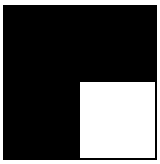
# 上 海 交 通 大 学 试 卷（B 卷）

（ 2019 至 2020 学年 第 2 学期 ）

班级号_____ 学号_____ 姓名 _____

课程名称_____《算法设计与实现》_____成绩 _____

1. （10 分）设 B 是一个$n \times n$的棋盘，其中 $n = 2^k$。试问，如何试用一个 L 型的条块(面积为 3)去覆盖 B 的除一个方格以外的所有方格。如，当 k=1 时，使用 1 块 L 型条块来覆盖，如下：



2. （10 分）**汉密尔顿图问题（Hamilton problem）**：给定一个无向图，请问该图是否存在一个环，在该环上每个顶点出现仅出现一次。

**旅行商问题（TSP problem）**：给定一个带权无向图，找出权重最小的一个环，在该环上每个顶点出现且仅出现一次。

试写出从汉密尔顿图问题到旅行商问题的多项式规约。

3. （10 分）给出一个线性时间的算法，找出有向图中一条路径长度为奇数的环。

4. （15 分）在一场体育锦标赛中，有$n$个球队，我们将其标记为 *1,…,n.*

(1) 每一个球队已经赢了 $x_i$ 场比赛

(2) 仍然有$m$ 场比赛，每场比赛具有两个参数$i$、$j$，满足$1 \le i \le j \le n$，表示比赛的双方。为了简化问题，我们假设每两支队伍之间剩下最多一场比赛，因此 $|m| \le O(n^2)$.

如果某一支球队可以赢得$z$场比赛，则它就赢得了最后的锦标赛。假设目前对于所有的球队 $i$， $z > x_i$，试问，是否仍然存在可能在 m 场比赛后，至少有一支球队赢得锦标赛？

5. （15 分）考虑**三分问题（3-Partition problem）**：给定整数集合$\{a_1, a_2, \dots, a_n\}$，判断是否可以将其分割为三个相互不相交的子集$I, J, K$，使得

$$\sum_{i \in I} a_i = \sum_{j \in J} a_j = \sum_{k \in K} a_i = \frac{1}{3}\sum_{i=1}^{n} a_i$$

6. （10 分）**Log-structured Merge Tree** 是键值存储系统中非常流行的数据结构。小 Y 同学和大家一样，根据本学期的 project 要求实现了一颗自己的 LSM 树。小 Y 的 LSM 树是一个单线程的程序，只实现了最基本的 LSM 树结构和操作，全局只有一个 memtable，且并未实现 bloom filter 和持久化操作日志等高阶功能。为了验证自己的实现效果如何，小 Y 设计了一个不断进行随机插入和删除操作的程序（每个插入操作的 value 大小相同），在每个操作完成后，小 Y 的程序都会打印一行信息表示操作成功。在测试运行时，小 Y 同学发现了一个奇怪的现象：

**我承诺，我将严格遵守考试纪律。**

**承诺人：_____**

| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 总分 |
|---|---|---|---|---|---|---|---|---|---|---|
| 得分 | | | | | | | | | | |
| 批阅人(流水阅卷教师签名处) | | | | | | | | | | |

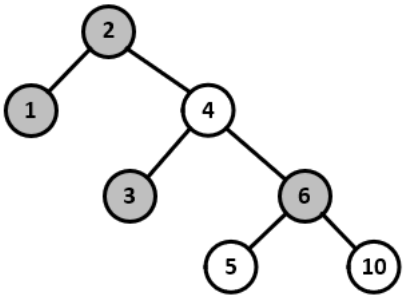
测试开始时，操作完成的非常快，操作成功的提示飞快地“刷屏”，而每隔一段时间，屏幕上的打印就会停下来，再过一段时间，又恢复飞快地“刷屏”。（提示：LSM 树在内存中使用 memtable 保存数据，在存储设备中用 sstable 保存数据。）

(1) 请解释为何测试时会出现“刷屏”和停滞两种不同的现象，且为何会周期性出现（3 分）。

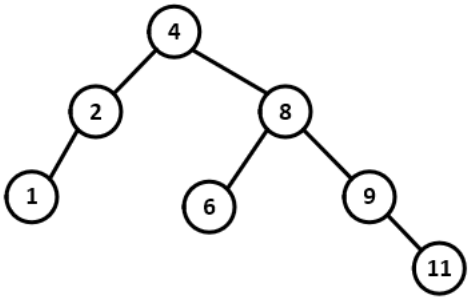
(2) 小 Y 随后将每个 sstable 的最大大小变大（从 2M 变成了 8M，memtable 的最大大小也相应进行了相同的变化），请问这会对“刷屏”和停滞两种现象的持续时间产生什么样的影响？请给出你的简要分析（4 分）。

(3) 请问有什么方法可以缩短每次停滞的时间（3 分）？（回答出一种方法即可）

7. （10 分）(1) 请画出以下红黑树插入节点 8 的过程（给出具体的每一步操作过程，如左旋/右旋，假定已经找到插入节点 8 的位置，不需要给出查找插入节点 8 的步骤。注：灰色节点为黑节点）（5 分）。




(2) 在如下 Splay 树中插入节点 5，请给出插入节点的整个过程（Bottom-Up），给出具体的每一步操作过程。假定已经找到插入节点 5 的位置，不需要给出查找结点 5 的步骤。（5 分）。



8. （10 分）现在有 n 个节点的有向图，节点依次标记为 1, 2, 3, …, n，存在一个列表 T，其中每个元素表示一段水流经过每条有向边的传递时间，表示为 T[i] = (u, v, w) ，其中 u 是源节点，v 是目标节点，w 是从节点 u 到节点 v 需要经过的时间（如果不在 T 中，表示两个节点没有有向边连接）。现在我们从节点 M 注入一段水流（假设水流无限），经过多久能使水流漫过所有节点？请简述你的思路，并完成代码。

```
class Solution {
public:    int getTime(vector<vector>& T, int n, int M) {
// 在此完成解决方法，若无法漫过所有节点，则返回-1
  }
};
```

9. （10 分）现有一种新的 bloom filter 的设计，对于一个值 v，通过四个哈希函数（hash1~hash4）产生四个哈希值。新的 bloom filter 并非将四个哈希值对应的 bit 置为 1，而是采取以下策略：

第一个哈希值对应的 bit 置为 1；

只有当值 v 二进制的从右往左数第三个 bit 为 1 时，才将第二个哈希值对应的 bit 置为 1；

只有当值 v 二进制的从右往左数第二个 bit 为 1 时，才将第三个哈希值对应的 bit 置为 1；

只有当值 v 二进制的从右往左数第一个 bit 为 1 时（即最低位），才将第四个哈希值对应的 bit 置为 1；

(1) 请使用 C++语言写出这种 bloom filter 的 insert 和 check 实现，四个哈希函数已给出，可直接调用。buckets 的 index 和哈希值对应。变量声明和函数签名如下( 6 分)：

```
    char buckets[MAX_HASH];
void insert(unsigned v);
    bool check(unsigned v);
    unsigned hash1(unsigned v);
    unsigned hash2(unsigned v);
    unsigned hash3(unsigned v);
    unsigned hash4(unsigned v);
```

(2) 应用这种新的设计后，相比原先的 bloom filter（即所有哈希值对应的桶都置 1），假阳性和假阴性会如何变化？（4 分）