

匹配算法: $\left\{ \begin{array}{l} BF: \text{串逐个往后匹配} \\ KMP: next[i] = \left\{ \begin{array}{l} 0, i=1 \\ 1, \text{其他情况} \end{array} \right. \end{array} \right.$
 例: $\begin{array}{ccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ a & b & a & b & a & a & b & a & b & a & a & a \end{array}$
 数组值 next: $0 \ 1 \ ① \ 2 \ 3 \ 4 \ 2 \ 2 \ 3 \ 4 \ 5 \ 6$ -1, 0, 0, 1, 2, 3, 1, 1, 2, 3, 4, 5
 nextval: $0 \ 1 \ ① \ 0 \ 1 \ 0 \ 4 \ 2 \ 1 \ ② \ 1 \ 0 \ 4$
 例: \boxed{ababab} $k=1,3,5,7,9,11$ $k=4 = next$

树
 二叉树性质: $\left\{ \begin{array}{l} ① \text{结点数} = \text{度数之和} + 1; \text{空指针} = \text{总节点数} + 1 = \text{总度数} + 2 \\ ② h = \lceil \log_2(n+1) \rceil \quad h_{\max} = \lceil \log_m[n(m-1)+1] \rceil \\ ③ h \text{高} \rightarrow \text{多有 } 2^{h-1} \text{ 结点} \end{array} \right.$

完全二叉树: 叶结点数 = $\lfloor \frac{\text{总结点数}}{2} \rfloor$

线索二叉树:

lchild	leaf	data	right	rchild
--------	------	------	-------	--------

leaf $\begin{cases} 0, & \text{指向左孩子} \\ 1, & \text{指向结点前驱(如直接结点前驱)} \end{cases}$

right $\begin{cases} 0, & \text{指向右孩子} \\ 1, & \text{指向后继} \end{cases}$

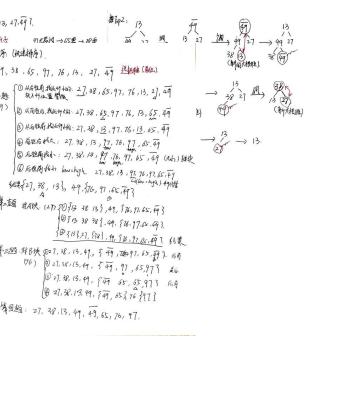
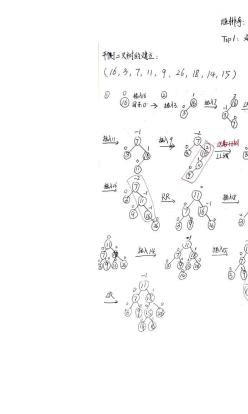
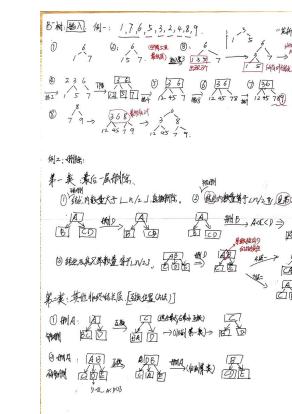
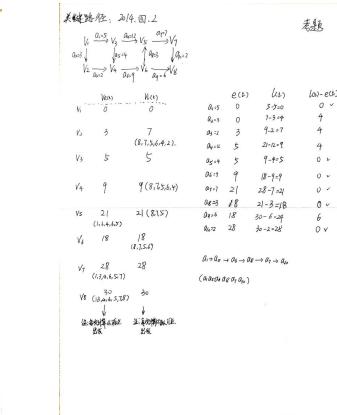
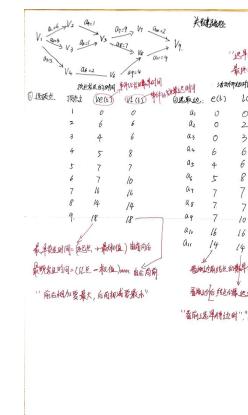
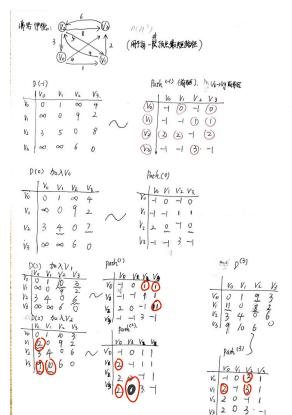
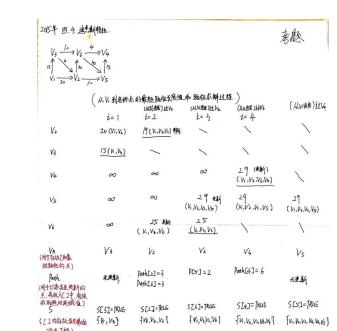
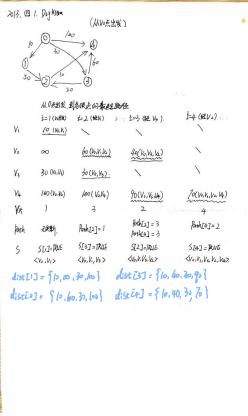
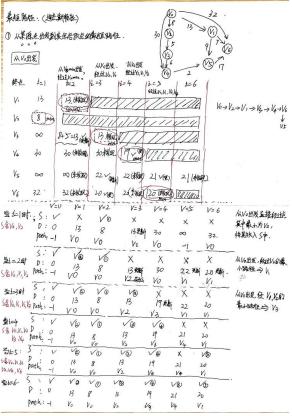
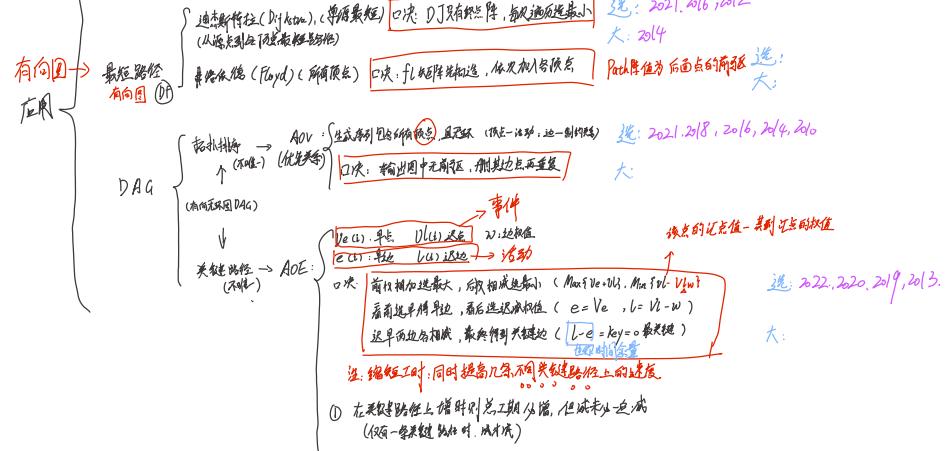
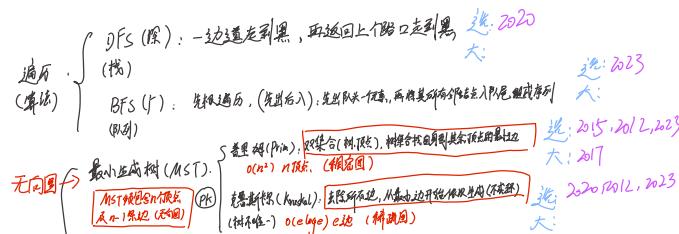
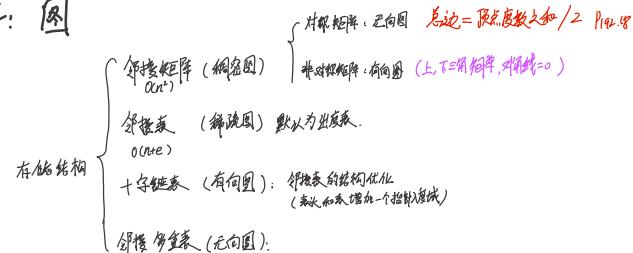
森林
 双亲表示法, 孩子表示法, 链式双亲表示法(一孩子指针及兄弟指针)
 树 \rightarrow 二叉树 (兄弟相连, 偏强)
 森林 \rightarrow 二叉树 (小树变父, 把根连起来)
 二叉 \rightarrow 森林 (右孩子断开, 小二叉变森林)
 树, 森林, 二叉树 不一样, 森林将二叉拆开

哈夫曼树
 选最小两个值合并(小左大右冲)
 $WPL = \text{权值} \times \text{路径长}$
 哈夫曼编码: 左0右1, 哈夫曼树总结点: $2n - 1$



$$\log_3(4+1) \quad \log_5(3^3) \quad \frac{\log_3(3(3+1)+1)}{2} \quad \log_3(3 \times 2 + 1)$$

第6章：图



第7章. 魏

(①顺序存储 (顺序存储 & 链式存储) 有序

线性结构
② 折半查找：mid = $\lfloor \frac{(low + high) / 2}{1} \rfloor$, $h = \lceil \log_2(n+1) \rceil$
（仅顺序存储）有序排列
折半查找排序树，同一算法生成的折半查找排序树是唯一的（即始终向上或向下取整）

二叉排序树 { 左 < 根，右 > 根：利用中序遍历判定是否为二叉排序树
 割除：① 杠削：结点左（右）子树空，用原左（右）子树补。
 ② 左右都不空，则在右子树找中序第一个结点 *

①左右子树高度差 ≤ 1 (平衡因子) $h = \log_2 n - \lfloor \log_2 n \rfloor$

平衡三叉树: ② 检查: 最少结点的平衡因子是: $N_h = N_{h-1} + N_{h+2} - 1$ ($N_1=1, N_2=2, N_3=4, N_4=7, N_5=12$)
 $N_h = \text{树的最大高度} - \text{带叶结点数}$ (2012)

③平衡调整: LL, RR 中间度根, LR, RL 呼度根 (同中异底)

B树：性质 {

- ① 阵阶 m 树， $m-1$ 个关键数 同理， n 个关键数至少有 $n+1$ 个结点。
- ② (根和叶除外) 至少有 $\lceil m/2 \rceil$ 根， $\lceil m/2 \rceil - 1$ 关键数 (至少有 2 子树，最少有一关键数)
- ③ 叶子都在同一层

插入：
 $\left\{ \begin{array}{l} ①$ 性质 1,3
 $②$ 如 3 个 B 树出现 3 个关键字， $\boxed{135}$ $\boxed{1}$ 先拆后升。 $\begin{matrix} 3 \\ 5 \end{matrix}$ $\begin{matrix} 6 \\ 7 \end{matrix}$ 性质
 删除：
 $\left\{ \begin{array}{l} \text{最后一层} \text{ 拆 } \\ 1. \text{ 关键字} \rightarrow [m/1] \Rightarrow \text{直删} \\ 2. \text{ 关键字} \rightarrow [m/2] \text{ 或 } [m/3] \rightarrow \begin{matrix} A \\ CD \end{matrix} \rightarrow \begin{matrix} C \\ D \end{matrix} \text{ 合并操作} \\ 3. \text{ 关键字} \rightarrow [m/2] \text{ 或 } [m/3] \rightarrow \begin{matrix} AB \\ C \\ CD \end{matrix} \rightarrow \begin{matrix} B \\ CA \\ C \end{matrix} \text{ or } \begin{matrix} A \\ BE \end{matrix} \\ \text{删除最终点：根叶互换} \quad \boxed{[m/2-1]} \rightarrow \begin{matrix} A \\ A \end{matrix} \end{array} \right.$

B + 树 { ① m 阶 m 个关键字 (插入 and 删除)
② 支持顺序查找 (B 仅支持多路查找)

红墨树 } 性质：①根长叶一黑，②不育在顶部红，③任意结点到叶子路径，黑都相同
 ④根→叶路径：最长 \leq 2倍最短路径 ⑤ $h \leq 2\log_2(n+1)$ 为内部结点数
 ⑥插入都为红色

〔 故列函数：除留余数 〕

冲突处理：① 开放地址：利用增量解决 a
② 扩链法：假连表，容易表示冲突

$$d = \frac{\text{关数}}{\text{散列表长}} \quad (d \text{ 为装载因子}, d \uparrow \text{冲突} \uparrow)$$

$$ASL = \frac{\text{查找元素失败总次数}}{\text{失败数}}$$

4. 用树状图表示抛掷一枚质地均匀的骰子两次的所有可能结果数。并求出出现两个奇数的概率。

(1) 列表法

	1	2	3	4	5	6
1	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
2	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
3	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
4	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
5	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
6	(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)

(2) 树状图

可能出现的结果数为 $6 \times 6 = 36$ 种。

出现两个奇数的结果数为 $3 \times 3 = 9$ 种。

所以出现两个奇数的概率是 $\frac{9}{36} = \frac{1}{4}$ 。

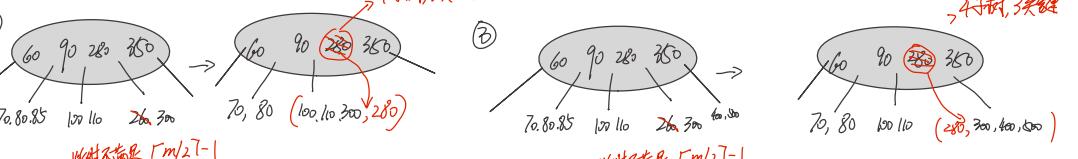
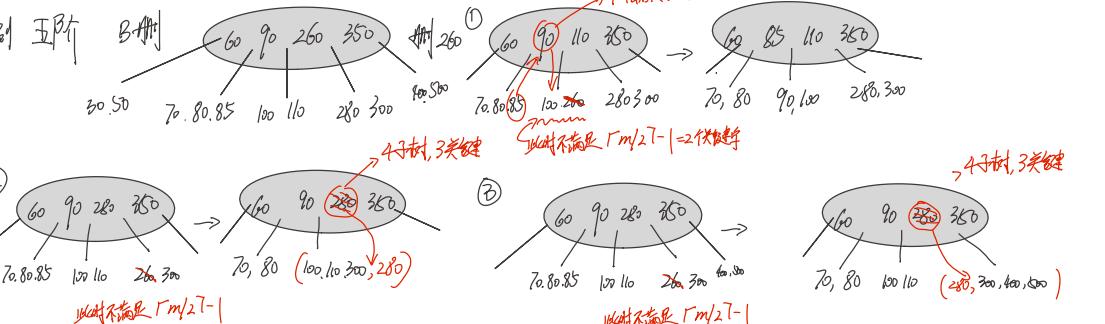
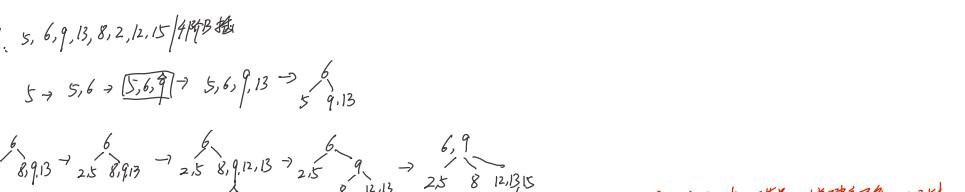
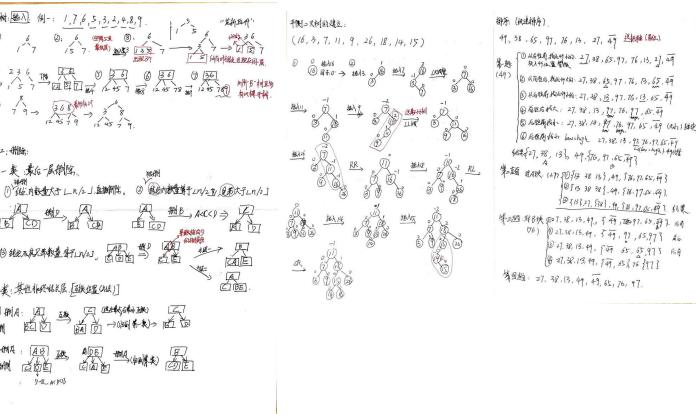
ASALMEN: $= \frac{(x+y)(x+y-1)}{2} + \frac{(x-y)(x-y+1)}{2} = \frac{x^2+y^2}{2} + \frac{y^2-x^2}{2} = \frac{2y^2}{2} = y^2$

ASALMEN: $= (1+2+3+4+5+6)^2 - (1^2+2^2+3^2+4^2+5^2+6^2)$

11

0奇找1次失败
1-2奇找2次失败
2-3奇找3次失败
3-4-5-6-7-8奇找6次失败
以此类推
10奇找12次失败

$$\text{ASLSecc} = \frac{1+1+1+2 + 1+2 + 1+2}{8} = \frac{11}{8} = 1.375$$



第8章 排序

插入排序

- ① 直接插入 (扑克牌：从无序块中选一元素，复制后将其放到每个元素后面后检查是否正确，将元素插入)
- ② 折半插入 (类似于直接插入，区别在于利用折半查找合适位置插入)
- ③ 希尔排序 (折半间隔性，间隔性从大到小，且为质数 7, 5, 3, 1)

本质是直接

交换排序

- ① 冒泡：从头到底对每个元素进行两两排序。(12 33 33 33 ...)
- ② 快排：枢组 (pivot)，选定枢组，后用用依次继续填补

$n \log n$

若枢组第一趟 放最左端，则第二趟仅确定一个枢组值是 $1+1=2$
若枢组第一趟在中间，则第二趟为 2 个枢组值，是 $1+2=3$

选择排序

- ① 简单选择：从序列中选最大 (小) 元素与有序列的下一个位置交换
- ② 堆排序：① 按原队列遍历构造完全二叉树：根 > 左孩子 (标准)
② 从 n/2 开始调整，若为大根堆，则左孩子小于根，右孩子大于根 (根堆堆积)
③ 非堆排序：输出堆顶，并用最后一个元素替之，并将新的堆顶逐层下沉

Step: (1) 建立 (2) 调整为堆 (3) 将堆顶重新调整为堆 (归并删除根) 步骤 3

① 比较次数：根的两个孩子的比较基准，再用最小与根比，共两次比较 ② BC 比较 B > C
③ 则 AC 比较 A > B
④ 且从根堆的元素需要逐步下降
⑤ 插入空的根堆是边插边调整。

二路归并：两两比较后成新组，面对新组两两比较。(12 34 56 78 ...) 分别泡两两不同 (归并 n)

基数排序：分配 + 收集 (例：个位、十位、百位、千位依次排序，准备 10 个桶)
(桶排序)
耗时: $O(k * (n/m))$ 大于快排，m 取值基数依赖
空间: $O(m * n)$ (桶数是 $0 \sim 9$)

① n 较小:
 ⚡ 直接：移动次数少 (信息量少)
 ⚡ 简单：移动次数少，但比较次数多 (信息量大)
 ⑥ 信息量少与比较次数少，信息量多与比较次数多
 ⑦ 比较次数少，但比较次数多 (信息量大)
 完美：选择 (选择次数少)

② n 较大:
 ⚡ 快排 (最优化) 空间大于堆，不稳定 空间: $\log n$
 ⚡ 堆排序 不稳定，不会出现快排的最坏情况 $O(n^2)$
 ⚡ 归并 空间 n

③ n 很大: 基排

④ 关键基本有序，直接冒泡泡

外部排序: ① 提高效率通常减少读写次数: (1) 增大归并路数 (2) 减少片段个数

② 购物车增大归并路数，置换一趟减少归并个数

③ 多路平衡归并 一致性归并树 (本质上是 K 个哈夫曼树)

④ $N_k = k - 1 - (n_0 - 1) \% (k - 1)$ 其中 k 为 K 路哈夫曼树 K 路归并
 n_0 为初始归并段个数, N_k 为虚段个数

第8章 排序

插入排序

- ① 直接插入（扑克牌：从无序中选一位置，复制后将有序部分依次插入到正确位置，特指插入）
- ② 折半插入（类似于直接插入，区别在于利用折半查找合适位置插入）
- ③ 希尔排序（折半间隔性，间隔性从大到小，且为质数 7, 5, 3, 1）

本质是直接插入

交换排序

- ① 冒泡：从头到底对每个元素进行两两相邻。（12 33 33 ...）
- ② 快排：数组（pivot），选定数组，后用冒泡法确定一个基准值 i+1~2
 ↗ 第一趟：从左到右，按左往右，则第二趟仅确定一个基准值 i+1~2
 ↗ 第二趟：基准值在中间，则第二趟为两个基准值，类 H2=3

选择排序

- ① 简单选择：从序列中选最大（小）元素与有序列的下一个位置交换
 - ② 堆排序：根据遍历构造完全二叉树：根 > 左孩子（根堆）
 ↗ 第一趟：从 i/2 开始调整，若为根堆，则石碑子按一个顺序依次交换，往上调整至根堆上（根堆化）
 ↗ 第二趟：将新堆顶替换为最后一个元素替之，并将新的堆顶逐层下沉
 - ③ 堆排序：输出堆顶，并用最后一个元素替之，并将新的堆顶逐层下沉
- Step: (1) 建立 (2) 调整堆 (3) 将输出重新调整为堆 (归并删除根) 三步
- ④ 比较数据：根的两个儿子比较进堆，再用最小与根比，共两次比较 BC 比较小 BC 大
 ↗ ⑤ 则 A BC 比较小 BC 大
- ⑤ 且从根拿的元素需要重新调整
 ⑥ 插入空的根堆先边插边调整。

二路归并

两两比较后成新组，再两新组两两比较。（12 34 56 78 ...）分层泡两不同（bridge）

基数排序：分配和收集（倒个位，十位，百位，千位依次排序，准备 10 位桶）

（桶排序）

时间 O(K * n/m) 大数快排，m 为基数个数

MD 高效优先 ↗ (...百, 十, 个)

空间: O(m * n)

（插入排序 0 ~ 9）

① n 较小

- 直接：操作数据少（信息量小）
- 筛选：操作数据多，但比较数据少（信息量大）

- ⑥ 无比较数据与初始数据无关，基数 = 此数据数
- ⑦ 此数据数 无关：选择（选择困难）

② n 较大：

- 快排（最优）空间大于堆，耗费空间: $\frac{1}{2}n^2$
- 堆排：空间不足，不会出现快排的基本情况 $O(n^2)$

③ n 很大：基排

④ 文件基本有序：直接或冒泡

性能排序

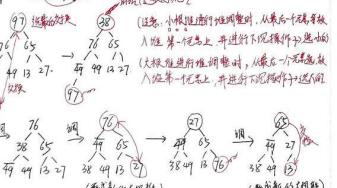
Top 1: 38, 65, 97, 76, 13, 27, 49, 1.



Top 2: 13, 27, 38, 49, 65, 76, 97, 100.



Top 2: 选择堆排序：



1. 设待排序的关键字序列为 $(12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18)$ ，试分别写出使用以下排序方法，每趟排序结束后关键字序列的状态。

- ① 直接插入排序 16 - 16*
- ② 折半插入排序
- ③ 希尔排序（增量选取 3, 5 和 1）
- ④ 冒泡排序
- ⑤ 快速排序
- ⑥ 简单选择排序
- ⑦ 堆排序

⑧ 二路归并排序 (类似插入)

- 第1趟: (2, 12, 16, 30, 28, 10, 16^*, 20, 6, 18)
- 第2趟: (2, 12, 16, 30, 28, 10, 16^*, 20, 6, 18)
- 第3趟: (2, 12, 16, 30, 28, 10, 16^*, 20, 6, 18)
- 第4趟: (2, 12, 16, 30, 28, 10, 16^*, 20, 6, 18)

⑨ 简单选择排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

- 第1趟: (2, 12, 16, 30, 28, 10, 16^*, 20, 6, 18)
- 第2趟: (2, 6, 16, 30, 28, 10, 16^*, 20, 12, 18)
- 第3趟: (2, 6, 10, 30, 28, 16, 16^*, 20, 12, 18)
- 第4趟: (2, 6, 10, 12, 28, 16, 16^*, 20, 12, 18)
- 第5趟: (2, 6, 10, 12, 16, 28, 16, 16^*, 20, 12, 18)
- 第6趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第7趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第8趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)

- 第9趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第10趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第11趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第12趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第13趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第14趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第15趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第16趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第17趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第18趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)

- 第19趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第20趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第21趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第22趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第23趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第24趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第25趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第26趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第27趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第28趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第29趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第30趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)

- 第31趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第32趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第33趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第34趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第35趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第36趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第37趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第38趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第39趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第40趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第41趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第42趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第43趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第44趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第45趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第46趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第47趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第48趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第49趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第50趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第51趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第52趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第53趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第54趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第55趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第56趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第57趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第58趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第59趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第60趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第61趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第62趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第63趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第64趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第65趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第66趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第67趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第68趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第69趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第70趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第71趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第72趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第73趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第74趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第75趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第76趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第77趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第78趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第79趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第80趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第81趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第82趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第83趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第84趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第85趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第86趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第87趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第88趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第89趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第90趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第91趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第92趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第93趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第94趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第95趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第96趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第97趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第98趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第99趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)
- 第100趟: (2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)

- ⑤ 快速排序 (12, 6, 2, 10, 28, 30, 16^*, 20, 16, 18).
 ↗ 第一趟：(12, 6, 2, 10, 28, 30, 16^*, 20, 16, 18)
 ↗ 第二趟：(16, 2, 10, 12, 28, 30, 16^*, 20, 16, 18)
 ↗ 第三趟：(2, 6, 10, 12, 16, 28, 30, 16^*, 20, 16, 18)
 ↗ 第四趟：(2, 6, 10, 12, 16, 16^*, 20, 28, 30, 18)

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑥ 简单选择排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑦ 二路归并排序 (类似插入)

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑧ 二路归并排序 (类似插入)

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑨ 堆排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑩ 希尔排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑪ 快速排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

- ⑫ 冒泡排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

18 > 12

6 < 12

2 < 12

16 > 12

10 > 12

30 > 12

28 > 12

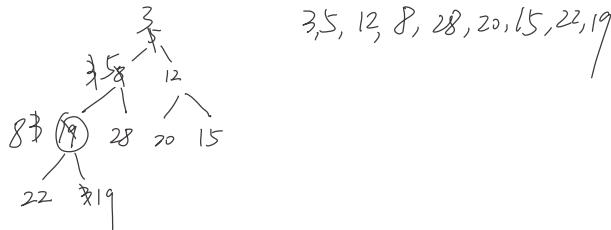
- ⑬ 希尔排序 (12, 2, 16, 30, 28, 10, 16^*, 20, 6, 18).

18 > 12

6 < 12

2 < 12

16 > 12



3, 5, 12, 8, 28, 20, 15, 22, 19

0 1 2 3 4 5 6 7 8 9
 110 911 122 114 007 119
 120

110, 120, 911, 122, 114, 007, 119

0 1 2 3 4 5 6 7 8 9
 151 372 93 485 916 327 9
 31 892 43 146
236

