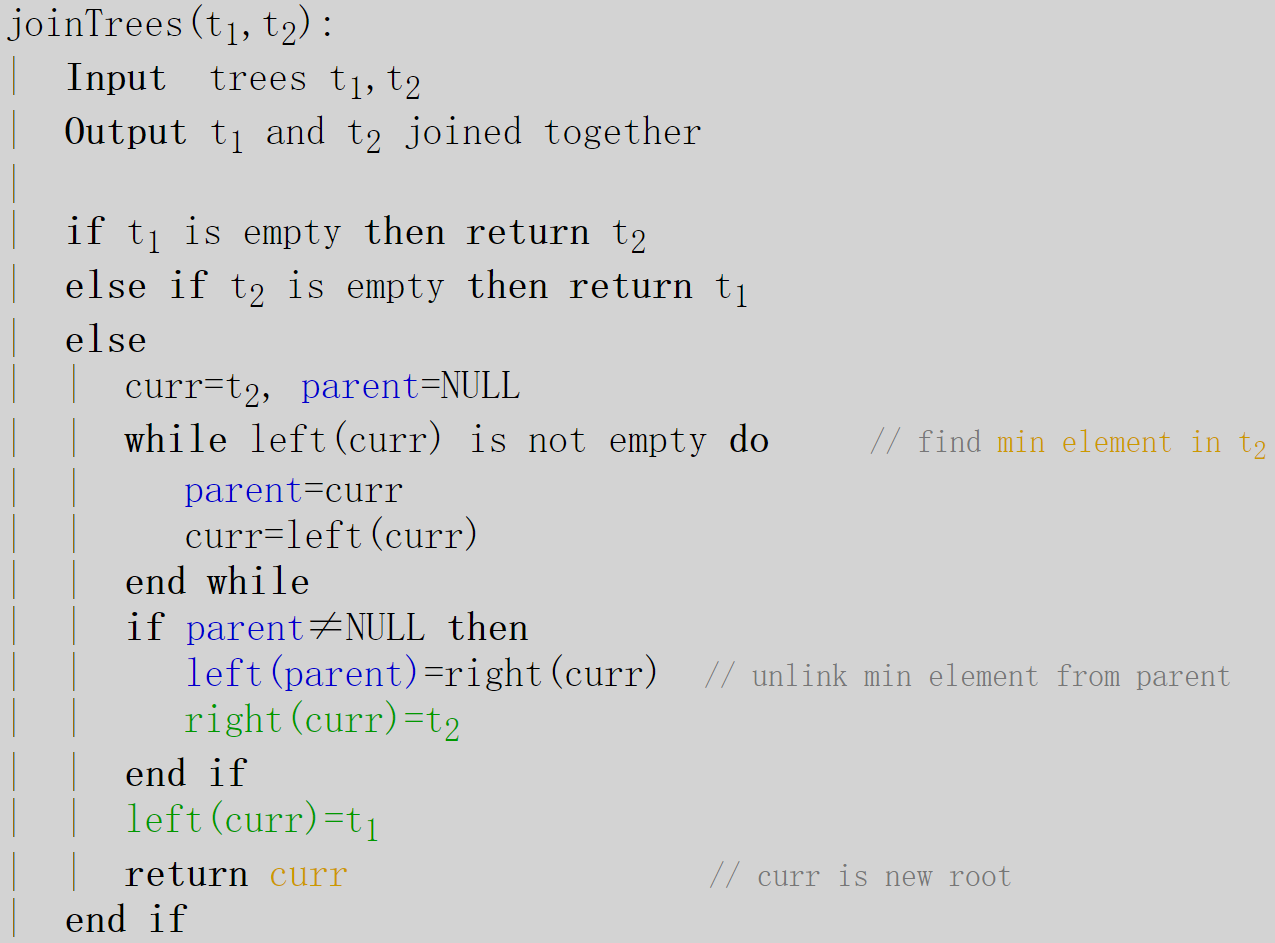
二叉树搜索（递归） 二叉树插入（递归）



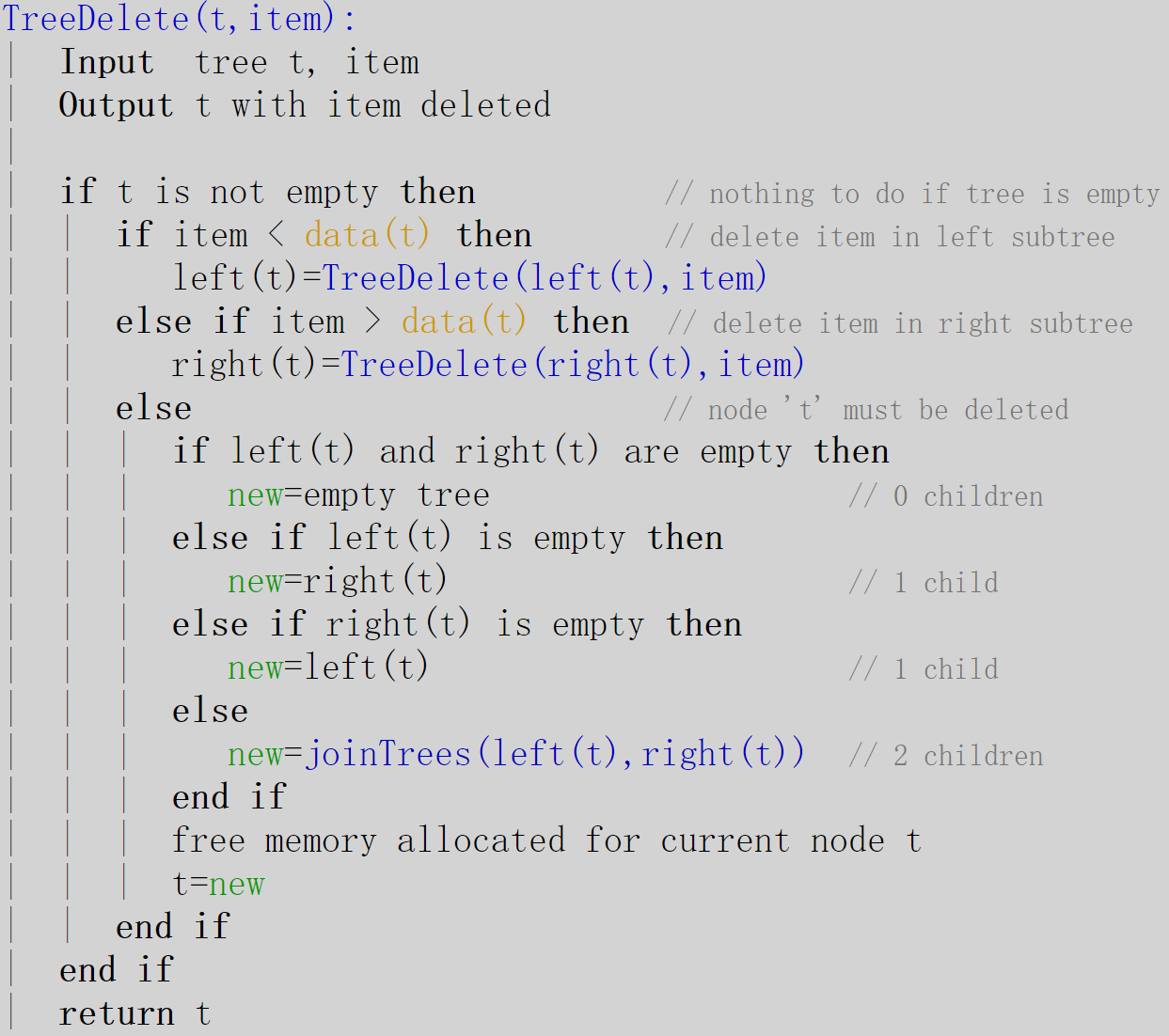
二叉树遍历



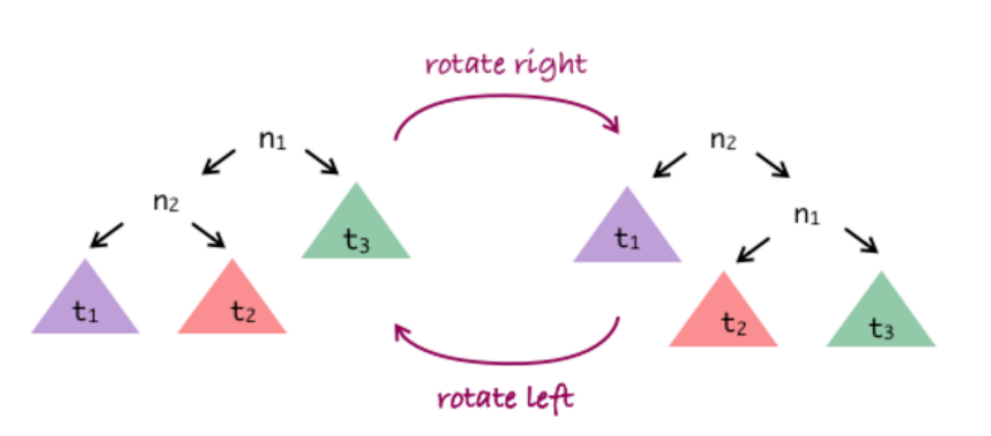
两棵二叉树连接

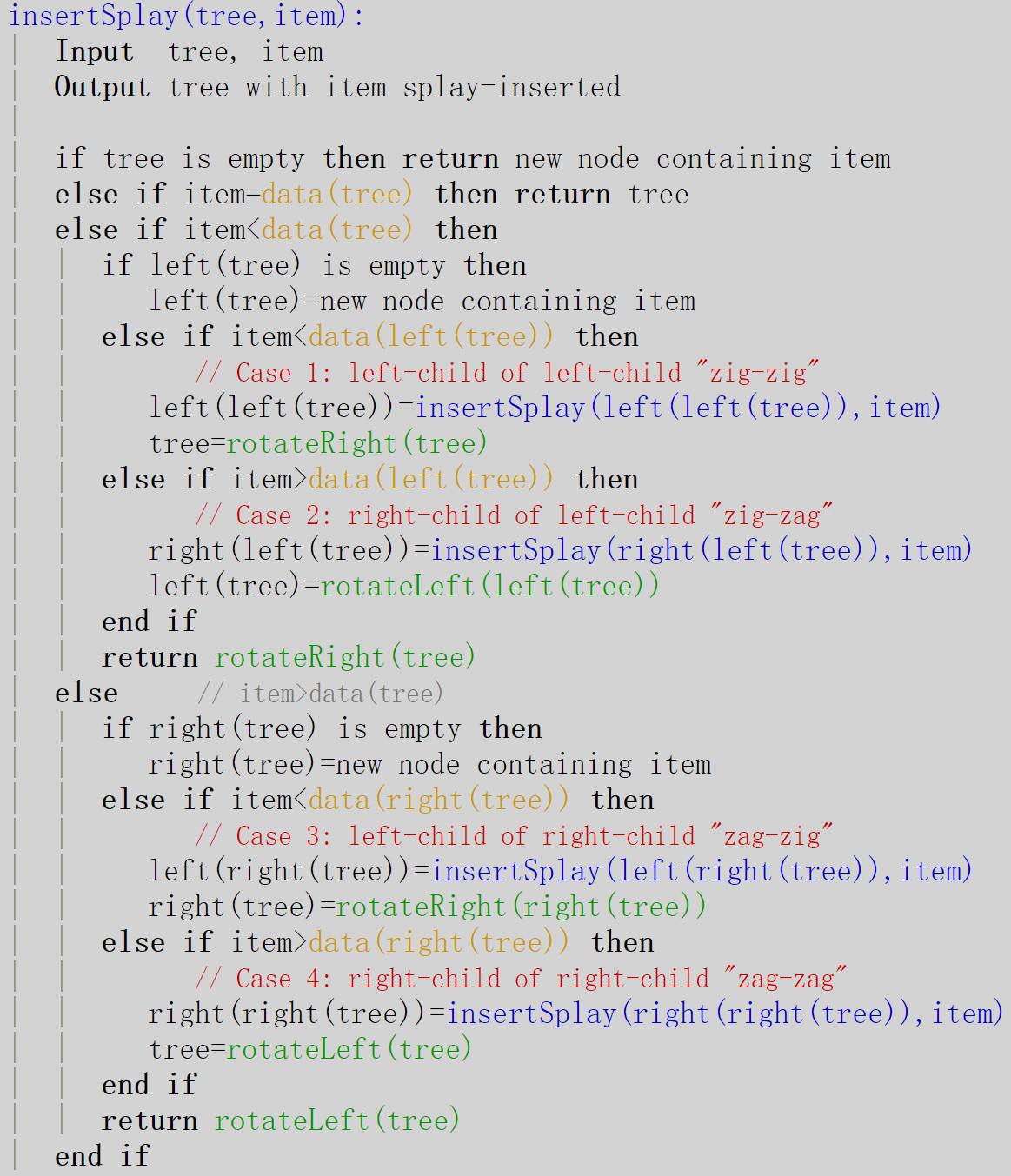
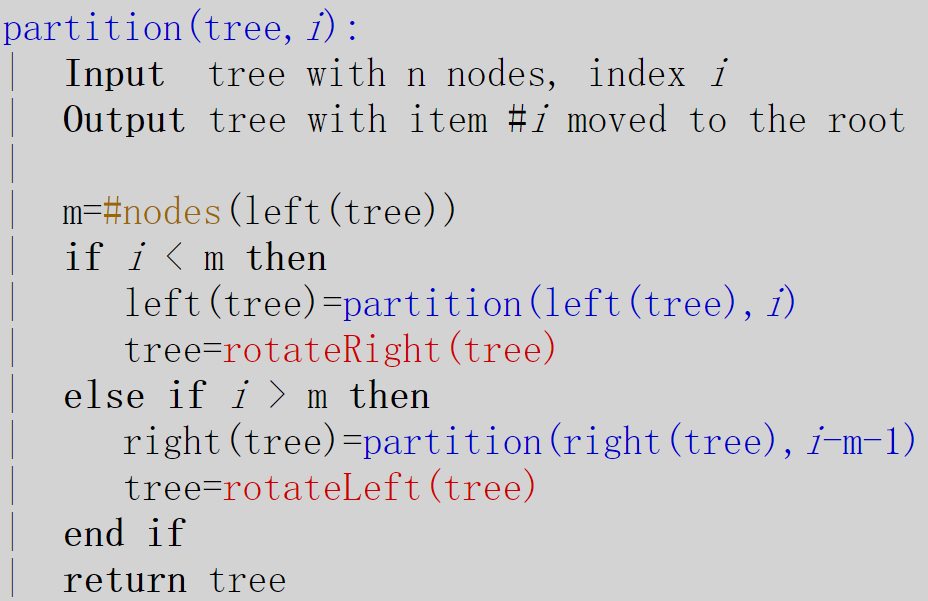


二叉树删除

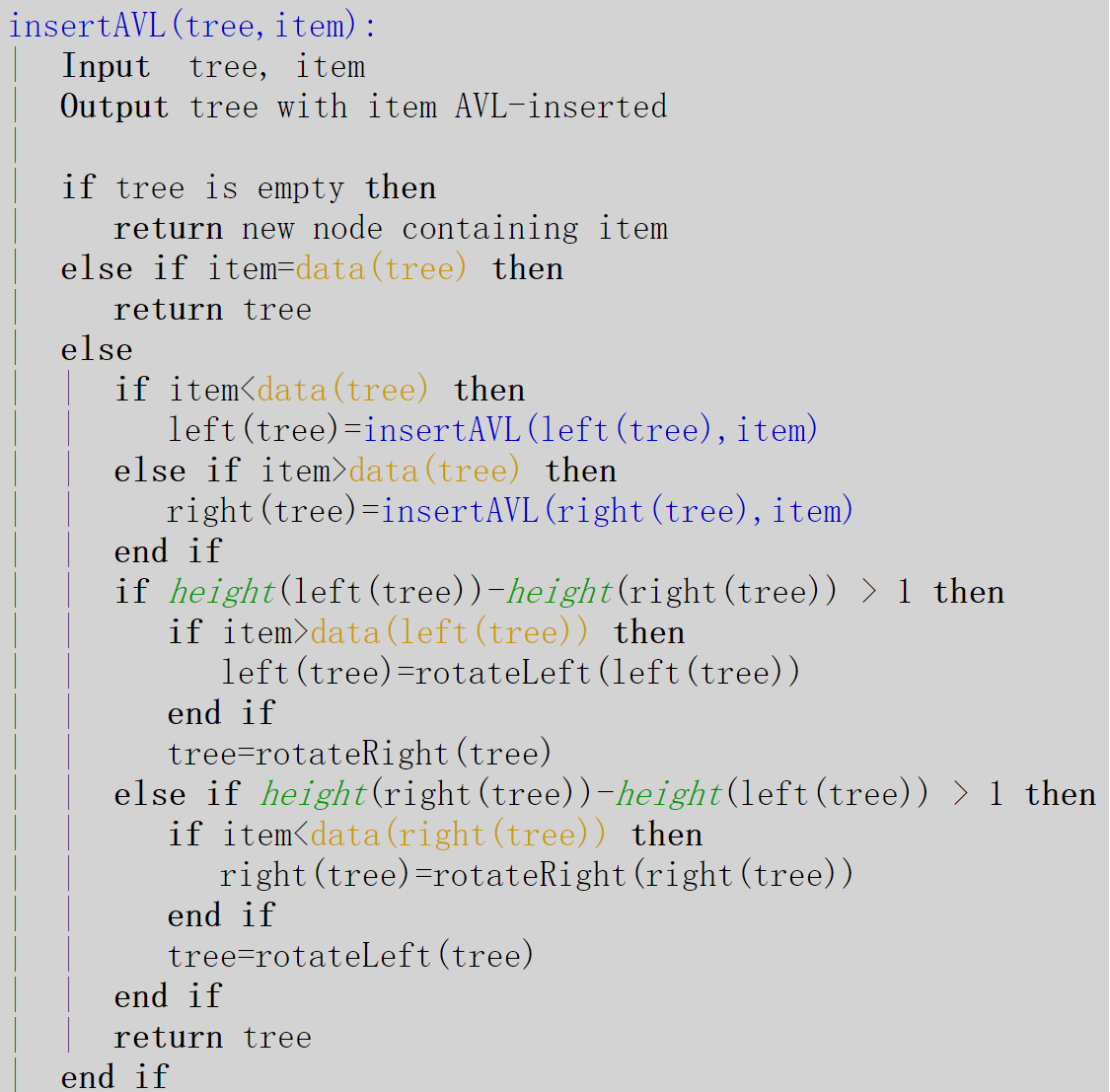
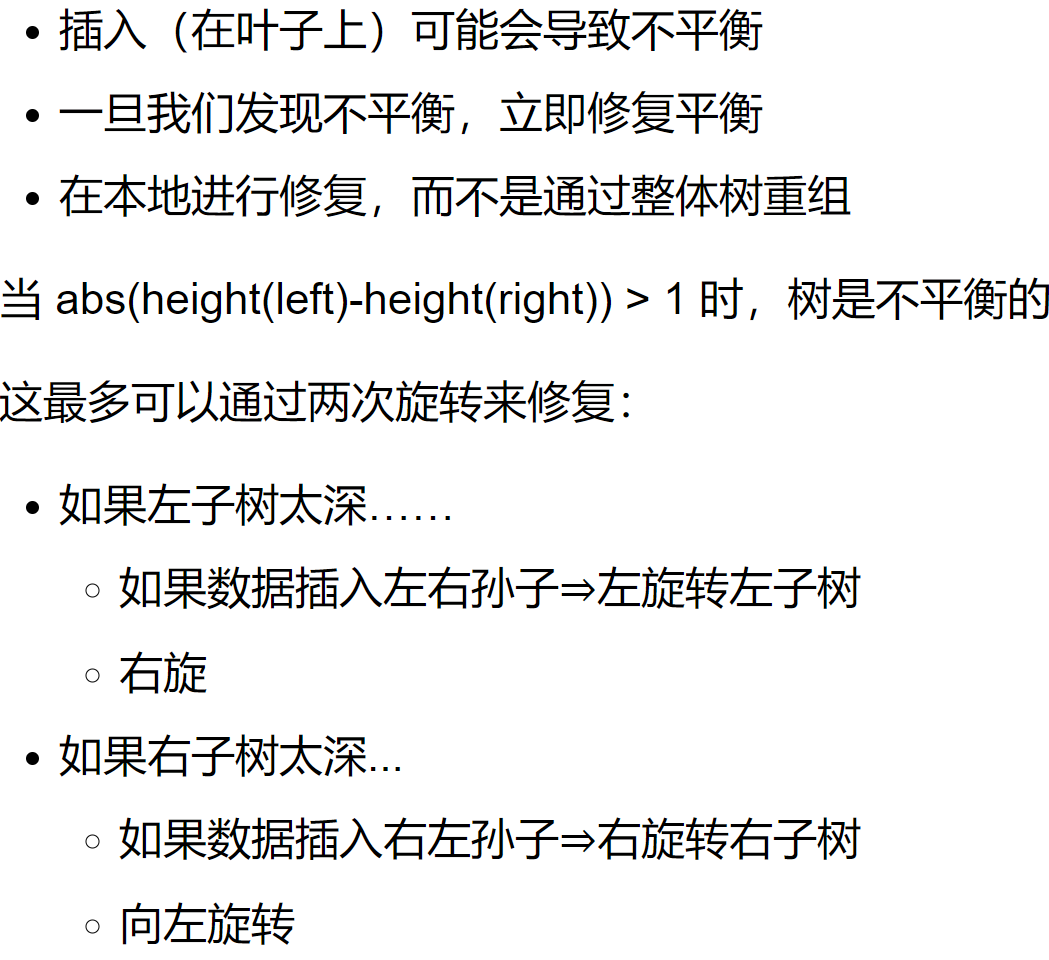


二叉树旋转

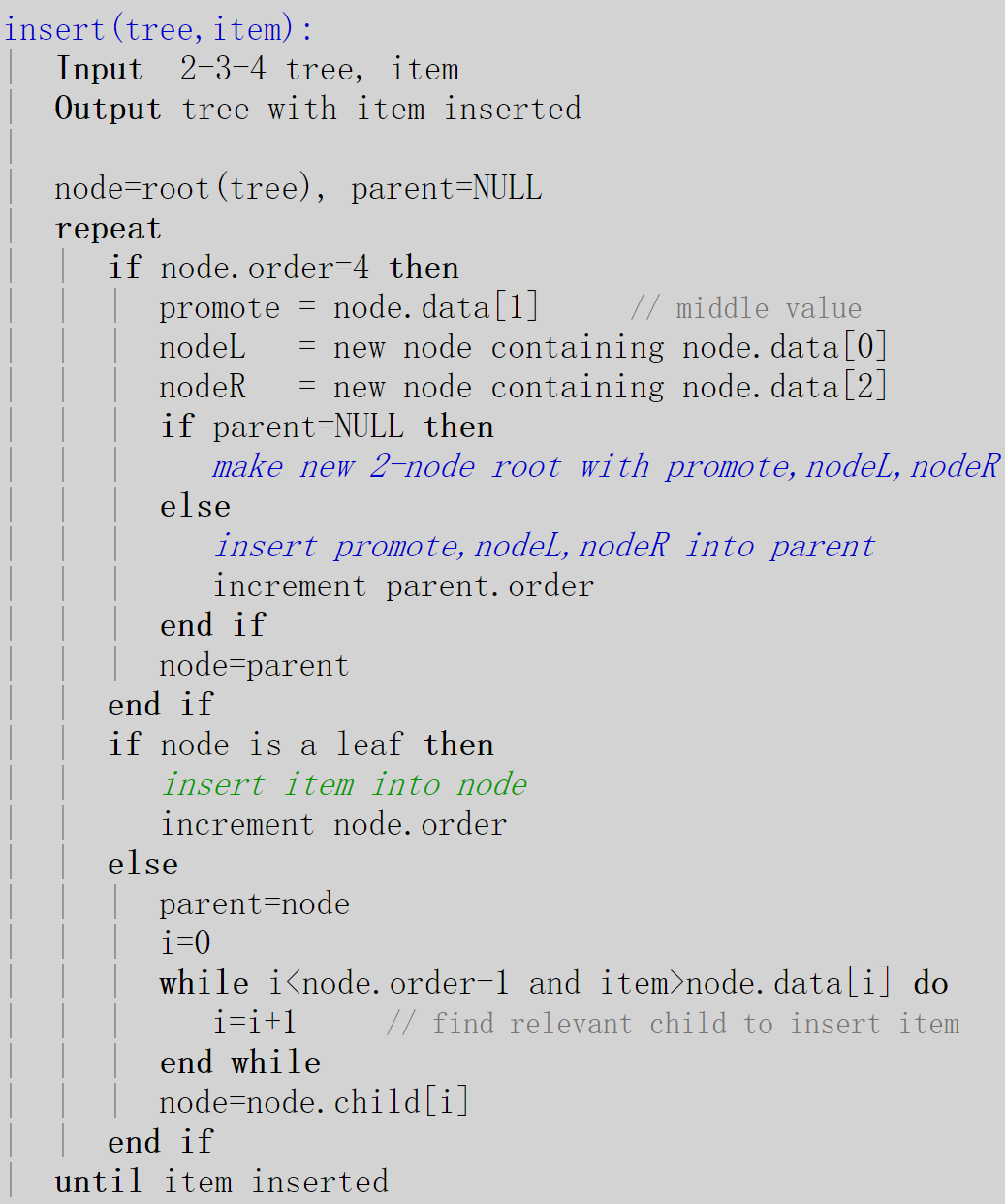
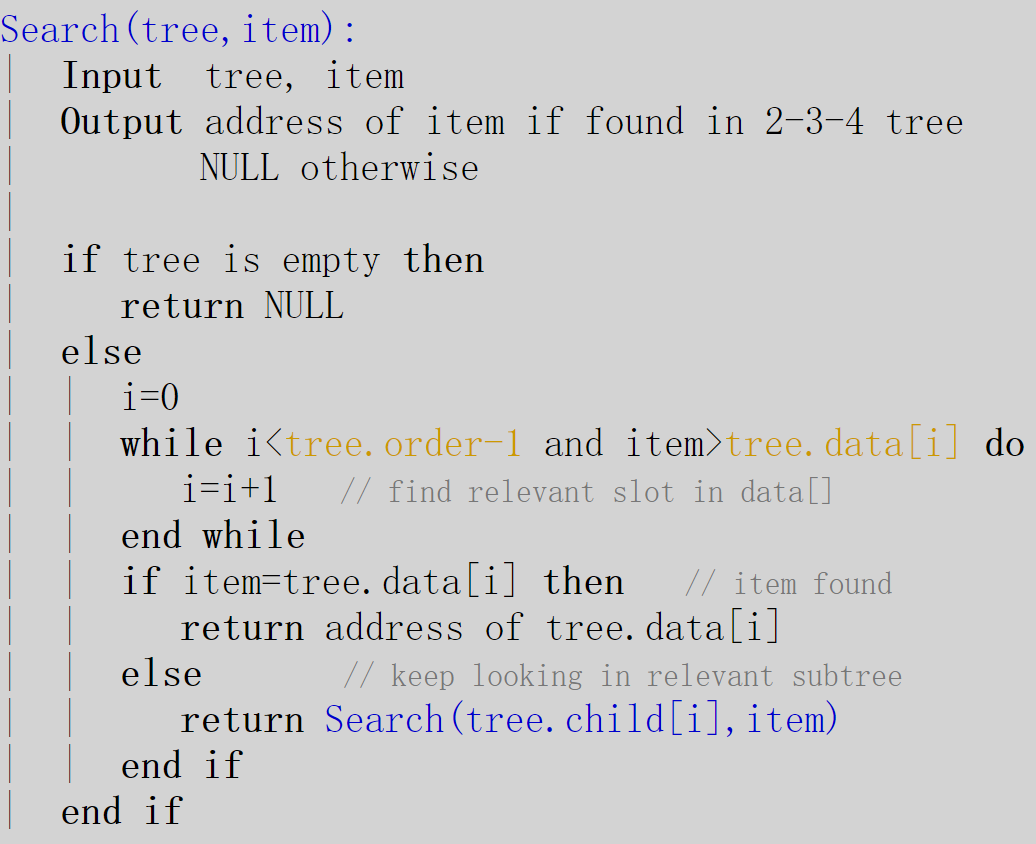




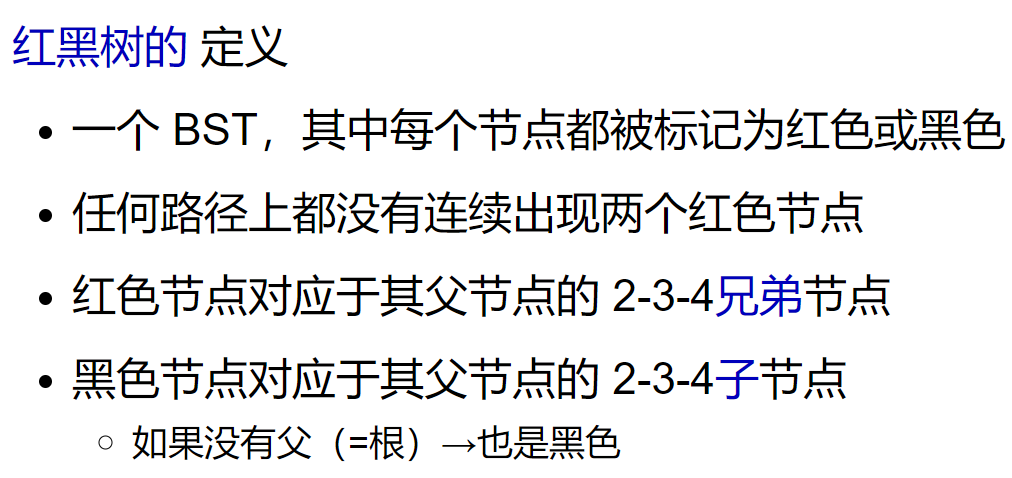
AVL树



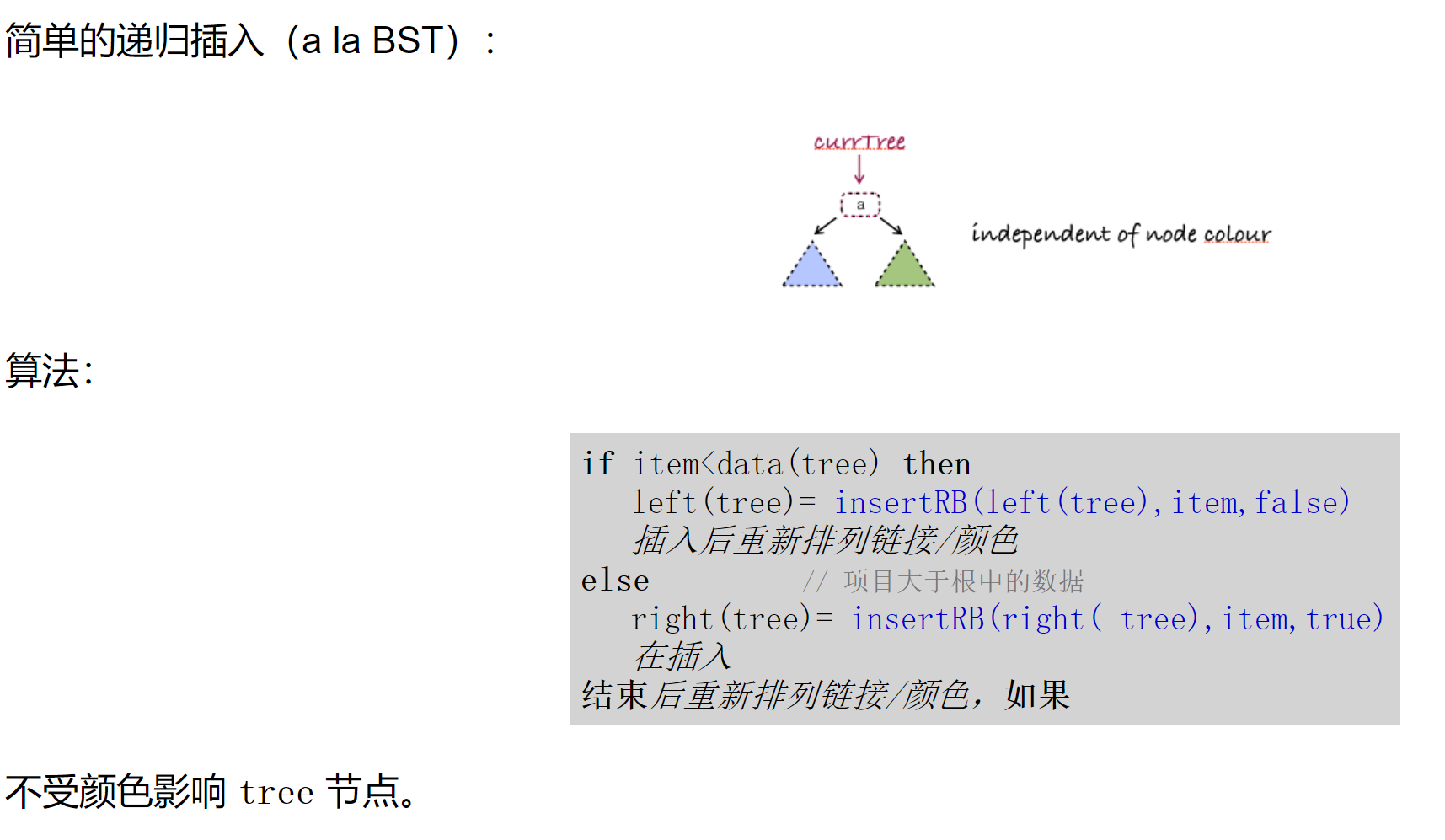
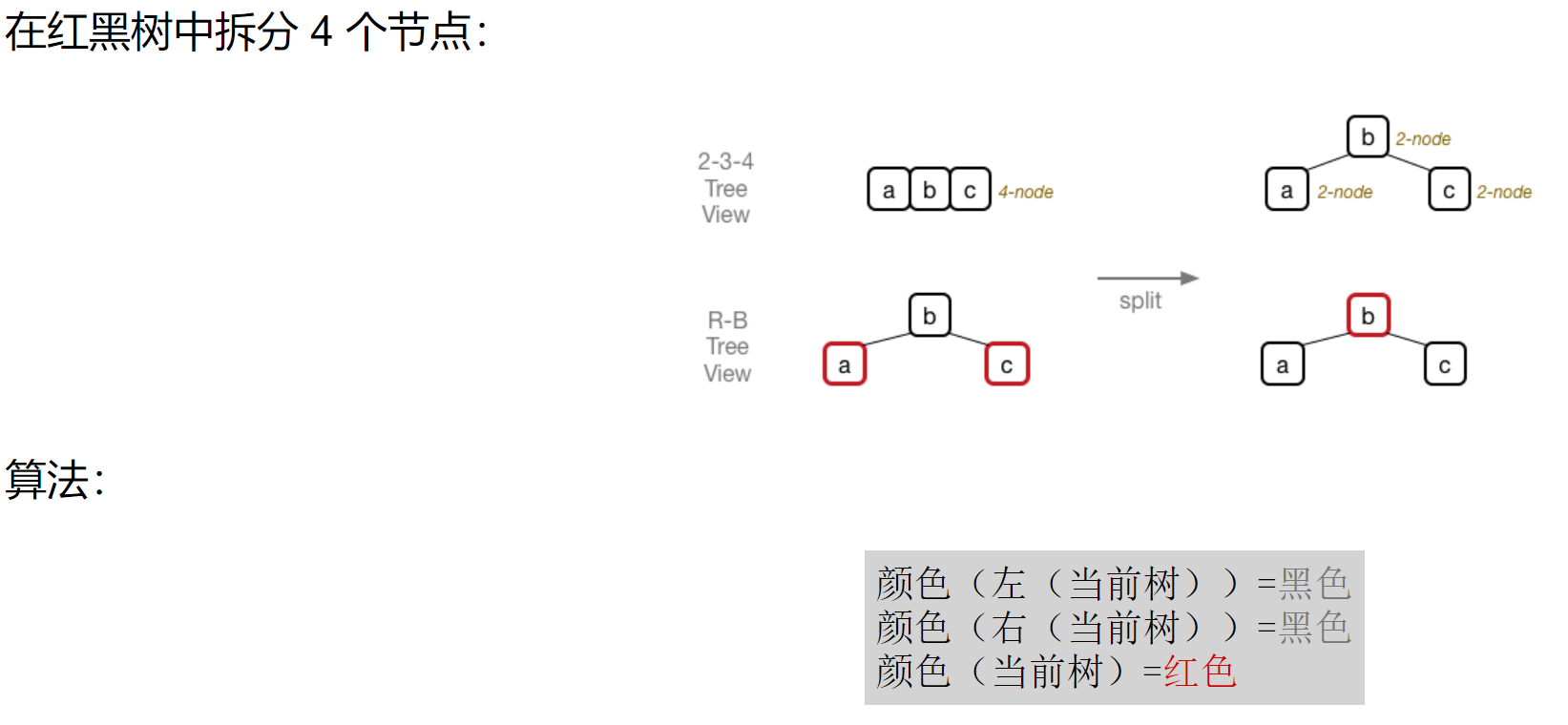
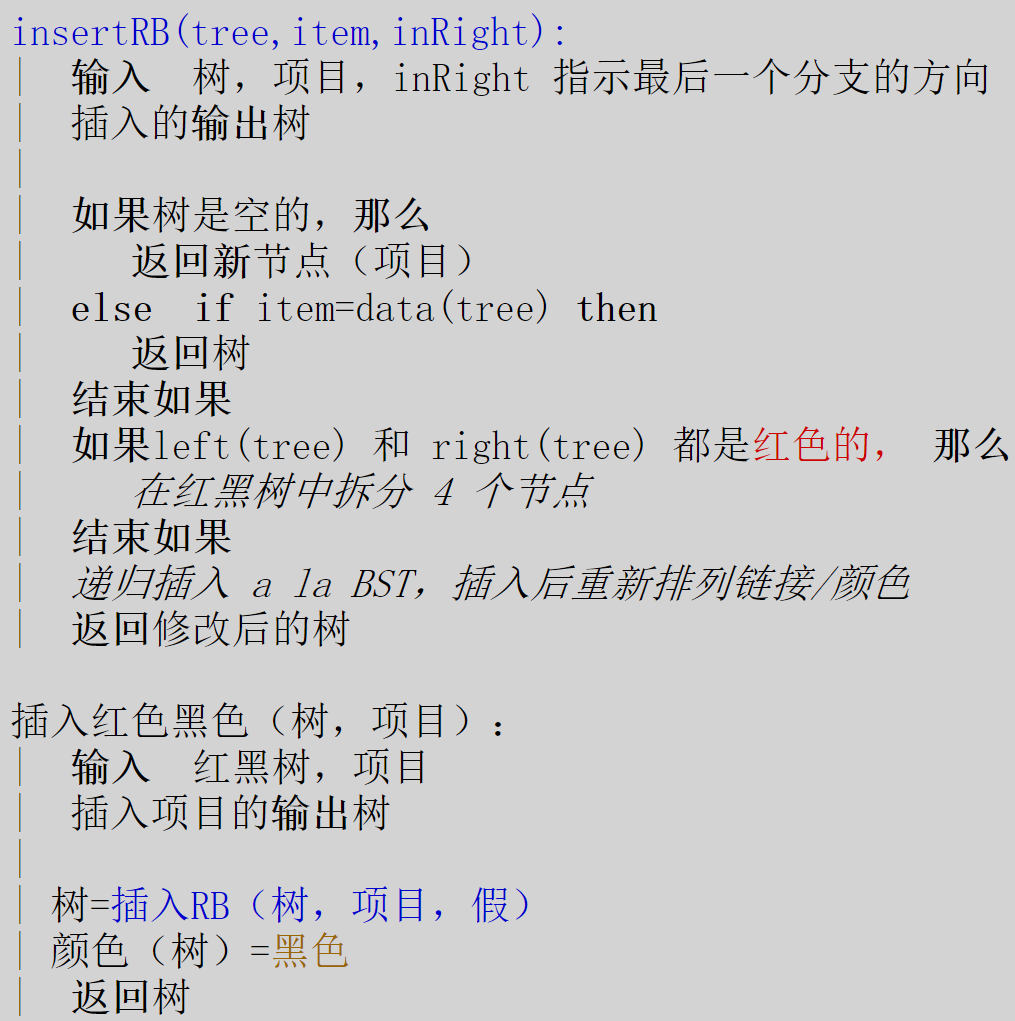
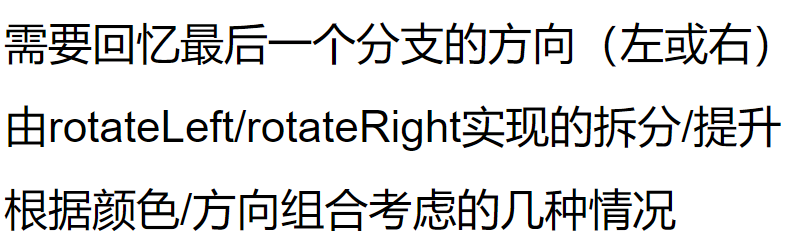
2-3-4 树



红黑树

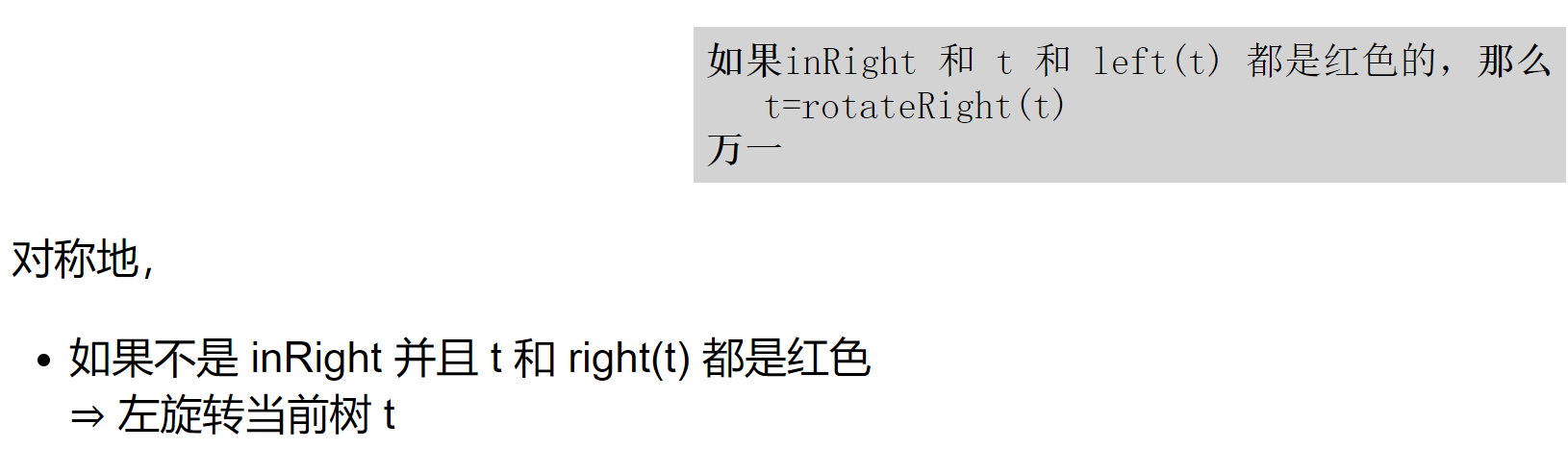
红黑树的搜索是标准的BST搜索

插入

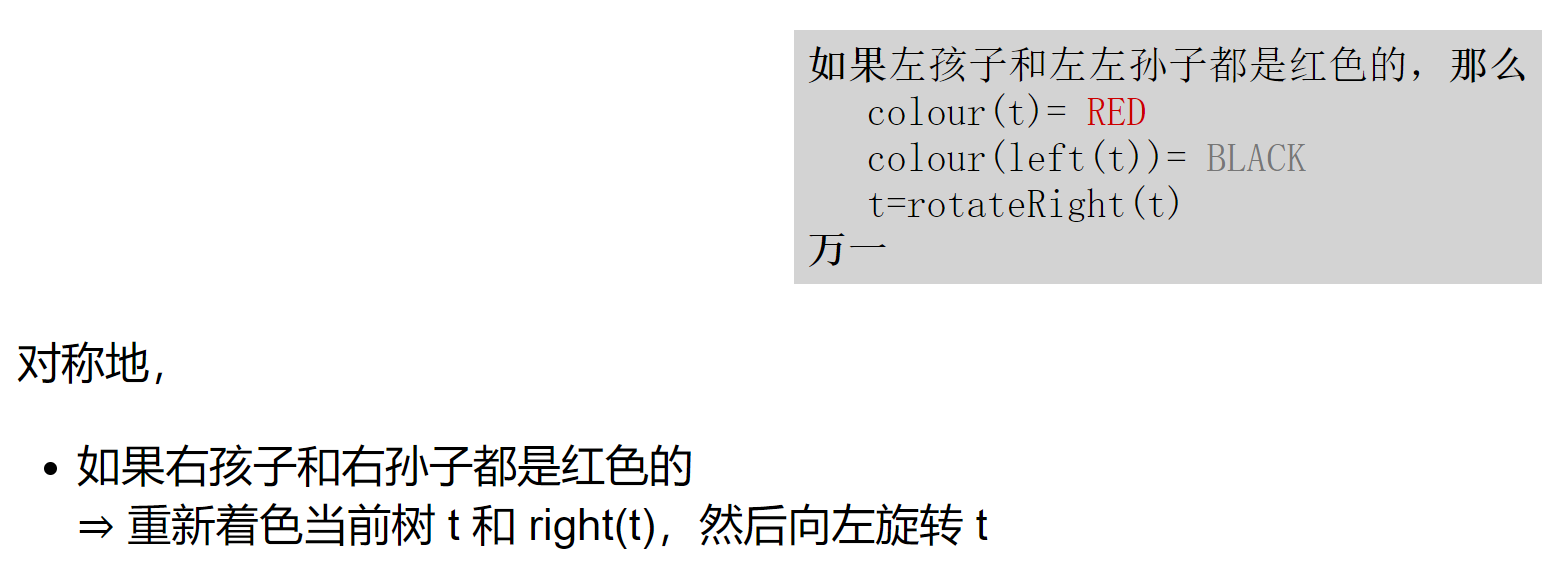


插入后重排：

1. “标准化”连续红色链接的方向

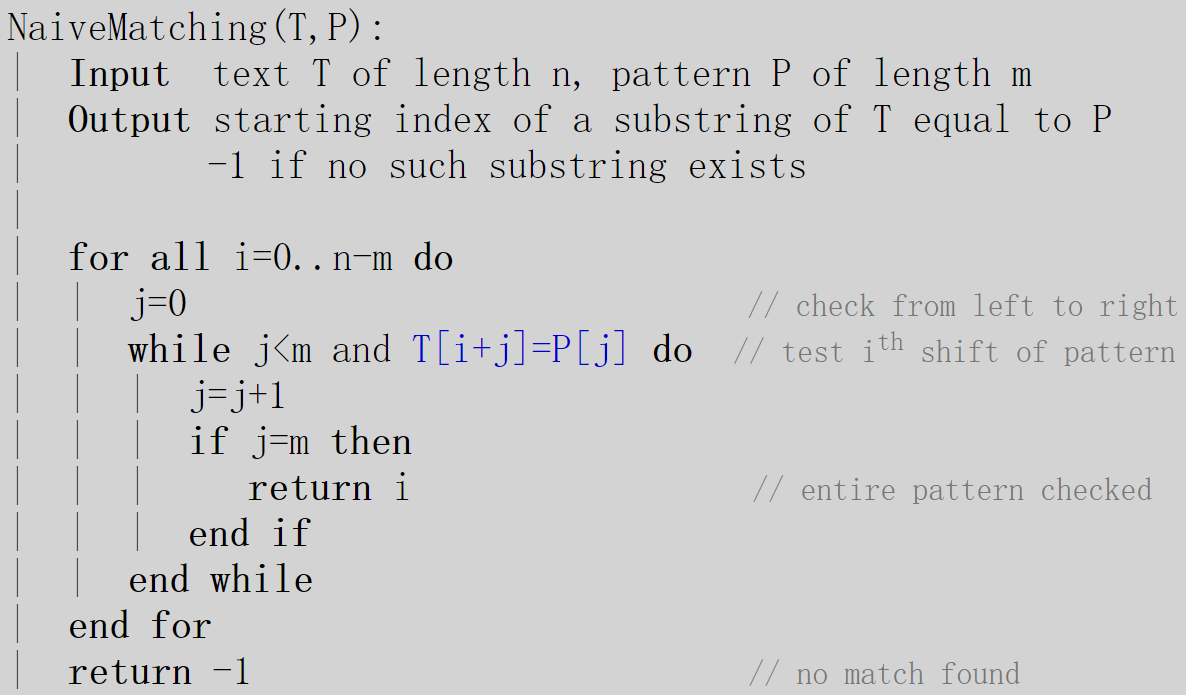


2.两个连续的红色链接 = 新创建的 4 节点

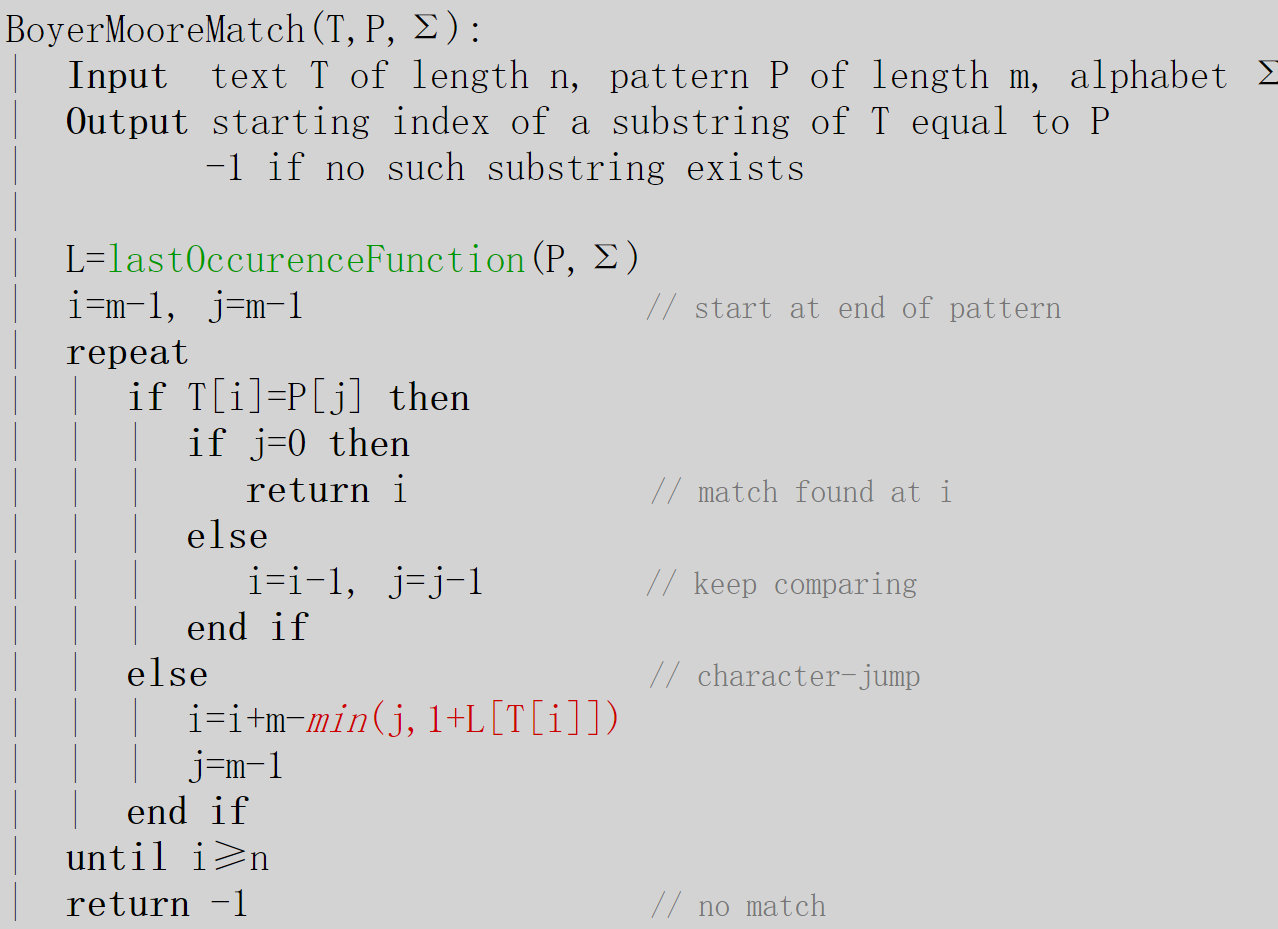
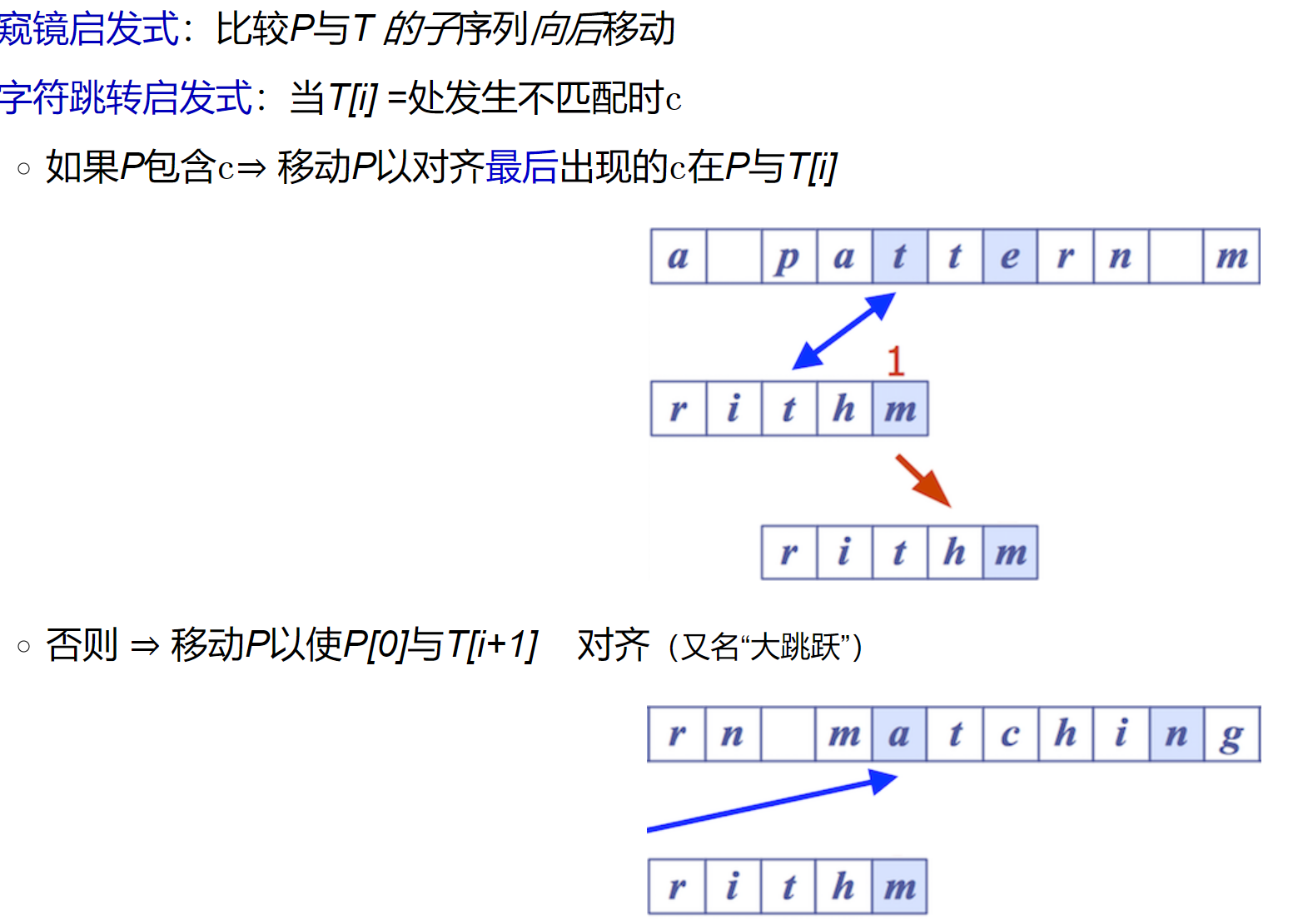


字符串匹配：

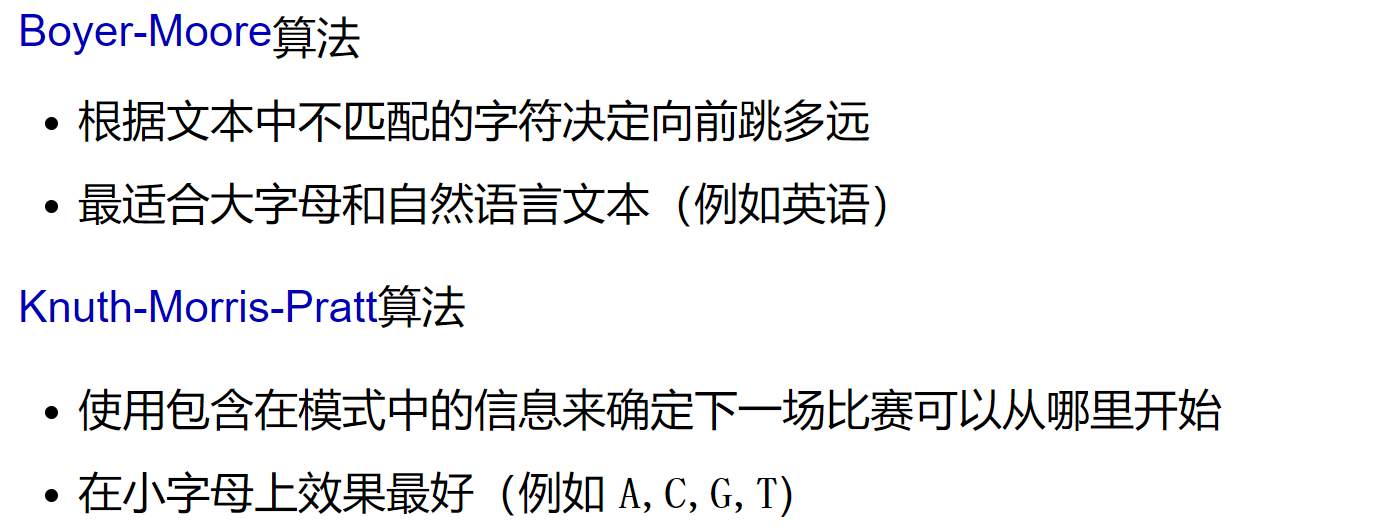
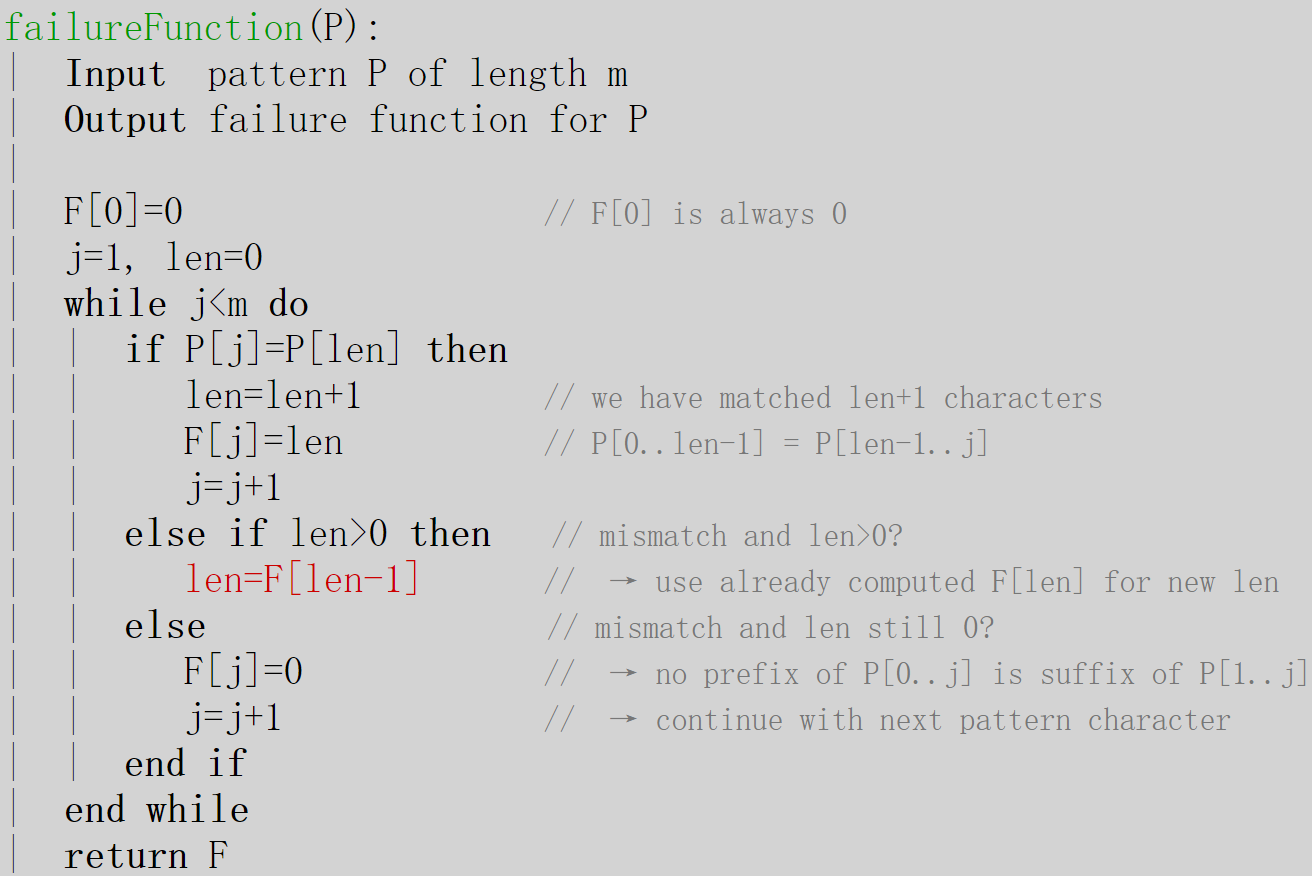
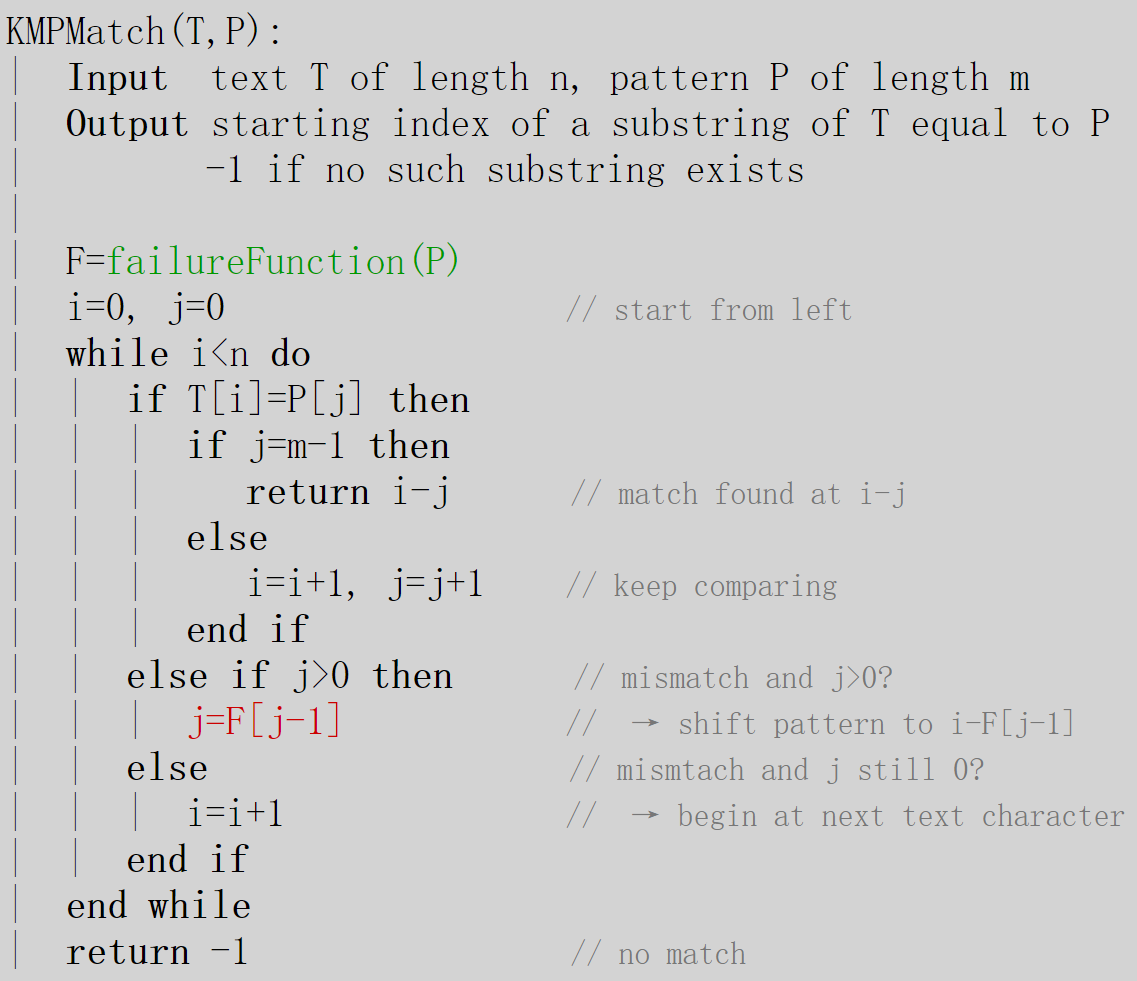
朴素模式匹配



Boyer-Moore 算法



KMP算法



霍夫曼编码

