《数据结构I》课程三级项目报告

专业：

班级：

学号：

姓名：

# 基于内存的半结构化存储系统

模块1：模拟线程池

1. 项目目的（介绍项目要解决的问题）

线程池依据不同的业务场景有不同的设计方式。本项目有以下三个对象需要完成。 ① 模拟运行状态线程监控表 为运行的线程模拟一个运行状态线程监控表（用简单的数组来模拟，然而实际上Linux内核里应该是一个双链表list\_head或者散列表hlist\_head） ② 模拟线程池 使用循环队列模拟，等待状态的线程入队。遇到新任务使用等待状态线程， 则出队变成运行状态的线程。任务运行结束后，线程入队。 ③ 模拟任务等待队列 使用链式队列模拟 。当新来任务发现线程池内已经没有等待状态线程了， 就进入任务等待队列。当发现循环队列有等待状态的线程时，任务队列队头出队，派发给循环队列也就是线程池内队头的线程去执行

2.项目代码

自行完成

将此项目的填充代码截图到此处，要求有自己的学号姓名。

3.项目流程

要求用**文字或图**的方式说明项目代码中main函数的执行流程，**此部分要求手写或者手画，之后拍照插入图片。**

3.项目结果

（1）测试数据，请对测试数据进行解释说明

测试数据使用：

T1 8

T2 5

T3 8

T4 2

T5 2

T6 2

T0 0

测试数据包括任务名和任务生命周期时间。

（2）运行结果（项目运行截图）

自行运行

4.项目思考问题解答

（1）在注释、命名、错误处理、可读性等方面还有一些不足之处，并缺乏项目实战经验

（2）将 task 定义为结构体类型的指针，可以提供更好的灵活性和可扩展性，使得在运行时动态地管理和操作 Task\_Info 结构体对象成为可能

（3）通过使用二级指针来定义结构体数组，可以实现更加灵活和高效的内存管理和数据访问方式，能够提高程序的可扩展性和可维护性。

模块2：模拟云计算弹性扩容最优费用下的资源选择

1.项目目的

（1）理解深度优先搜索；

（2）能够编码实现深度优先搜索的基本操作。

2.项目流程

要求：用文字或图的方式说明项目代码中dfs()函数的执行流程，**此部分要求手写或者手画，之后拍照插入图片。**

3.项目结果

（1）测试数据，请对测试数据进行解释说明

样例输入： 3 20 5 3 7 2 11 1

样例输出： 0.21

（2）运行结果（项目运行截图）

4.项目思考问题解答

一个数字只使用一次

允许数字复用但组合出的数字长度不超过数字个数

（1）只有一位数字，可以组合成1、2、3。

两位数字，可以组合成11、12、13、21、22、23、31、32、33。

三位数字，可以组合成111、112、113、121、122、123、131、132、133、211、212、213、221、222、223、231、232、233、311、312、313、321、322、323、331、332、333。

共有6+9+27=42个

（2）

状态搜索树的节点表示问题的状态，而树的边则表示状态之间的转移关系。根节点表示初始状态，而每个节点的子节点表示从父节点状态经过一步操作得到的新状态。通过遍历状态搜索树，可以找到问题的解，或者找到最佳解的路径。

状态集合是指问题中可能出现的所有状态的集合。在搜索算法中，需要维护一个状态集合，用来记录已经访问过的状态以及待访问的状态。状态集合可以帮助避免重复访问相同的状态，提高搜索效率。

在搜索算法中，状态搜索树和状态集合通常是密切相关的。搜索算法通过状态搜索树来探索可能的状态空间，并通过状态集合来记录已经访问过的状态，以便更高效地搜索问题的解空间。

总之，状态搜索树是用来描述问题状态和状态转移关系的树形结构，而状态集合则是问题中可能出现的所有状态的集合，在搜索算法中起着重要的作用。

模块3：半结构化数据的部分排序TOP-K问题

1. 项目目的

利用堆排序统计指定销量名次的品牌

2.项目流程

要求：用文字或图的方式说明项目代码：

void adjust\_heap(ADT arr[], unsigned len, unsigned hole)；int main()；

函数的执行流程，**此部分要求手写或者手画，之后拍照插入图片。**

这段代码实现了一个堆排序算法，并且在主函数中使用了这个算法来对销量前 M 名的商品进行排序和输出。

首先，让我们来看看 void adjust\_heap(ADT arr[], unsigned len, unsigned hole) 函数。这个函数用于调整堆，使得以 arr[hole] 为根的子树成为一个最小堆。在这个函数中，使用了很典型的堆调整的方式，通过递归地将当前节点和其孩子节点进行比较交换，直到满足堆的性质为止。

接下来是 int main() 函数。在这个函数中，首先定义了一个长度为 N 的 BRAND 结构体数组 data\_in，其中存储了商品的销量信息。然后，定义了一个长度为 N 的 ADT 结构体数组 array，并通过遍历将 data\_in 中的数据转化为 array 中的数据表示。接着调用了 partial\_sort(array, N, M); 对 array 进行部分排序，找出销量最高的 M 个商品。最后，通过循环输出了销量最高的 M 个商品的名称和销量信息。

整体流程如下：

1. 初始化数据：初始化了存储商品销量信息的数组 data\_in 和存储销量目标和指针的数组 array。
2. 数据转换：遍历 data\_in 数组，将其中的数据转化为 array 中的形式。
3. 部分排序：调用 partial\_sort 对 array 进行部分排序，找出销量最高的 M 个商品。
4. 输出结果：通过循环输出销量最高的 M 个商品的名称和销量信息。

3.项目结果

（1）测试数据，请对测试数据进行解释说明

10 5 样本容量与指定的需要统计的名次个数

29000 "LG" 销量与品牌

124000 "XIAOMI"

296000 "APPLE"

119000 "OPPO"

113000 "VIVO"

238000 "HUAWEI"

25000 "REALME"

39000 "LENOVO"

196000 "SAMSUNG"

66000 "ONEPLUS"

（2）运行结果（项目运行截图）

自行运行

4.项目思考问题解答

（1）使用堆结构进行部分排序的好处包括：高效、空间利用率高、可扩展性强、具有部分排序功能，适用于处理大规模数据和实时数据

（2）优先使用小顶堆。

模块4：半结构化数据的基于内存构造与查找

1.项目目的（介绍项目要解决的问题）

2.项目流程

要求：用文字或图的方式说明项目代码int main()；函数的执行流程，**此部分要求手写或者手画，之后拍照插入图片。**

3.项目结果

（1）测试数据，请对测试数据进行解释说明

（2）运行结果（项目运行截图）

4.项目思考问题解答

（1）

（2）