Data Structure 02

MKQ

September 10, 2019

Contents

1 算法效率分析

- 时间耗费

1.1 语句的频度

算法中某条语句执行的次数

$$T = \sum frequencyofcodei$$

取每条语句执行一次的时间为单位一

- 很多情况下,T 是数据规模 n 的函数 T(n)

1.2 时间复杂度

假设 f(n) 是一个函数, 且有

$$lim\frac{f(n)}{T(n)} = C$$

记为:

$$T(n) = O(f(n))$$

例如一个三重循环的时间复杂度是

$$O(n^3)$$

这样就只关注数量级了, 而且只关注其中执行次数最多的那条语句前面的系数也不用管

1.2.1 注意

一些和数据初始状态相关的不确定性问题

- 按照平均情况估计
- 按照最坏的情况估计

1.2.2 常用复杂度

- O(1)
- O(log n)
- O(n)
- O(n log n)
- $O(n^c)$
- $O(c^n)$
- O(n!)

1.3 空间复杂度

- 数据元素
- 程序占用
- 辅助变量

前两者不评估第三个所需要的大小

$$S(n) = O(f(n))$$

2 typedef

- typedef int yourtype

3 作业

P60 1 2

4 线性表

- 定义
- 表示
- 实现
- 操作
- 应用

4.1 逻辑结构

```
Linear_list=(D,S)
D={a_1 ,a_2 ,a_3 ... a_n }
S={s}
s={<a_1 ,a_2 >,<a_2 ,a_3 >...}
```

- 是 n 个节点的有限序列
- 可以为空, 叫做空线性表

- 有一个头结点, 有一个尾节点
- 直接前驱, 直接后继
- 强调元素之间的相互关系

```
ADT List{
 Element:D={a_1 ...}
 Relation:R={<>,<>,<>...}
 Method:
     InitList(&L)
     DestoryList(&L)
     ...
}ADT List
```

- ADT 中包含的运算要足够基本
- 所以里面没有排序一类的

4.2 举例

4.2.1 就地运算

计算结果放在原来的线性表里面

4.2.2 异地运算

就是放在不同的表里面