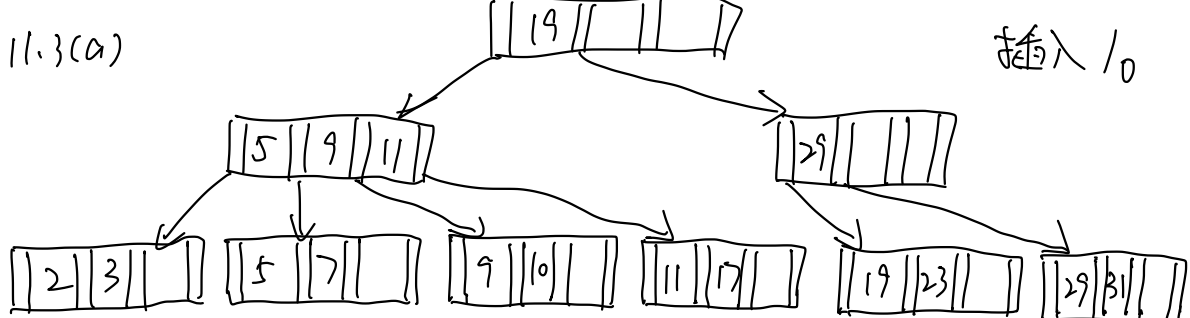


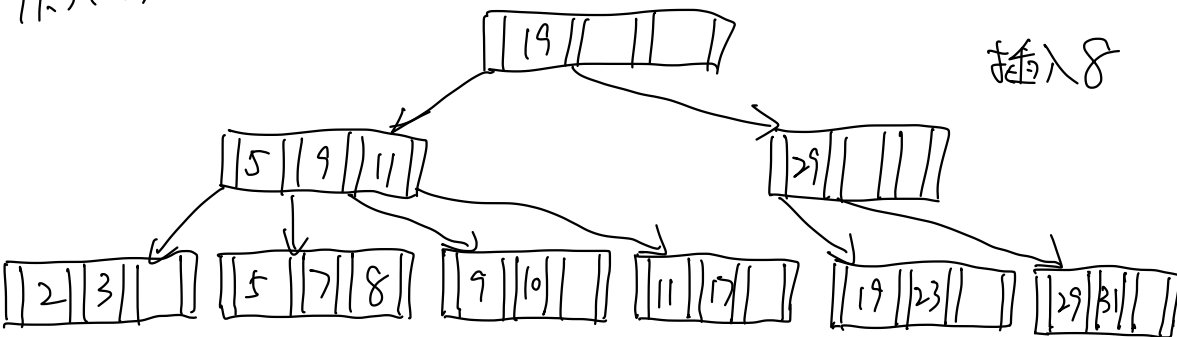
11.3(a)

插入 10



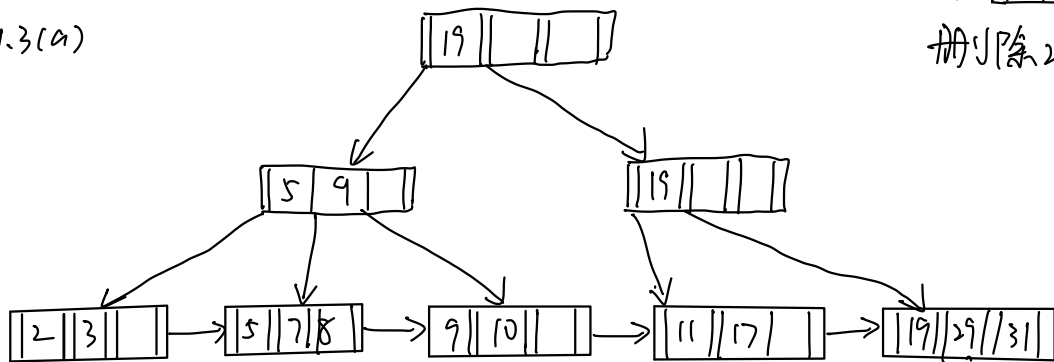
11.3(a)

插入 8



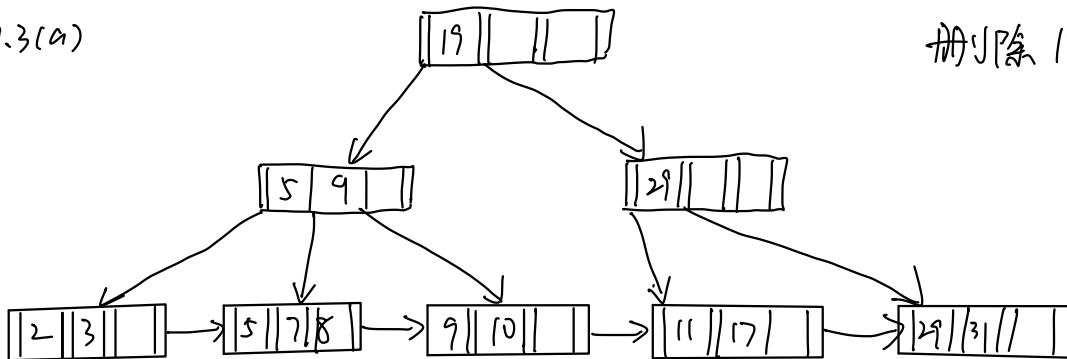
11.3(a)

删除 23



11.3(a)

删除 19



11.3(c)



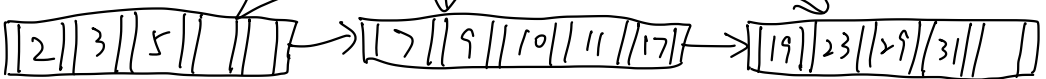
插入 9



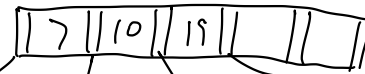
11.3(c)



插入 10



11.3(c)



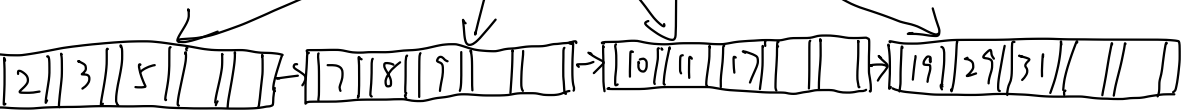
插入 8



11.3(c)



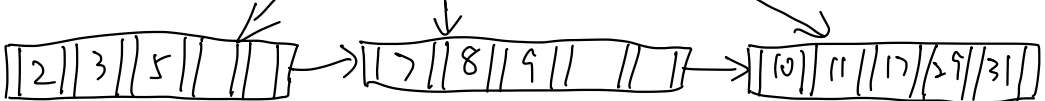
删除 23



11.3(c)

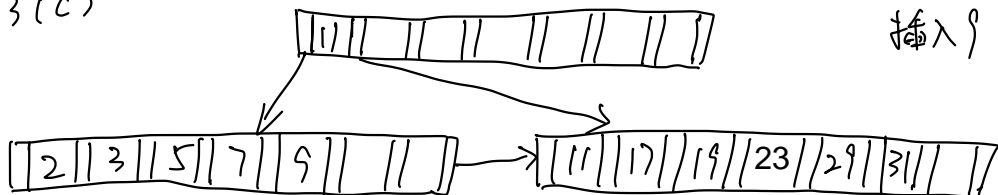


删除 19



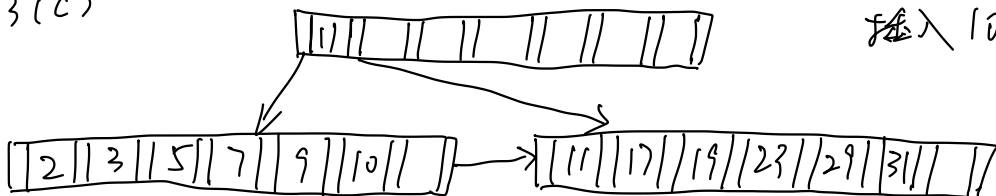
11.3(c)

插入 9



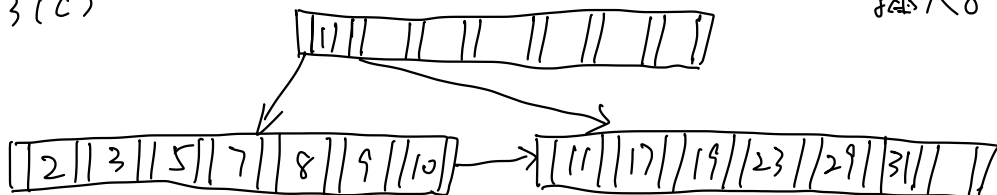
11.3(c)

插入 10



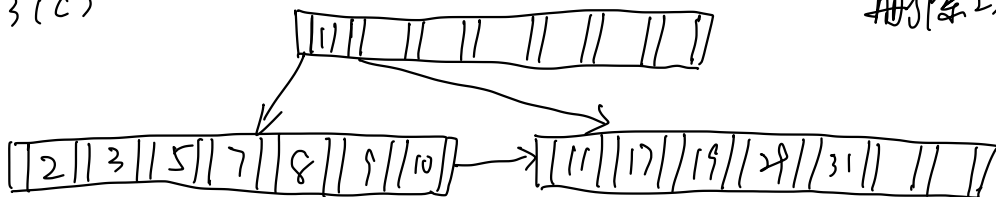
11.3(c)

插入 8



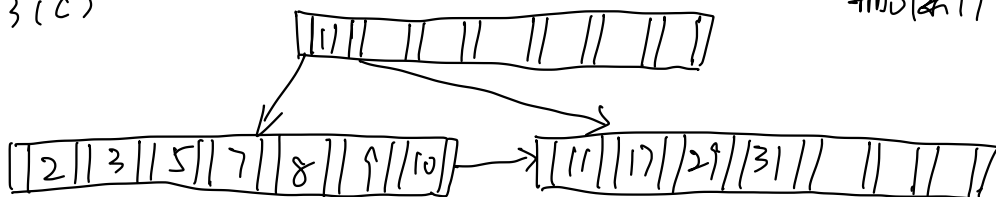
11.3(c)

删除 11



11.3(c)

删除 19



第十一章作业

11.19

11.19 解释闭地址和开地址的区别。讨论这两种技术在数据库应用中的相对优点。

开地址：可能会放置具有相同散列函数值的键在不同的桶里。

优点：很难删除，确定一个桶之前可能必须检查所有的桶的键值已被删除。对于一个小的、静态的数据查找集可能更有效。

闭地址：将键放在同一个桶。不同的桶可以是不同的大小，尽管实现可能是通过将固定大小链接在一起，使用溢出链的存储桶。

优点：只有一个桶需要检查通过哈希键值获得的地址。删除在数据库中更常见，闭地址更适合数据库。

11.20

11.20 在散列文件组织中导致桶溢出的原因是什么？如何减少桶溢出的发生？

桶溢出原因：

- 我们对关系将拥有的记录数量的估计值太低，因此分配的桶数不够。
- 记录到桶的分布偏斜。可能因为有许多记录具有相同的搜索键值，或者因为选择的哈希函数没有理想的均匀性和随机性特性。

减少桶溢出的发生，我们可以：

- 更仔细地选择哈希函数，并做出更好的估计的关系大小。
- 如果关系的估计大小是 n 并且每个块的记录是 f ，分配 $(n/f) \cdot (1+d)$ 桶而不是 (n/f) 桶。 d 是一个模糊因子，通常在 0.2 左右。浪费了一些空间：存储桶中大约 20% 的空间将是空的。但好处是处理了一些偏斜，溢出概率减少。