

Real World Physics

对声音与波形关系的一些思考

上海交通大学致远学院 12 级 ACM 班

刘爽 5112409048

华逸青 5120309062

王奕仑

June 25, 2013

Abstract

为什么不同乐器听起来的声音不同？为什么人耳能够区分同一频率的不同乐器？
为什么人耳能区分同时奏响的不同频率的声音？为什么从声音中
能听到情感的表达？自然界的聲音可以完全用简洁的
数学表达吗？音乐数字化会遇到哪些问题？

Contents

1	动机阐述	2
1.1	Summary	2
1.2	Motivation	2
2	研究过程	3
2.1	对不同乐器频谱图的分析	3
3	成果概述	4
3.1	Gains	4
3.2	Advices	4
A	Testing Scripts	5
B	Testing Main Programs	6
B.1	MainArrayList	6
B.1.1	MainArrayList.cpp	6
B.1.2	MainArrayList.java	6
B.2	LinkedList	7
B.2.1	LinkedList.cpp	7
B.2.2	LinkedList.java	7
B.3	HashSet	8
B.3.1	HashSet.cpp	8
B.3.2	HashSet.java	8
B.4	TreeSet	8
B.4.1	TreeSet.cpp	8
B.4.2	TreeSet.java	9

Chapter 1

动机阐述

1.1 Summary

物理学引论 (1) 的课程中, 在讲解振动的知识时设计了一些声学的内容
在给定一个 `Utility.h` 头文件以及其余六个头文件的函数定义框架的情况下
补全每个函数体内部的内容。

最终需要实现线性查找表 `ArrayList` 与 `LinkedList`, 哈希查找表 `HashSet` 与
`HashMap`, 树形查找表 `TreeSet` 与 `TreeMap` 六个文件。

1.2 Motivation

首先, 进行这次大作业是为了完成本学期数据结构课的学习任务。

其次, 作为对深入了解 Java 与 C++ 的一条途径, 我把这次作业看成是对
Java 与 C++ 的语法与内部架构的复习。

再次, 这次作业也是为了给以后写大型工程打好基础。

Chapter 2

研究过程

2.1 对不同乐器频谱图的分析

Chapter 3

成果概述

3.1 Gains

- 提高了 C++ 的编译能力，对 C++ 与 Java 的区别有了更感性的认识
- 加强了写工程的能力

3.2 Advices

- 希望大作业可以提供更高的自由度
- 希望大作也可以有更加详尽的说明文档

Appendix A

Testing Scripts

该脚本是用 bash 编写的，每次读取当前要测试的主程序名（如 MainArrayList），初始数据规模（如 10000），最大数据规模（如 100000000），脚本对于每个数据规模分别运行 Cpp 与 Java 的主程序，输出运行时间。

```
#!/bin/bash

echo -n "Your program name: "
read program_name
echo -n "Begin Value: "
read beginValue
echo -n "End Value: "
read endValue

echo ""
echo "#####"
echo "          Testing starts...          "
echo ""

num=0;
for ((N=$beginValue; N<=$endValue; N=N*10))
do
    ((num++))
    echo "TEST CASE $num"
    echo "size: "$N
    echo -n "My Cpp STL: "
    (time -p ./ $program_name $N) 2>&1 | grep real | sed 's/real/TIME/'
    echo -n "Offical Java STL: "
    (time -p java $program_name $N) 2>&1 | grep real | sed 's/real/TIME/'
    echo ""
done

echo "          All tests done.          "
echo "#####"
echo ""
```

Appendix B

Testing Main Programs

每个 Java 与 Cpp 的主程序都从 main 参数读入数据规模 N，然后运行。不执行 IO。

B.1 MainArrayList

B.1.1 MainArrayList.cpp

```
int main(int argc, char **argv) {  
  
    int N = atoi(argv[1]);  
    int tot = N / 10;  
  
    ArrayList<int> arr;  
    for (int te = 0; te < 10; ++te) {  
        for (int i = 1; i <= tot; ++i) arr.add(i);  
        for (int i = 0; i < arr.size(); ++i) arr.get(i);  
        while (!arr.isEmpty()) arr.removeIndex(arr.size() - 1);  
    }  
  
    return 0;  
}
```

B.1.2 MainArrayList.java

```
public class MainArrayList {  
    public static void main(String args[]) {  
  
        int N = Integer.parseInt(args[0]);  
        int tot = N / 10;  
  
        ArrayList<Integer> arr = new ArrayList<Integer>();
```



```

    for (int te = 0; te < 10; ++te) {
        for (int i = 1; i <= tot; ++i) arr.add(i);
        for (int i = 0; i < arr.size(); ++i) arr.get(i);
        while (!arr.isEmpty()) arr.remove(arr.size() - 1);
    }
}

```

B.2 LinkedList

B.2.1 LinkedList.cpp

```

int main(int argc, char **argv) {

    int N = atoi(argv[1]);
    int tot = N / 10;

    LinkedList<int> lnk;
    for (int te = 0; te < 10; ++te) {
        for (int i = 1; i <= tot; ++i) lnk.add(i);
        while (!lnk.isEmpty()) lnk.removeLast();
    }

    return 0;
}

```

B.2.2 LinkedList.java

```

public class MainLinkedList {
    public static void main(String args[]) {

        int N = Integer.parseInt(args[0]);
        int tot = N / 10;

        LinkedList<Integer> lnk = new LinkedList<Integer>();
        for (int te = 0; te < 10; ++te) {
            for (int i = 1; i <= tot; ++i) lnk.add(i);
            while (!lnk.isEmpty()) lnk.removeLast();
        }
    }
}

```

B.3 HashSet

B.3.1 HashSet.cpp

```
int main(int argc, char **argv) {

    int N = atoi(argv[1]);
    int tot = N / 10;

    HashSet<int, Hashint> set;
    for (int te = 0; te < 10; ++te) {
        for (int i = 1; i <= tot; ++i) set.add(i);
        for (int i = 1; i <= tot; ++i) set.remove(i);
    }

    return 0;
}
```

B.3.2 HashSet.java

```
public class MainHashSet {
    public static void main(String args[]) {

        int N = Integer.parseInt(args[0]);
        int tot = N / 10;

        HashSet<Integer> set = new HashSet<Integer>();
        for (int te = 0; te < 10; ++te) {
            for (int i = 1; i <= tot; ++i) set.add(i);
            for (int i = 1; i <= tot; ++i) set.remove(i);
        }
    }
}
```

B.4 TreeSet

B.4.1 TreeSet.cpp

```
int main(int argc, char **argv) {

    int N = atoi(argv[1]);
    int tot = N / 10;

    srand(time(NULL));
    TreeSet<int> tree;
    for (int te = 0; te < 10; ++te) {
```

```

        for (int i = 1; i <= tot; ++i) tree.add(rand());
    }

    return 0;
}

```

B.4.2 TreeSet.java

```

public class MainTreeSet {
    public static void main(String args[]) {

        int N = Integer.parseInt(args[0]);
        int tot = N / 10;

        Random random = new Random();
        TreeSet<Integer> tree = new TreeSet<Integer>();
        for (int te = 0; te < 10; ++te) {
            for (int i = 1; i <= tot; ++i) tree.add(random.nextInt());
        }
    }
}

```