以小知识点为主，by 瞿

Note that：const，&，template<class T>，

Sort

插入排序 — 直接 $O(n)\sim O(n^2)$；折半（移动次数不变）；希尔，gap：$n\sim 1$

选择 — 直接 $O(N^2)$；堆 $O(N\log N)$ 空间 $2N$

交换 — 冒泡 $N\sim N^2$；快排（递归）$N\log N\sim N^2$

归并 left mid right $O(N\log N)$ 要重新开外空间

基数 —— 分配至 bucket + 重组 p（多看几次）

To conclude
- 直接插入、直接选择排序适合于非常少量的数据。希尔排序对中等的数据量是一个好选择。冒泡排序适合原来就较有次序的序列。归并排序、快速排序、堆排序都有O（N log N）的性能。

外部查找与排序

主存储器 - 内存，外存储器 —— 存储量大 bit 速度慢

B树

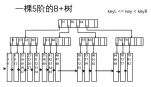
definition：根节点 $[2,m]$ 个子；内节点 子数 $[\lceil m/2\rceil, m]$；节点：$\boxed{s, a_1, \cdots, a_s}$ $\lceil m/2\rceil-1\le s\le m-1$，叶子同层

insert：永远从最底层插入，当超出规定时，分裂

remove：从左/右借一个 or 合并

将一个磁盘块作为一个B树的结点，总数据量：$(M-1)\log_t(M+1)$ 个键映射 + N 个分布的地址

B^+ 树 —— 叶子节点才有data，所有叶子连成一个单链表



「$\frac{L}{2}$」至 L 个数据元

insert：在data中插入，"分裂"

remove：从叶层中借 ⟶ 合并 ⟶ 更新父节点 ⟶ 自下而上

两路归并、多阶段归并

Stack 相关

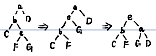
application：递归，括号匹配，前中后缀式（派生式）转换

Set 查找和排序（动态）

二叉搜索树 insert 用 R 往回

remove：分类讨论 removed 节点是否有二个子节点，前驱：左最大 or 右最小

AVL

insert：LL —  ；LR 

remove：以左子树删除为例  依据 D.E 高度进行 RR/RL，高度变化影响上位；D=E 时，高度未变，结束

哈希法/散列表

hash function 散列函数 H：H值域为 $0\sim m-1$，数组称哈希表，实际元素个数/m = 装载系数

H选择：取余，数字，平方取中，折叠

线性：删除加懒找 —— 加入标记

Tree

完全二叉树，高度 规模

遍历相关：

非递归实现：层次遍历 - 队列；前序遍历 — 右入栈 + 左入栈

Significant THU P128

中序遍历 —— 1. 01出栈标记 2. 一路向左 + pop + 转右 3. 回溯 flag（需重 present，不建议）

后序遍历 —— 1. 出栈标记 0, 1, 2

层 / 前+中 / 中+后 ⟶ 唯一地确定二叉树

前：ALBECDWX
中：BLEACWXD
后：BELXWDCA

Huffman code（用长度为 2N 的数组）

树 ⟷ 二叉树 "左子右兄" 树先根遍历 = 二叉树 先序；后根 = 中序

孤树/森林的 先序+后序遍历 可唯一确定

表达式树 等价类（并查集）

Graph

术·概念：有向无环图（DAG）

存储形式：邻接矩阵，邻接表 —— 删除 结点、代码 PPT PB!

DFS & BFS

Application：

Euler path iff degree 为偶数（走后 remove，一直走） 多回顾 P113

拓扑排序（AOV网）

关键路径（AOE）