一，数据结构

1-1线性表list

实现：数组array，链表list，

延申：

队列queue(先进先出),

栈(stack)(先进后出),

堆(二叉堆，优先队列，提供插入元素和删除最小元素，链表和二叉查找树都可以实现，但不是最好，堆实现最好)，堆实际上是满足如下性质的完全二叉树（最小堆 最大堆）。

任意非终端节点的值不大于左节点和右节点。

建立堆的时间复杂度：O(n),

建堆方法1:O(n),全部放好，自底向上；

建堆方法2:O(NlogN),每插入一个元素，变化一次

删除最大元素：时间复杂度O(logN), 然后开始往下过滤.

插入一个元素：O(logN), 插入到最后，然后开始网上滤.

1-n树 tree

二叉树：

二叉树的前序 中序 后续遍历；

完全二叉树(应用:二叉堆)：

二叉查找树：左子树<根节点<右子树

平衡二叉查找树：

AVL树：

左右旋，两次旋转，

每次插入旋转的复杂度：logN

红黑树：

根节点是黑色，null是黑色，红节点的附近只能是黑色，单条路径上的黑色节点数量一样。

保证了：如果黑色高度是n,则单条最短路线是n-1,最长路线是2(n-1).

伸展树：splay tree

2-3 tree:

2-3 trees and 2-3-4 trees are search trees that allow more than one item to be stored in a tree node.

n-n 图 graph

二，常见算法思想

算法：

1. brute force,
2. 递归(自己调用自己recursion)，迭代（循环调用一个方法iteration），
3. 贪心算法：总是认为当前是最好的，

Prim算法: 图的最小生成树

图的djesterla 图的单源点最短路径

Huffman encoding for data compression:

根据概率为所有元素进行构造树编码，概率高的编码少，效率高。

1. 动态规划： dynamic programming 多阶段决策过程的最优化理论。

We want to solve a recurrence relation and the recursion involves overlapping instances.

An optimal solution to an instance can be obtained by combining optimal solutions to the sub-instances.

我们想要解决一个递归关系，而递归涉及到重叠的实例。通过组合子实例的最优解，可以得到实例的最优解。

Examples:

1.Fibonacci, bottom-up;

2. Coin-row problem: S(n)= {s(n-1),s(n-2)+Vn};

有一排硬币，只能选择不相邻的硬币，求最大价值。

3. the Knapsack problem:

f[i][v]=max{ f[i-1][v],f[i-1][v-w[i]]+val[i] }

依次从背包中拿东西，看剩余重量对应的价值。

自顶向下求解最大值，问题：重复计算

方法1：自下向上求出所有的最大值，存在浪费，

方法2：自上向下求出的值进行记录，避免重复计算。

4.动态规划和图的问题：

4.1 Warshall’s Algorithm-- Computing the transitive closure of a directed graph

计算一个图中，经过k步哪些点是连通的。

三次循环，第一层，从1到k,第二三层，矩阵求积。

4.2 Floyd--Finding shortest distances in weighted directed graphs

所有点经过1，2，3…n之后的最短路径

1. 减分转治：

减治：decrease and conquer, 将每个大问题减少为更小的问题，

分治：divide and conquer, 将每个大问题分为两个或以上更小的问题。

转治：transfer and conquer, 将大问题转化为其他问题解决，预先排序。

三．数据结构的常见问题

1.图的常见问题：

图的表示：adjacency list, adjacency table 邻接矩阵 邻接表

Complexity: N+M, NN

图的遍历：深度优先遍历(先序遍历)，广度优先遍历(层序遍历)，复杂度：n,

栈，队列，判断是否已经访问过。

通过遍历判断图是否有环：判断是否有back edge.

图的拓扑排序：

1. 统计所有点的入度，从入度为0开始循环调用，放入队列中。
2. 深度遍历的栈实现，

图的最小生成树：

Prime’s Algorithm

Kruskal’s Algorithm

图的最短路径：

Djisterla算法：单原点到所有点的最短路径。单原点非循环正权重最短路径。

四．排序算法

1.冒泡排序bubble sort，依次交换向后放 --

2.选择排序selection sort，每次比较选择最小的下标放在开头

3.插入排序insection sort，减治算法，减变治排序，

4.希尔排序Shell sort,减治法，减少每次减少的数值，直到1.

5.快速排序quick sort，快速排序的基本思想是：采取分而治之的思想，把大的拆分为小的，每一趟排序，把比选定值小的数字放在它的左边，比它大的值放在右边；重复以上步骤，直到每个区间只有一个数。此时数组已经排序完成。--

6.归并排序merge sort，归并排序的基本思想：分治，一分二，然后比较大小合并到一起。--

7.堆排序 heap sort,找堆最大值N×logN --

特定问题排序：

8.Sorting by counting: 数字很多，但种类很小，hashing思想，数字的种类作为index,计算每个index的数量，进行排序。 复杂度：N

五．空间换时间算法

1. Horspool’s String Search Algorithm：

2. 其他字符串匹配算法：

3. hashing： 通过时间换空间的最佳例子。

1. 图的遍历和回环判断。