

实验二、数据库系统算法的实现

姓名：瞿久尧 学号：120L022314

1 实验环境的构建

1.1 记录生成

每条记录使用一个类对象 Record 来记录，Record 中包含 3 个属性：num 为数值型，str 为字符串型，syn 为 int 型，并不参与记录，只是会在临时文件中记录，表示该 record 所在块号

同时该类中会重写 compareTo 方法，方便后期直接调用集合的排序方法

```
private static class Record implements Comparable<Record> {  
    5 个用法  
    int num;  
    3 个用法  
    String str;  
    3 个用法  
    int syn;  
}
```

记录为随机生成，但需要保证 20w 条记录不重复，因此需要用集合来记录已经存在的记录

```
Random rand = new Random();  
Set<Integer> usedNums = new HashSet<>();  
for (int i = 0; i < NUM_RECORDS; i++) {  
    int num = rand.nextInt(NUM_RECORDS);  
    while (usedNums.contains(num)) {  
        num = rand.nextInt(NUM_RECORDS);  
    }  
    usedNums.add(num);  
    String str = "Record " + num;
```

1.2 文件准备

使用 FileOutputStream 和文件名 data.dat 创建文件输出流用于存储我们随机生成的 20w 条记录

```

FileOutputStream fos = new FileOutputStream(FILE_NAME);
DataOutputStream dos = new DataOutputStream(fos);
Random rand = new Random();
Set<Integer> usedNums = new HashSet<>();
for (int i = 0; i < NUM_RECORDS; i++) {
    int num = rand.nextInt(NUM_RECORDS);
    while (usedNums.contains(num)) {
        num = rand.nextInt(NUM_RECORDS);
    }
    usedNums.add(num);
    String str = "Record " + num;
    dos.writeInt(num);
    dos.writeUTF(str);
}
dos.close();
fos.close();

```

将记录中的 num 和 str 写入文件中

2 基本算法设计

2.1 分块并对块中记录进行排序

由于记录太多，因此选择将 20w 条记录分成 20 份，每块 1w 条记录，每读取 1 块，就将这一块进行排序并存到一个临时文件中

这里排序采用的是快速排序，时间复杂度为 $O(n\log n)$

```

List<Record> block = new ArrayList<>();
for (int i = 0; i < BLOCK_SIZE && dis.available() > 0; i++) {
    int num = dis.readInt();
    String str = dis.readUTF();
    Record record = new Record(num, str, j);
    block.add(record);
}
j++;
Collections.sort(block);
File sortedRun = saveSortedRun(block);
sortedRuns.add(sortedRun);

```

2.2 对硬盘中的文件进行归并排序

将存在硬盘中的临时文件进行归并排序，这里使用了优先队列的数据结构，将每个流的第一条数据放入优先队列中，然后从优先队列中挑选最小的数据写到目标文件中，随后根据被挑选出来的数据的块号 syn 找到对应的输入流，在对应输入流加入下一条数据到优先队列中，直到优先队列所有记录都被写入到目标文件中

```

PriorityQueue<Record> pq = new PriorityQueue<>();
for (DataInputStream dis : streams) {
    int num = dis.readInt();
    String str = dis.readUTF();
    int syn = dis.readInt();
    Record record = new Record(num, str, syn);
    pq.add(record);
}
while (!pq.isEmpty()) {
    Record minRecord = pq.poll();
    dos.writeInt(minRecord.num);
    dos.writeUTF(minRecord.str);
    DataInputStream dis = streams.get(minRecord.syn);
    if (dis.available() > 0) {
        int num = dis.readInt();
        String str = dis.readUTF();
        int syn = dis.readInt();
        Record record = new Record(num, str, syn);
        pq.add(record);
    }
}

```

3 程序运行结果

3.1 随机生成记录

```

NULSTX!NUL
Record 139732NULSTXNjNUL
Record 151146NULSOHNAKNUL
Record 115221NULSOHnsNULFFRecord 93811NULSTXI NUL
Record 149967NULSOHbNULFFRecord 90826NULSTX^NUL
Record 155334NULSTX NUL
Record 178103NULSOH&NULFFRecord 75422NULSTXM+NUL
Record 150827NULSTX NUL
Record 195376NULSTXSUB NUL
Record 137796NULSTXrNUL
Record 184178NULNULDELNULFFRecord 53375NULSTXzFFNUL
Record 154124NULSOHapNULFFRecord 90480NULSTX NUL
Record 179075NULNULNULFFRecord 52375NULNUL$XNULVTRecord 9304NULNUL"NULFFRecord 38946NULSTXC
Record 148366NULSTXRS5NUL
Record 138805NULNULP$NULFFRecord 20516NULNULSTXNULFFRecord 64258NULNUL NULFFRecord 44029NUL
NULFFRecord 72202NULSOHVDELNULFFRecord 87679NULNUL8eNULFFRecord 14437NULSTX NUL

```

3.2 排序后的记录

```
NULNULNULNULNULBSRecord 0NULNULNULSOHNULBSRecord 1NULNULNULSTXNULBSRecord 2NULNULNULETXNULBSReci ✓
NUL Record 10NULNULNULVTNUL Record 11NULNULNULFFNUL Record 12NULNULNUL
NUL Record 13NULNULNULSONUL Record 14NULNULNULSINUL Record 15NULNULNULDLENUL Record 16NULNULNUL
Record 100NULNULNULeNUL
Record 101NULNULNULfNUL
Record 102NULNULNULgNUL
Record 103NULNULNULhNUL
Record 104NULNULNULiNUL
Record 105NULNULNULjNUL
Record 106NULNULNULkNUL
Record 107NULNULNULlNUL
Record 108NULNULNULmNUL
Record 109NULNULNULnNUL
Record 110NULNULNULoNUL
```

3.3 输出总时间

```
R:\aCodingTools\Jdk\JDK8\bin\java.exe ...
Sorting took 28434 milliseconds
```