

# 实验指导书

## 1.跳表实现与分析

### 问题描述

实现并分析跳表结构。

### 基本要求

1. 构造并实现跳表 ADT，跳表 ADT 中应包括初始化、查找、插入、删除指定关键字的元素、删除关键字最小的元素、删除关键字最大的元素等基本操作。
2. 生成测试数据并验证你所实现的跳表结构的正确性。
3. 分析各基本操作的时间复杂性。
4. 对跳表维护动态数据集合的效率进行实验验证。给定随机产生的  $N$  个数据并将其初始化为严格跳表，在此基础上进行一系列插入、删除、查找操作（操作序列可以随机生成），观察在操作序列执行后，跳表结构的改变对跳表各操作的执行时间的影响。

可以设计如下实验：

- 随机产生的  $N$  个数据并将其初始化为严格跳表；
- 设定每个操作序列中包含插入、删除（包括删除指定关键字的元素、删除关键字最小的元素、删除关键字最大的元素）、查找操作的次数都是  $K$  次（操作数据随机生成），在初始跳表的基础上，依次执行  $M$  个操作序列，统计每个操作序列中各个操作执行所需的平均时间（以元素的比较次数衡量），获得随着  $M$  的增加而导致的操作时间的变化情况。分析产生这样变化的原因。当操作时间大到一定程度后应进行跳表的整理操作，设计相应的整理算法，并从数量上确定何时较为合适。观察在添加整理操作后执行时间的变化情况。

### 程序正确性的验证

你需要设计程序验证你所实现的跳表的正确性，本次实验我们提供了示例数据集供你验证程序的正确性，数据集可以在此处下载。除了我们所提供的数据集，你也需要在自己生成的数据上验证程序的正确性。你可以通过如下工具比较输出文件是否一致：

- Web Tools: <https://text-compare.com/>

- Windows: 在命令行中使用 `FC` 命令, `FC file1 file2`
- OSX and Linux: 在终端中使用 `diff` 命令, `diff file1 file2`

## 成绩

实验分数由现场验收与实验报告组成, 验收时应对你所实现的代码进行讲解并当场验证你所完成的程序的正确性。以下是可能导致你成绩降低的因素:

1. 跳表基本功能实现不完全, 缺少基本操作。
2. 对程序代码理解不清楚, 对于特定的操作不能解释其实现原理。
3. 不能验证程序的正确性。程序只能接受手动输入的数据, 不能在大规模随机数据上验证程序各个操作的正确性。
4. 缺少跳表各操作时间复杂度与跳表效率的分析。

对于实现规范, 分析全面合理的同学将获得较高的分数。以下是获得较高成绩的几个需要考虑的方面:

1. 程序设计规范, 封装良好, 拥有详细的注释。
2. 合理且完善的跳表性能的分析实验的设计, 对实验结果的详细分析。

## 提交详情

实验验收的截止时间是 **第 5 周的实验课下课前**, 实验代码与报告的提交截止时间是 **第 6 周周日 23:59 前**, 在截止时间后验收与提交会导致你的实验成绩下降, **所以请及时验收与提交**。本次实验应该独立完成, 请避免参考其他同学的代码, 我们会对提交的代码进行查重, 查重率过高会影响你的实验成绩。

实验代码与实验报告请提交至实验平台, 超过提交的截止时间将无法在平台上提交, 若未能在截止时间内提交将影响实验成绩。实验代码中无需包含测试数据, 并请压缩成 **100M** 以内的 `zip` 格式文件提交。

实验代码中应至少包含以下三个文件 `"skiplist.h"`、`"skiplist.cpp"`、`"main.cpp"`, 分别是你实现的跳表的头文件, 跳表实现文件与运行测试程序。其余部分程序例如数据生成程序, 正确性验证程序, 跳表性能分析实验等程序均可放于实验代码中一起上传。可以写一个简单的 `README` 来介绍各个文件的作用。

若是实验平台的账号没有注册或忘记密码, 请联系你所在实验室的助教或联系 QQ: 2741591545。

## 2. 外排序

### 问题描述

应用竞赛树结构模拟实现外排序。

### 基本要求

1. 设计并实现最小输者树结构 ADT，ADT 中应包括初始化、返回赢者，重构等基本操作。
2. 应用**最小输者树**设计实现外排序，外部排序中的生成最初归并串以及 K 路归并都应用**最小输者树**结构实现；
3. 随机创建一个较长的文件作为外排序的初始数据；设置归并路数以及缓冲区的大小；获得外排序的访问磁盘的次数并进行分析。可采用小文件来模拟磁盘块。

### 程序正确性的验证

你需要设计程序验证你所实现的目录系统的正确性，本次实验我们提供了示例数据集供你验证程序的正确性，数据集可以在此处[下载](#)。

### 成绩

实验分数由现场验收与实验报告组成，验收时应对你所实现的代码进行讲解并当场验证你所完成的程序的正确性。

以下是可能导致你成绩降低的因素：

1. **最小输者树**基本功能实现不完全；
2. 外排序没有采用**最小输者树**
3. 对程序代码理解不清楚，对于特定的操作不能解释其实现原理；
4. 程序代码封装、嵌套混乱，出现大量冗余；
5. 不能验证程序的正确性。程序只能接受手动输入的数据，不能在大规模随机数据上验证程序各个操作的正确性。

以下是获得较高成绩的几个需要考虑的方面：

1. 程序设计规范，封装良好，代码有必要的注释；
2. 实现了基本操作之外的其他功能；
3. 能应对大规模随机数据；

以下是加分项：

1. 对内外存读取实现了更高程度上的模拟。

### 提交详情

实验验收的截止时间是 **第 9 周的实验课下课前**，实验代码与报告的提交截止时间是 **第 10 周周日 23:59 前**，在截止时间后验收与提交会导致你的实验成绩下降，**所以请及时验收与提交**。本次实验应该独立完成，请避免参考其他同学的代码，我们会对提交的代码进行查重，查重率过高会影响你的实验成绩。

实验代码与实验报告请提交至实验平台。超过提交的截止时间将无法在平台上提交，若未能在截止时间内提交将影响实验成绩。实验代码中无需包含测试数据，并请压缩成 **100M 以内** 的 zip 格式文件提交。

实验代码中应至少包含（包括但不限于）以下四个文件 "minLoserTree.h"、"minLoserTree.cpp"、"externalSorter.cpp"、"main.cpp"，分别是你最小输者树的头文件，最小输者树的实现文件，外排序的实现代码与运行测试代码。其余部分代码文件选择性提交，包括但不限于数据生成程序，正确性验证程序等。**提倡**写一个简单的 README 来介绍各个文件的作用。

若是实验平台的账号没有注册或忘记密码，请联系你所在实验室的助教或联系 QQ: 2741591545。

## 3. 模拟文件目录系统

### 问题描述

使用树结构实现一个简单文件目录系统的模拟程序。

### 基本要求

1. 设计并实现目录树 CatalogTree 的 ADT。
2. 应用以上 CatalogTree 结构设计并实现一文件目录系统的模拟程序。
3. 文件目录系统程序是一个不断等待用户输入命令的解释程序，根据用户输入的命令完成相关操作，直到退出（quit）。目录系统应支持如下基本操作：
  - (1) dir ——列出**当前目录下**的所有子目录与文件项。
  - (2) cd ——列出当前目录的绝对路径。
  - (3) cd ..——当前目录变为当前目录的父目录。
  - (4) cd str——当前目录变为 str 所表示路径的目录。
  - (5) mkdir str ——在(当前目录下)创建一个子目录(名为 str)，若存在则不进行任何操作。
  - (6) mkfile str ——在(当前目录下)创建一个文件(名为 str)，若存在则不进行任何操作。
  - (7) delete str ——删除(当前目录下)名为 str 的目录或文件，若不存在则不进行任何操作。

(8) `save str`—— 将从根节点开始的目录树结构保存到文件(名为 `str`)中。

(9) `load str` —— 从文件 `str` 中读取之前保存的目录树结构，并根据其重新建立当前目录树

(10) `quit` —— 退出程序

### 选做功能

你可以自行充实目录树支持的指令，添加一些参数支持，这将有助于你对目录系统的理解。提倡实现其他指令或带参数的现有指令。

#### tips:

你应区分出 `str` 代表的是绝对路径还是相对路径；  
指令内容或功能可以[参考这里](#)。

### 程序正确性的验证

你需要设计程序验证你所实现的目录系统的正确性，本次实验我们提供了示例数据集供你验证程序的正确性，数据集可以在[此处下载](#)。

### 成绩

实验分数由现场验收与实验报告组成，验收时应对你所实现的代码进行讲解并当场验证你所完成的程序的正确性。

以下是可能导致你成绩降低的因素：

1. 目录树基本功能实现不完全，缺少基本操作；
2. 对程序代码理解不清楚，对于特定的操作不能解释其实现原理；
3. 程序代码封装、嵌套混乱，出现大量冗余；
4. 不能验证程序的正确性。

以下是获得较高成绩的几个需要考虑的方面：

1. 程序设计规范，封装良好，代码有必要的注释；
2. 实现了基本操作之外的其他指令；
3. 实现了带参数指令。

### 提交详情

实验验收的截止时间是 **第 10 周的实验课下课前**，实验代码与报告的提交截止时间是 **第 11 周周日 23:59 前**，在截止时间后验收与提交会导致你的实验成绩下降，**所以请及时验收与提交**。本次实验应该独立完成，请避免参考其他同学的代码，我们会对提交的代码进行查重，查重率过高会影响你的实验成绩。

实验代码与实验报告请提交至实验平台,超过提交的截止时间将无法在平台上提交,若未能在截止时间内提交将影响实验成绩。实验代码中无需包含测试数据,并请压缩成 **100M 以内** 的 zip 格式文件提交。

实验代码中应至少包含以下三个文件 "catalogTree.h"、"catalogTree.cpp"、"main.cpp", 分别是你实现的目录树的头文件、实现代码与文件目录系统程序。其余部分代码文件选择性提交。**提倡**写一个简单的 README 来介绍各个文件的作用。

若是实验平台的账号没有注册或忘记密码,请联系你所在实验室的助教或联系 QQ: 2741591545。

## 4. 网络放大器设置问题

### 问题描述

对于一个石油传送网络可由一个加权有向无环图  $G$  表示。该图中有一个称为源点的顶点  $S$  (保证  $S$  的入度为 0), 从  $S$  出发, 石油流向图中的其他顶点.  $G$  中每条边上的权重为它所连接的两点间的距离。

在输送石油的过程中, 需要有一定的压力才能使石油从一个点到达另一个点, 但压力会随着路程的增加而降低 (即压力的损失量是路程的函数)。

因此为了保证石油在网络的正常运输, 在网络传输中必须保证在任何位置的壓力都要不小于最小压力  $P_{min}$ 。为了维持这个最小压力, 可在  $G$  中的一些或全部顶点放置压力放大器, 压力放大器可以将压力恢复至该网络允许的最大压力  $P_{max}$ 。

可以设  $d$  为石油从压力  $P_{max}$  降为压力  $P_{min}$  所走的距离, 在无压力放大器的情况下石油可运输的距离不超过  $d$ 。

在设置信号放大器问题中, 需要在  $G$  中放置最少数量的放大器, 使得该传输网络从  $S$  出发, 能够将汽油输送到图中的**所有**其他顶点。

### 基本要求

1. 给出两种方法以解决上述问题, 验证两种方法的正确性。
2. 比较两种方法的时间和空间性能, 用图表显示比较结果。

### 程序正确性的验证

你需要在不同的随机数据上验证你所实现算法的正确性, 本次实验我们提供了大量的单源点加权有向无环图的数据, 可以在[此处下载](#)。

## 成绩

实验分数由现场验收与实验报告组成，验收时应对你所实现的代码进行讲解并当场验证你所完成的程序的正确性。基本要求的每一部分都应在实验报告中作出详尽的阐述。以下是可能导致你成绩降低的因素：

1. 对程序代码理解不清楚，对于特定的操作不能解释其实现原理。
2. 程序只能接受手动输入的数据，不能在随机数据上验证算法结果的正确性。
3. 不能验证程序的正确性。
4. 缺少两种算法的时间复杂度与空间复杂度的分析与比较。

对于实现规范，分析全面合理的同学将获得较高的分数。以下是获得较高成绩的几个需要考虑的方面：

1. 程序设计规范，封装良好，拥有详细的注释。
2. 可以直观的展示你的程序的结果，包括但不限于放大器的位置，每个位置的应力，边权等。
3. 合理且完善的性能分析与图表展示
4. 对程序的正确性有推导或说明。

## 提交详情

实验验收的截止时间是 **第 11 周的实验课下课前**，实验代码与报告的提交截止时间是 **第 12 周周日 23:59 前**，在截止时间后验收与提交会导致你的实验成绩下降，**所以请及时验收与提交**。本次实验应该独立完成，请避免参考其他同学的代码，我们会对提交的代码进行查重，查重率过高会影响你的实验成绩。

实验代码与实验报告请提交至实验平台。超过提交的截止时间将无法在平台上提交，若未能在截止时间内提交将影响实验成绩。实验代码中无需包含测试数据，并请压缩成 **100M 以内**的 zip 格式文件提交。

实验代码中应至少包含以下两个文件 "solution1.cpp"、"solution2.cpp"，分别是你实现的两种解决方法，其中应**包含加权有向无环图类的定义和实现**。其余部分代码文件选择性提交，包括但不限于数据生成程序，正确性验证程序等。**提倡**写一个简单的 README 来介绍各个文件的作用。

若是实验平台的账号没有注册或忘记密码，请联系你所在实验室的助教或联系 QQ：2741591545。