

数据与算法知识与方法

T^TT

2024 年 9 月 10 日

目录

0 绪论：数据、数学模型、算法	2	0.3 算法	2
0.1 数据及其结构	2	A LambdaOJ 平台介绍	4
0.2 数学模型	2	A.1 判题结果	4
		A.2 注意事项	4

0 绪论：数据、数学模型、算法

0.1 数据及其结构

定义 0.1.1. 数据的基本单元称为**数据元素**。

数据元素并不是孤立存在的，而是存在密切的联系。

数据的逻辑结构：

- **集合结构**：数据元素之间没有任何关系
- **线性结构**：数据元素之间存在一对一的关系，如线性表、栈、队列、串
- **树形结构**：数据元素之间存在一对多的关系，如树
- **图形结构**：数据元素之间存在多对多的关系，如图

数据的存储结构：

- **顺序存储结构**：将数据元素存放在地址连续的存储单元里
- **链式存储结构**：将数据元素存放在任意的存储单元里，通过指针相连

0.2 数学模型

定义 0.2.1. **数学模型**是对于现实世界的某一特定对象，根据其内在规律，为特定目的而得到的一个抽象的、简化的数学结构。

常用的数学模型的类型：

- **线性方程组**：超定线性方程组的线性拟合、欠定线性方程组的线性规划
- **非线性方程组**：非线性方程组的求解
- **微分方程**：常微分方程的数值解法、偏微分方程的数值解法
- **概率模型**：概率预测、概率统计
- **统计模型**：统计预测、统计分析
- **离散模型**：线性结构、树结构、图结构
- **优化模型**：线性规划、整数规划、非线性规划

0.3 算法

定义 0.3.1. **算法**是用以解决某一问题的有限长度的指令序列。

算法的基本特点：

- **有穷性**：算法必须在执行有穷步之后能够结束，且每一步都可在有穷时间内完成
- **确定性**：算法的每一步必须有确切的含义，算法的实际执行结果是确定的、且精确地符合要求或期望
- **可行性**：算法中描述的操作都可以通过已经实现的基本操作运算的有限次执行来实现
- **输入**：算法必须有零个或多个输入
- **输出**：算法必须有一个或多个输出

好的算法应该具有以下特点：

- **正确性**：算法应该能够解决问题
 - 不含语法错误

- 对一般的输入数据能够产生正确的输出结果
- 对精心选择的苛刻数据也能产生正确的输出结果
- 对于所有的输入数据都能产生正确的输出结果

- **可读性**：算法应该容易理解
- **健壮性**：算法应该能够处理各种异常情况
- **高效性**：算法应该能够在合理的时间和空间开销内解决问题

一个特定算法的运行工作量的大小，与问题规模的大小有关，这种关系称为算法的复杂度。

定义 0.3.2. 渐进时间复杂度

算法的（**渐进**）**时间复杂度**是指算法的运行时间与问题规模之间的关系。

例 0.3.1. 现有程序如下：

```
1  for(int i = 0; i < n; i++)
2      for(int j = i; j < n; j++)
3          if(a[i] > a[j])
4              swap(a[i], a[j]);
```

其中，基本操作swap函数的时间复杂度为 $O(1)$ ，其执行次数为 $\sum_{i=0}^{n-1} \sum_{j=i}^{n-1} 1 = \sum_{i=0}^{n-1} (n-i) = \sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$ ，则该

程序的时间复杂度为 $O\left(\frac{n(n-1)}{2}\right) = O(n^2)$ 。

例 0.3.2. 现有程序如下：

```
1  for(int i = 1; i < n; i++)
2      for(int j = 0; j < i; j *= 2)
3          a[i] += a[j];
```

其内层循环的时间复杂度为 $O(\log_2 i)$ ，则该程序的时间复杂度为

$$\begin{aligned} \sum_{i=2}^{n-1} O(\log_2 i) &= O\left(\sum_{i=2}^{n-1} \log_2 i\right) = O\left(\log_2 \left(\prod_{i=2}^{n-1} i\right)\right) \\ &= O(\log_2 (n-1)!) = O(\log_2 n!) = O(n \log_2 n) \end{aligned}$$

定义 0.3.3. 渐进空间复杂度

算法的（**渐进**）**空间复杂度**是指算法的空间开销与问题规模之间的关系。

A LambdaOJ 平台介绍

A.1 判题结果

判题有如下结果：

- ACCEPTED：结果完全正确。
- WRONG_ANSWER：程序正常执行，也没有超时和超内存，但是答案不对。
- TIME_LIMIT_EXCEEDED：程序超时，可能是死循环，也可能是算法不够优，或者实现不够优；还没到核对答案那一步，内存情况也不明。
- MEMORY_LIMIT_EXCEEDED：程序超过内存限制，可能是算法不够优，或者实现不够优；也没有到核对答案那一步。
- OUTPUT_LIMIT_EXCEEDED：输出数据过多，要么程序有 bug，要么是恶意代码。
- BAD_SYSCALL：使用了非法的系统调用，要么是恶意代码，大部分情况是 C++ 在 `new` 一块内存的时候，超过了限制失败了，C++ 运行时环境需要一个系统调用关闭信号，然后杀死这个进程。
- RUN_TIME_ERROR：运行时错误，可能是由于除零、引用空指针或者数组越界等等。

A.2 注意事项

尽量使用 `stdio.h` 而不使用 `iostream`，前者的输入输出效率更高，时间和空间开销更小。