复习要点：

① 数据结构三元素：逻辑结构、存储结构、数据运算

存储结构：顺序 链式

逻辑结构：集合 线性 树形 图形

② put(r,e)：向量 a[r]=e （捐部）

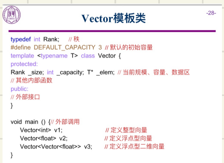
disordered()：return 降序对 4 3 7 4 9 6 ：3

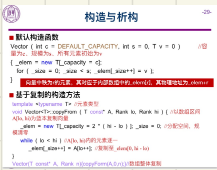
deduplicate()：无序

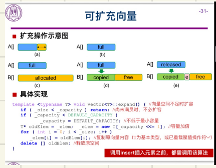
uniquify()：有序

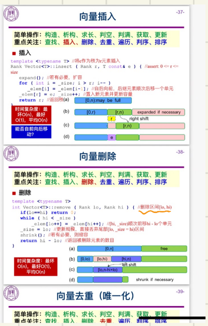
find()：无序 等于e且 rank max 后=>前 顺序查找 失败：lo-1

search()：有序 小于等于e且 rank max









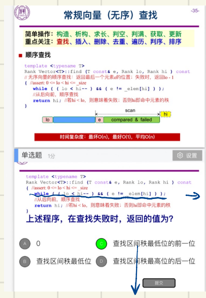
△ _elem[lo++] = _elem[hi++]

△ _size = lo 此时：[lo, hi) 为 [□, _size)

lo已更新

不要的 part

原来



hi-- 等价于 { hi--; return hi+1; }

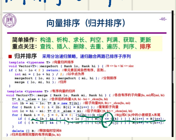
So, be like：hi--；判断 lo 与 hi+1

之后判断 e 和 _elem[hi]

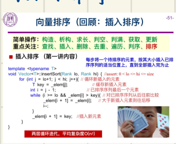
lo == hi 时，hi--，返回 hi+1 (lo)，得 false. now hi = lo-1

失败 lo-1

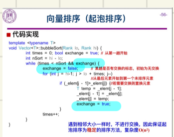
归并排序 O(nlogn)



插入排序 O(n^2)



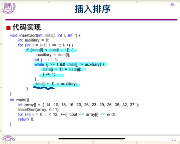
起泡排序 O(n^2)

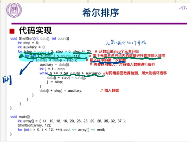


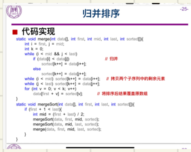
选择排序 O(n^2) 不稳定

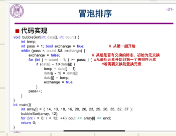
排序

插入改进版

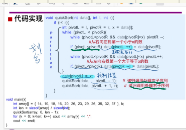


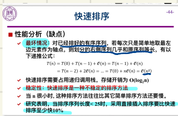


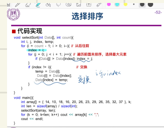


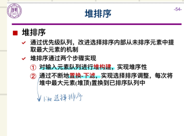


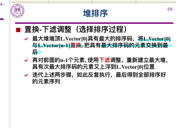
$O(n^{1.3} \sim n^2)$

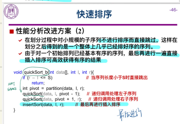


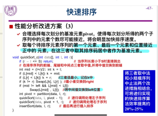




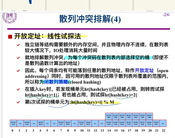


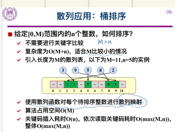


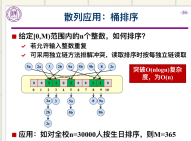


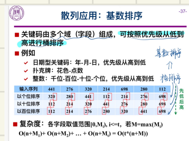


散列







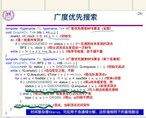


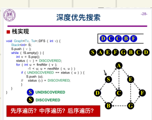
ht[(hash(key)+i)%M]

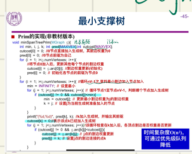
图

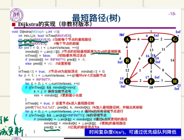
elogn

重点算法



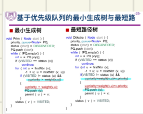


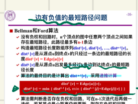




end 是 index

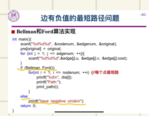
优先级维护距离

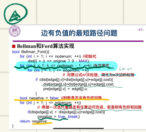




实现

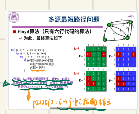
(单源：起点定)

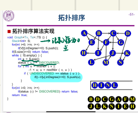


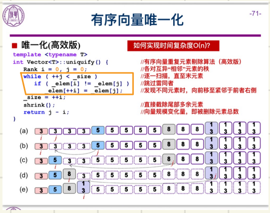


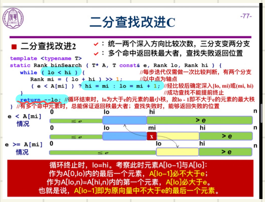
不必单独设 K-value 变量

迭代更新K次，不变即停

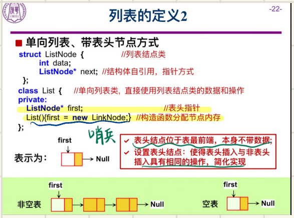


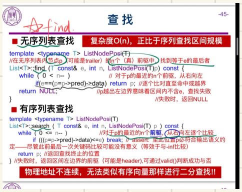


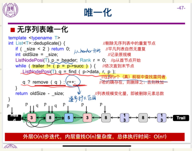


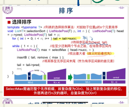


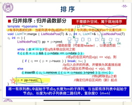
③ 列表 哨兵











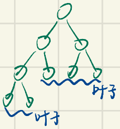
列表上删max，新插max

∴ 向量上 max 和 最后一位 swap

列表的选择：稳定

双向列表：header, tailer 哨兵

④ 完全二叉树：除最后一层，rest 全是满的，且 最后一层尽可能往左靠



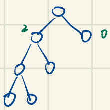
叶子可能不止最后一层有

满二叉树

真二叉树：每个节点孩子为0 or 2

内部结点：除叶子外all点

平衡二叉树：每节点，左右子树高度差 ≤ 1.



$O(h) = O(\log n)$

空树树高 = -1

○ $h=0$

空 $h=-1$

x.

△ 性质：

i. 对二叉树，出度为 i 节点有 n_i 个，则 $n_0 = n_2 + 1$

边总数 $= 0 \cdot n_0 + 1 \cdot n_1 + 2 \cdot n_2$

$= n_0 + n_1 + n_2 - 1$

$\therefore n_2 = n_0 - 1$

ii. 树节点数 = 边总数 + 1

$n_0 = n_2 + 1$

长子兄弟法 旋转45°

☆ 森林到二叉树的转换

a. 每棵树 ⇒ 二叉树

b. all二叉树树根 ⇒ 看成兄弟 ⇒ 长子-兄弟

二叉搜索树
- Normal
  - 条件：中序
  - search
  - insert
  - remove
    - 叶子
    - 非叶子 — 中序直接后继
- 平衡
  - 条件：左右子树 $\Delta h = 1$
  - search
  - insert — 插完 → 高度更新 平衡检查 tallerchild
  - remove — 叶子? + one child? ↗ 3+4 旋转
- B树 — 条件：$(\lceil \frac{m}{2} \rceil, m)$ 分支
  - search
  - insert → 先插后 上溢 check
  - remove → 叶子? 删后 下溢 check

前序固定，可能的中序序列：卡特兰数

$h(n) = \frac{(2n)!}{n!\,(n+1)!}$

n节点二叉树形态数目（只看形状）：卡特兰数

$L_K = N + 1$

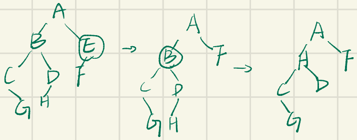
n 各异节点组成二叉搜索树：卡特兰数

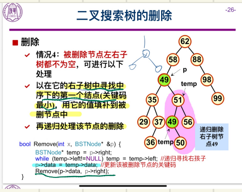
堆
- 构建
  - 蛮力 insert
  - 堆合并 $\frac{n}{2}-1$ 上滤
- insert — 插最后 上滤
- delMax — 置换 + 下滤

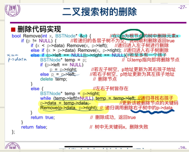
parent(i) $= \frac{i-1}{2}$

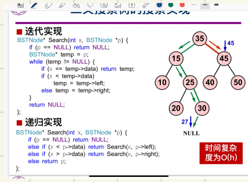
lc(i) = i×2+1

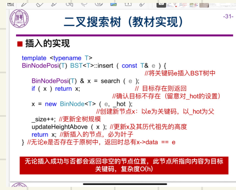
rc(i) = i×2+2

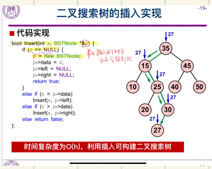






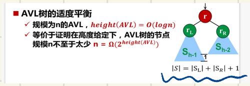




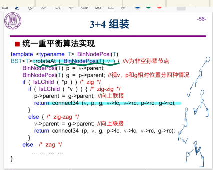


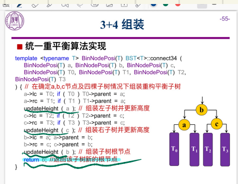
二叉搜索树 等价：中序遍历序列相同

平衡因子：$h_左 - h_右$



$N(h) = 1 + N(h-1) + N(h-2)$



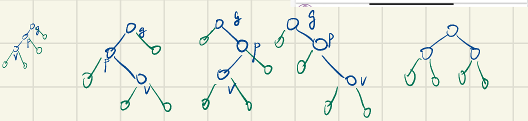


$a<b<c$

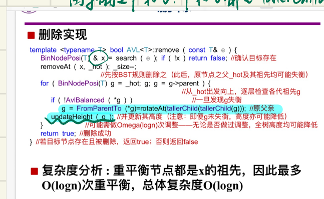
$T_0<T_1<T_2<T_3$

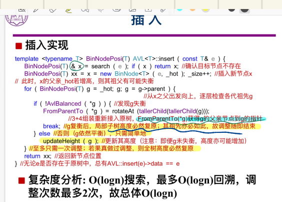
删除：用其中序直接前驱A代替后，删除原本的A

→之后检查父节点 看是否平衡

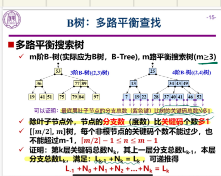


由g确定p和v：p和v都是tallerChild





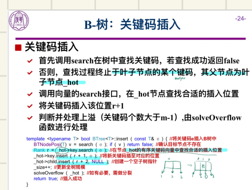
B树 $(\lceil \frac{m}{2} \rceil, m)$树

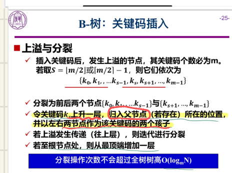


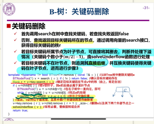
$\log_m(N+1) \le h \le \log_{\lceil m/2 \rceil}\lfloor \frac{N+1}{2} \rfloor + 1$

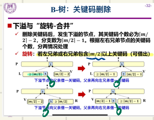
$\log_m(N+1) < h \le \log_{\lceil m/2 \rceil}\lfloor \frac{N+1}{2} \rfloor + 1$

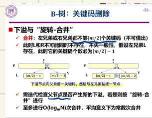
$L_k = N+1$

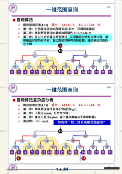


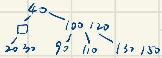




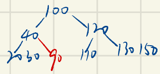




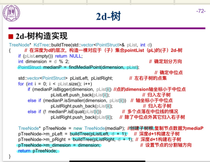


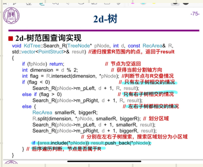


→

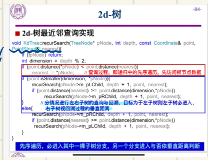


2-d 树

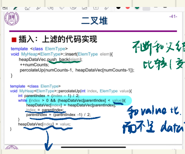




随时更新



堆



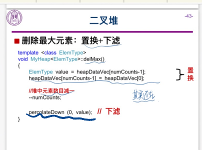
不断和父结点比较(交换)

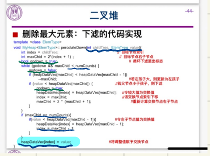
childTree 是可能放value的地方 place.

和value比，而不是data[index]

最后再赋value值

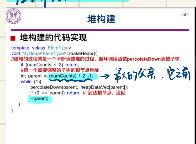
index 记录最后value放哪里



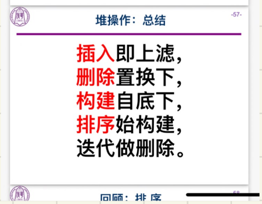


暴力:(逐个插入): O(nlogn)

堆合并法: O(n)



→ 最后的父亲，它之前全是父亲



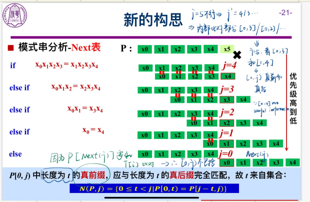
Master Theorem

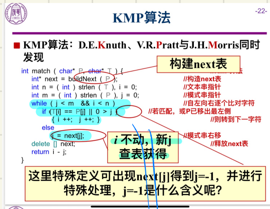
形如 $T(n)=aT(n/b)+O(f(n))$

① 严格大于: $T(n)=O(n^{\log_b a})$

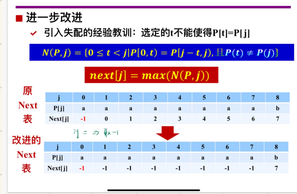
② $f(n)=\Theta(n^{\log_b a}\cdot\log^k n)$, $T(n)=O(n^{\log_b a}\cdot\log^{k+1} n)$ 如: $T(n)=1\times T(n/2)+O(1)=O(\log n)$

③ 严格小于: $T(n)=O(f(n))$





O(N+M)

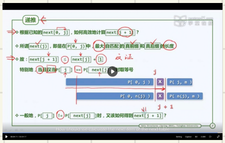


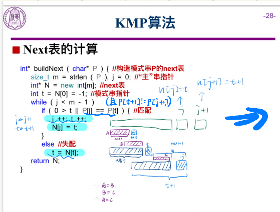
if P[next[j]] = P[j]

⇒ next[j]--, 继续找

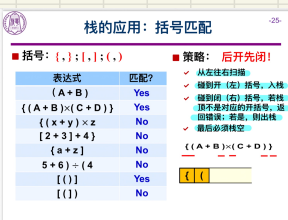
if 找不到, 试试 0

if 不行 ⇒ -1





O(M)



表达式	匹配?
(A + B)	Yes
{ (A + B)×(C + D) }	Yes
{ (x + y) × z	No
[2 + 3] + 4 }	No
{ a + z]	No
5 + 6) ÷ (4	No
[( )]	Yes
[( ])	No