上海交通大学在线考试诚信承诺书

SJTU Online Examination Honor Code Letter

考试不仅是对学习成效的检查，更是对道德品质的检验。自觉维护学校的考风考纪，营造公平、公正的考试环境是全体同学的共同责任和义务。特别在疫情防控的特殊时期，更应强化自律意识，恪守诚信，拒绝舞弊，做一名诚实守信的新时代大学生，用诚信的考试构筑诚信的人生。

Examination is the evaluation of both learning effect and morality. It is the responsibility and obligation of all students to consciously maintain the school's common examination practice, abide by the discipline and create a fair and just examination environment. Especially in the special period of epidemic prevention and control, we should strengthen the consciousness of self-discipline, abide by the integrity, refuse to cheat, be an honest and trustworthy college student in the new era, and build an honest life from the integrity test.

我郑重承诺 I solemnly promise:

（1）本人将履约践诺，知行统一；遵从诚信规范，恪守学术道德；自尊自爱，自省自律。I will fulfill my promise, unify between knowledge and action, abide by the rules of integrity, academic ethics, be self-respected and self-disciplined.

（2）在线考试过程中，自觉遵守学校和老师宣布的考试纪律（详见《上海交通大学本科生学生手册》中的《学生考试纪律规定》，沪交教【2019】28 号），不剽窃，不违纪，不作弊。In the process of online examination, I will consciously abide by the examination discipline announced by the school and the teachers (see the regulations on student examination discipline in the undergraduate student handbook of Shanghai Jiao Tong University, HJJ [2019] No. 28), and do not plagiarize, violate discipline or cheat.

（3）若违反相关考试规定和纪律要求，自愿接受学校的严肃处理或处分。In case of violation of relevant examination regulations and discipline, students shall bear the serious treatment or punishment from the school.

承诺人 Committed by: 李昱翰

（学号 Student No: 522022 910279）

日期 Date (Y/M/D): 2022 年 6月 17日

上 海 交 通 大 学 答 题 纸

（ 20 21 至 20 22 学年 第 2 学期 ）

班级号 F2003702 学号 5200219 0279 姓名 李昱翰

课程名称 高级数据结构 成绩 _____

我承诺，我将严格遵守考试纪律。

承诺人：李昱翰

题号										
得分										
批阅人(流水阅卷教师签名处)										

上 海 交 通 大 学 答 题 纸

（ 20 21 至 20 22 学年 第 2 学期 ）

课程名称 高级数据结构(A) 姓名 李翰

一、单选题：

1、A 2、D 3、B 4、D 5、A

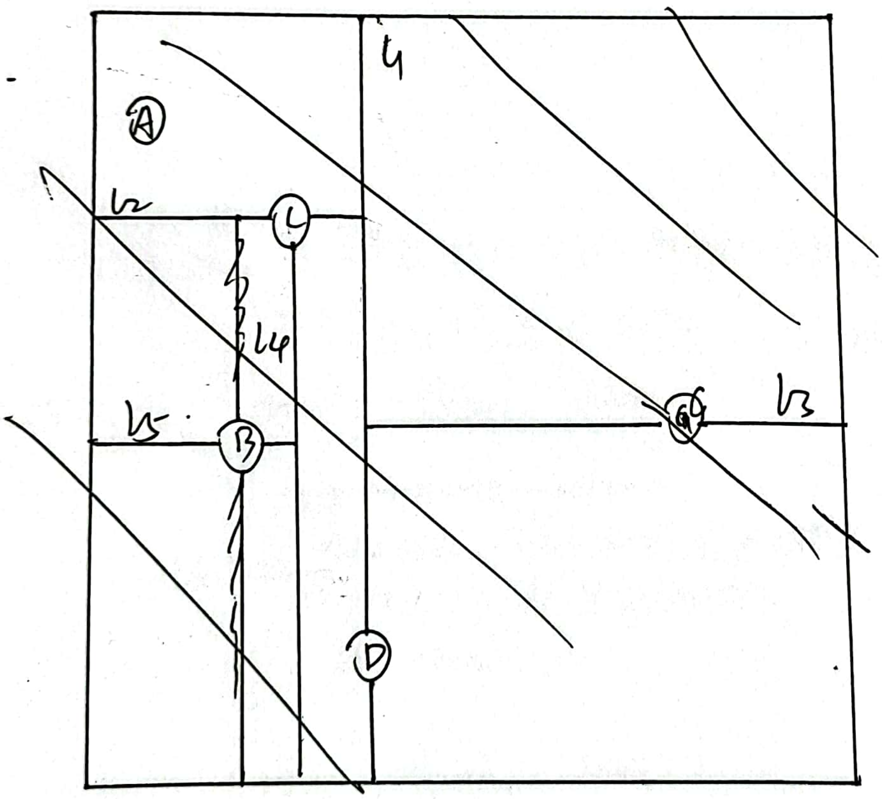
6、C 7、B 8、C 9、A 10、B

二、简答题：

1、最短路径为：A→B→C→E→D→G→H→I

最短距离为：1+3+1+2+3+2+4=16

2.

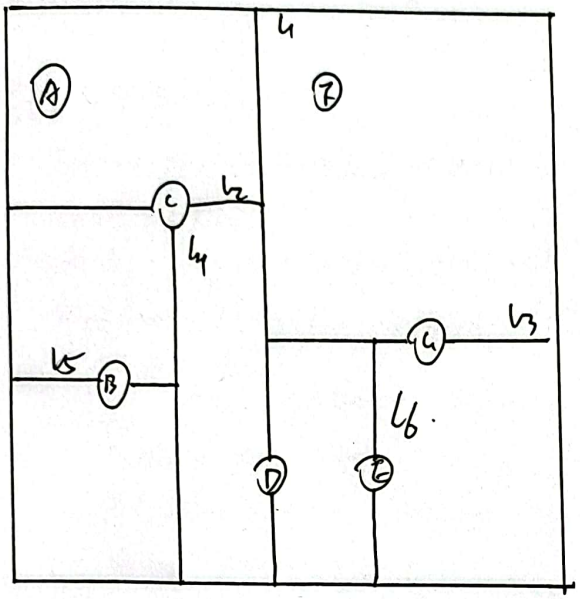


上 海 交 通 大 学 答 题 纸

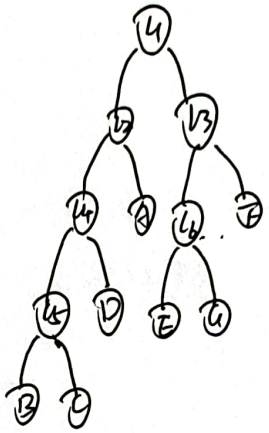
（ 20 21 至 20 22 学年 第 2 学期 ）

课程名称 高级数据结构 (A)　　姓名 赵翰

2.



树的结构：



3. (1): 红黑树相对于 AVL 树而言，其高对平衡要求不是很严格，故进行插入时 红黑树时进行旋转次数较少而性能更好。

(2): 由于红黑树对平衡要求低于AVL树故 红黑树进行查找时 时平均高度更高而使得查找性能弱于 AVL 树。

(3): 原因：顺序插入时时，时数据的连续性及有续性，往往只需进行单旋即可达到平衡，而乱序插入时数据不确定性，往往多次需要双旋进行调整，故往往顺序插入优于乱序插入。

4. (1): Lock-Hash 相对于 普通 Hash 而言，理论上有有相同的查找性能，但 Lock-Hash 的 PUT 性能时可能 block 或成环 而弱于普通 Hash、但是 Lock-Hash 发生冲突可能性小 且不需额外空间进行存放 key、

(2): ~~集群进行~~

上 海 交 通 大 学 答 题 纸

(20 21 至 20 22 学年 第 2 学期)

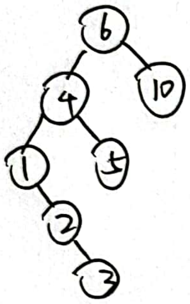
课程名称 ADS(A)　　姓名 王昱翰

2.2：①：插入4时，检查前后两半数组，发现两个位置均不为空

②：进行kick操作：首先4将前半数组中的2踢出，

③：发现后半数组中2对应位置为空，故将4放入原来2的位置，将2放入后半数组中的位置，重排结束。

5：查找6之后树的结构为



6. ①：设计思路：在写者进程进入后取锁，并在写入完成后放锁；

在读者进入之后也取锁，并在读取完成后放锁。

②伪代码：

```
void Thread_Write( )
{
    std::mutex mtx;
    std::lock lk(mtx); // mtx为定义好的全局变量
    write(); // 进行write
    lk.unlock();
}
void Thread_Read()
{   std::lock lk(mtx);
    read(); // 进行read操作
    lk.unlock();
}
```

③：若只有一个写者线程与一个读者线程，则可以不使用锁，而只使用如2个条件变量来控制读写，(如 empty、full) 来提升性能

上 海 交 通 大 学 答 题 纸

（ 20 21 至 20 22 学年 第 2 学期 ）

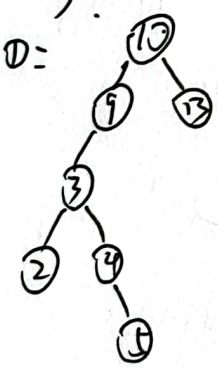
课程名称 ADS(A) 姓名 李昱翰

三、编程和实验题：

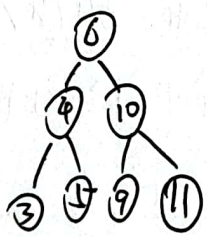
1. ①：

```
~~int size = nodes.size();~~
~~int end =~~
Node* RebuildNodes(vector<Node*>& nodes, int start, int end).
{ ~~if(start == end) return~~ //定义一个外部函数.
  ~~Node* t =~~
  ~~int~~ mid = (start+end)/2;
  Node* t = nodes[mid];
  if(start == end) return ~~t~~ t;
  RebuildNodes(nodes, start, mid-1);
  RebuildNodes(nodes, mid+1, end);
  return t;
}
Node* Rebuild(vector<Node*>& nodes).
{
  RebuildNodes(nodes, 0, nodes.size()-1);
}.
```

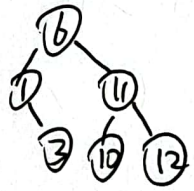
1：①：



②：



③：



上海交通大学答题纸

（ 2021 至 2022 学年 第 2 学期 ）

课程名称 ADS(A)　　　　姓名 李昱翰

(3)：①：插入：若采用权重平衡来检查是否重构，则与利用高度平衡进行检查相比，会更可能引起更多次的重构，从而使得插入平均重构次数上升而使性能下降。

②：查找：若采用权重平衡，则与高度平衡相比，多数情况下，在同时刻树的高度更低，查找相同结点时遍历平均层数更少，故性能上升。

一、(1)：①：方法一：直接遍历。即先检查跳表中所有满足条件的点，并存入一个vector中，之后再逐层检查磁盘中存储的SST文件，也存入vector中，之后对vector进行排序并返回。（header信息[min,max]）

②：方法二：也是首先检查跳表，之后对于SST采用类似于合并时的多路归并操作，对所有文件按key排序，之后利用多路归并取出所有符合条件的值，此时也无需再排序（此方法中一次性读出所有SST信息）。

③：优劣：①：方法一相对方法二，思想更直接也更易实现。

②：两种方法均需进行排序，但方法二是进行预处理(O(nlogn))，之后即类似二分查找，仅需O(logn)，明显快于方法一O(n)。

③：方法一多次访问disk也会使性能下降。

(2)：①：不能。②：若在非最后一层删除，~~因为存在可能还未到最底层就出现又插入一个"键key=k值value="~DELETED~"的情况~~ ~~若直接删除则可能~~ 由于新插入又生成新的SST，则无法区别实现操作先后顺序的识别，即可能先插入后删除这种情况若全部标记则无法识别。

(3)：①：伪代码：

T_i：第i层多出的SSTable文件集合。~~T_{all}：占了层中有key的集合~~

~~若对则删除~~：

~~for 每个 T_i：move(T_i, dst), (dst为第i层目录)~~

~~若有~~ 将对应T_i文件分别运行：move(T_i[i], dst)，

(dst为下一层目录)，之后删除对应T_i文件。

右边缘（截断）：①：优化……故可……可直接……3. ①：……②：……

上 海 交 通 大 学 答 题 纸

（ 2021 至 2022 学年 第 2 学期 ）

课程名称 ADS(A) 姓名 王昱翰

②优化理由：由于只有 insert与 get 和 delete，且 put 的值单增 且不重复，故可保证 SST 中不会出现相同值（因为每次合并会删除旧的），故可直接 move

3. ①主要思路：首先将数组根据线程数进行划分，（注意最后1组可能不均分），之后让每个子线程对自己部分进行二路归并排序，之后主线程利用多路归并实现合并操作，并返回最终数组。

②伪代码：

```
void main_thread(vector<int> a, int N) // N为thread_num
{
    int size = a.size();
    int piece = size / N; // 初始分
    for (int i=0; i<N-1; i++)
        std::threads.emplace_back(sub_thread, ref(a), start, end);
    // start_end为本分区起始终点
    std::threads.emplace_back(sub_thread, ref(a), end[N-1], a.size()-1);
    // 最后一块可不均分
    for (int i=0; i<N; i++) threads[i].join(); // 回收
    // 之后对每块进行多路归并合并
    while (true) {
        bool update = false;
        for (int i=0; i<N; i++) {
            if (pos[i] == end[i]) continue; // 到结尾
            update = true;
            if (i == 0 || a[i][pos[i]] == min)
                min_pos.push_back(i); // 第i块当前位最小
            if (a[i][pos[i]] < min)
                重置 min_pos,
                更新 min;
        }
        if (update == false) break; // update为false 则全处理完，则合并结束
    }
```

上海交通大学答题纸

（20_21至20_22学年 第_2_学期）

课程名称 ADS

姓名 李呈翰

```
void sub_thread (vector<int> & a, int start, int end) // 子线程
{
    int mid= (start +end)/2 ;
    if(start==end) return;
     mergesort( start, mid-1, a);
      mergesort ( mid+1,end, a);
      mergeTwoPart ( a, mid, start, end) ; // 进行部分二路归并
}
```