SBT使用手册

1. Insert插入不会处理相同的元素，相同的元素被相邻的放置，Insert优先将相同的元素插入右子树，这样排名就会是第一次的并列排名（第一次出现的排名），如果需要求当前最后一次出现的排名，可以改造SBT，将Insert中的>=改为>，插入会优先放入左子树；
2. Del删除操作必定会删除一个元素，只要树中尚有元素，会删除搜索到的最后一个元素，一般为前驱，如果没有前驱，会删除后继；
3. 当需要删除所有给定的值k时，可以采用以下解决方案：
4. while未删错，不断删除k
5. while可以find k，不断删除k
6. 当需要删除保留唯一个k，可以修改3
7. 如果树中不允许有相同的元素，可以采用如下解决方案：
8. 插入时，注意之前find
9. 配合set使用插入
10. Select求已知排名的元素，默认会将相同元素考虑，如果需要忽略相同元素，求以不同元素大小计的已知排名求数值，可以采用以下解决方案：
11. 重新建树
12. 为节点追加mul域，需要修改诸多函数
13. 插入set，使用distance查看迭代器距离
14. 使用伸展树
15. Rank函数查询排名，不可以查询不存在元素；如果第一行有边界判断，可以设置返回-OO判断，是否存在改元素，用以快速替代find函数；可以设置判断返回0，用以查询前驱的排名；或者直接先插入，在查询，最后删除之；
16. Pred\_Succ求前驱或后继如果不存在，会直接返回0；
17. Pred\_Succ优先会满足前驱，如果修改>为>=会改为优先满足后继，优先满足不会考虑相同元素（严格不等）
18. get\_max/get\_min可由Select函数取代；
19. 可以使用Delete替代Find函数，方法是删除x，判断返回值是否等于x，不管等不等于，都重新插入x;