

第一讲 基本概念

2019年5月3日, 星期五 上午 11:43

1. 什么是数据结构?

a. 引入图书馆存书的例子

i. 随便存放

1) 找书麻烦, GG找不到的兄弟;

ii. 按分类存放

按书的种类, 文科类, 工科类etc;在文科, 工科类细分, 建筑, 计算机blabla

好找, 但是类别分多细 是个值得考虑的问题

iii. 按书名音序存放

如果插入<<阿Q正传>>, 所有的书都要往后挪, 又要GG

小结a问题: 要考虑图书的插入和查找;

解决问题方法的效率 和数据的组织方式有关;

b. 计算多项式 $f(x)=$

$$\begin{aligned} & a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_{n-1}x^{n-1} + a_nx^n \quad \text{算法 } f_2 \\ & \left\{ \begin{aligned} &= a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + a_{n-1}x^{n-2} + a_nx^{n-1}))) \\ &= a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + a_{n-1}x^{n-3} + a_nx^{n-2}))) \\ &= a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + a_nx^{n-3}))) \\ &= a_0 + x(a_1 + x(a_2 + x(a_3 + x(a_4 + x(a_5 + \dots x a_n)))) \end{aligned} \right. \\ & \quad \text{算法 } f_1 \end{aligned}$$

比较算法1和算法2 的运行程序如下:

```
#include<stdio.h>
#include<time.h>
#include<math.h>
clock_t start,stop;
double duration;
#define MAXN 10
#define MAXK 1e7
double f1(int n,double a[], double x);
double f2(int n,double a[], double x);
//比较f1 和f2函数的运行时间
int main()
{
    int i;
    double a[MAXN];
    for(i = 0;i <MAXN;i++){
        a[i] = (double) i;
    }
    start = clock();
    for(i = 0;i<MAXK;i++){
        f1(MAXN-1,a,1.1);
    }
    stop = clock();
    duration = ((double)(stop - start))/CLK_TCK/MAXK;
    //MAXK: 因为调用一次的时钟花费太小,
    //精度不够,
```

```

//所以跑1e7次，再求平均。
printf("ticks1 = %f\n", (double)(stop-start));
printf("duration1 = %6.2e\n", duration);

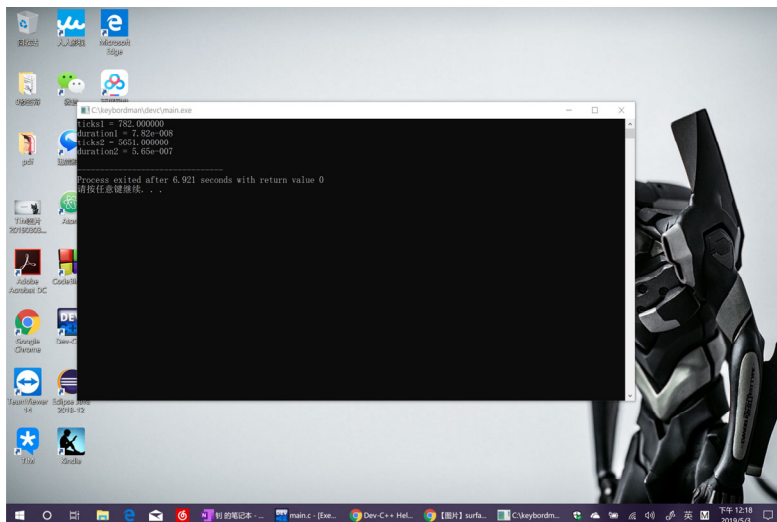
start = clock();
for (i = 0; i < MAXK; i++) {
    f2(MAXN-1, a, 1.1);
}
stop = clock();
duration = ((double)(stop - start)) / CLK_TCK / MAXK;
//MAXK: 因为调用一次的时钟花费太小,
//精度不够,
//所以跑1e7次，再求平均。
printf("ticks2 = %f\n", (double)(stop-start));
printf("duration2 = %6.2e\n", duration);
return 0;
}

//计算多项式 $f(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1} + a_nx^n$ 
//f1算法: 从n提x开始算
//f2算法: 直接 $a_0 + a_1x + \dots$ 开始算
double f1(int n, double a[], double x) {
    int i;
    double p = a[n];
    for (i = n; i > 0; i--) {
        p = a[i-1] + x * p;
    }
    return p;
}

double f2(int n, double a[], double x) {
    int i;
    double p = a[0];
    for (i = 1; i <= n; i++) {
        p += a[i] * pow(x, i);
    }
    return p;
}

```

运行结果



观察得f1的运行时间 远远小于 f2，不是一个数量级。

引出概念: **什么是数据结构?**

数据对象在计算机中的组织方式

- 逻辑结构
- 物理存储结构

数据对象必定与一系列加在其上的**操作**相关联

完成这些操作所用的方法就是**算法**

数据对象集 + 操作集 = D T
Data Type

抽象: 和实现的具体操作无关, 不关心怎么做到, Abstract

