C心得

笔记本:	C/C++		
创建时间:	2017/8/29 9:03	**更新时间:**	2020/1/1 15:56
作者:	beyourselfwb@163.com		

To use an analogy, if algorithms were about automobiles, it would be for the person who wants to know how cars work, how they are built, and how one might design fuel-efficient, safe, reliable vehicles for the 21st century. The people who hate algorithms are the ones who just want to know how to drive their car on the highway, just like everyone else.

Talk is cheap, show me the code.

总之，算法是一种将有限计算资源发挥到极致的武器，当计算资源很富余时算法确实没大用，但一旦到了效率瓶颈算法绝壁是开山第一刀（因为算法不要钱嘛！要不还得换CPU买SSD升级RAM，肉疼啊！！）。

★其实,算法世界里,要么用时间换空间,要么用空间换时间

时间长、效率低的根本原因：算法低级

证据：1，约瑟夫的环（笨方法—链表，巧妙—数学法）

2，二叉树（笨方法—模拟，巧妙—数学法（奇偶法））

3，给定一个整数n，求n!末尾有多少个0。

算法一： n! = (2^X)+(3^Y)+(5^Z)....质因数分解 Z即为所求 复杂度——n^2

void f(int n){ int ret = 0; for (int i = 1; i <= n; ++i){int j = i;

while (j%5==0){++ret; j /= 5;}} cout << ret << endl;}

算法二： Z = [N/5]+ [N/(5^2)]+ [N/(5^3)]+.... 高斯函数 复杂度——n

void f(int n){int ret = 0;

while (n){ret += (n/5);n /= 5;}cout << ret << endl;}

样例一：2147483647

算法一耗时估计半个小时，算法二不到一秒。

4，生物芯片问题 蓝桥杯2014年B组决赛

5，0/1 翻转题--给定A个0和B个1，每次只能翻转K个数字，最终变成全1，求最少翻转次数。---直接数学推导证明，见笔记（算法：0/1翻转 再次见证数学的力量）

事实证明：学好数学并巧妙利用是掌握高级算法的基础！！！！

评价一个程序的优劣可从以下几个方面考虑：代码长度；运行效率；适用范围； 出

错机率（变量的多少、或使用高风险的函数，如gets或指针）。

零碎

有用的宏定义

```
#define max(a, b) a > b? a: b
#define min(a, b) a < b? a: b
```

1. 参数传递 fun(int a []) 和fun(int *a) 等效
2. 甩掉y or n后面的其他字符（包括回车）

```
While ( getchar()!='y' )
{
。。。。。
While（ getchar() != ‘\n’ ）continue;
}
```

3. **较大的数组（n> 10^5）**要放在**main**外面，否则会异常退出。
4. float a = 1.456 保留2位小数并且四舍五入{ (int) (a*100+0.5)/100 }
5. scanf 后用gets的话，先打getchar(); gets能吃掉’\n’，并把它变成’\0’；
6. 建议把谓词命名为is__xxx，如is__prime（即is it a prime?），返回值0表示假。
7. 编写函数时，应尽量保证它能对任何合法参数都能得到正确的结果，如果不行，应在显著位置标明函数的缺陷，以免误用。
8. **memset** 一个字节一个字节的赋值 需要的头文件 **<string.h>** 或 **<cstring>**
9. 占位符
    1. %d 十进制 如10表示为012(数字零)

2. %o 八进制(小写字母) 前缀%#o
3. %x 十六进制(如为小写，则输出小写；反之，大写) 20表示为0x14或0X14（数字） 前缀 %#x, %#X
4. %e 指数法输出
5. %u unsigned int
6. %hd short int
7. %p或%u、%lu 打印地址

10. 在函数中使用static int tot = 0; 可以避免传参 或 定义全局变量了
11. struct的空间计算：
    1. 结构体空间占用总体上遵循两个原则：
        1. 整体空间是占用空间最大的成员(类型)所占字节的整数倍；
        2. 数据对齐原则----数据在内存中按照结构成员先后顺序进行排序，当排到该成员变量时，其前面已摆放的空间大小必须是该成员类型大小的整倍数，如果不够则补齐，以此向后类推。
    2. struct s1 { char a; double b; int c; char d;}; sizeof(s1) = 24
    3. struct s2 { char a; char b; int c; double d;}; sizeof(s2) = 16
12. 逻辑运算： 且（&&） 或（||） 非（！）
13. exit(0) 正常退出 与main相同
14. 二级指针 行指针
    1. int (*p)[4]; //p 是一个行指针，*p[1] 就是a[1][0]的值
    2. int* pp;
    3. int a[][]; // a: 行指针 a[0]: 普通指针 p = a; pp = a[0]; 如p = a[0]会出错
15. ascxas
16. ascas

实用函数大集锦：

string头文件

- memset(多用来清零) memset(a, 0, sizeof(a));
- memcpy int a[N], b[N];
- memcpy(b, a, sizeof(int)*k) 把a中的前k个元素赋给b数组;
- memcpy(b, a, sizeof(a)) 把a中所有的个元素赋给b数组。
- memcmp(str1,str2,25); 比较字符数组(多维)str1和str2前25个字节的大小关系，返回str1-str2的值，int型。
- strchr(char * str, 'A') 字符A在字符串str中的第一次出现的地址（即返回一个指针）
- strstr(char * str1, char * str2) 字符串str2在字符串str1中的第一次出现的首地址（即返回一个指针）

- strcpy(str1,str2); 字符串拷贝 str1 <- str2

math头文件

- int rand() ; -90 到32767随机数
- double sin (double);正弦
- double cos (double);余弦
- double tan (double);正切
- double sqrt (double);开平方
- double pow(double x, double y）; 计算以x为底数的y次幂
- double ceil (double); 取上整
- double floor (double); 取下整 ,相当于int强制转换
- double fabs (double);求绝对值

stdio头文件

- fgets(str, N, stdin/fin(文件名)) **最后会把回车吃进去（最后一个字符为'\n'）**
- sprintf (s, “%d”, x) 把整数x变成字符串s
- sscanf (s, “%d”, &b) 把字符串s以整数b输出
- fprintf 与之类似

ctype头文件

- isalpha 是否为英文字母
- isdigit
- isupper

- islower
- ispunct 是否为标点符号
- toupper
- tolower

assert头文件

- 作用: 规定参数范围,限制非法的函数调用，“迫使”程序员编出高质量的程序
- int is_prime(int x) {assert (x>=0);} 退出时，会显示 x>=0, 文件名，line 2。
- Assert 与 if 区别:
- assert只在Debug模式下起作用，在release模式下无效,if相反；
- if与客户直接打交道
- assert一般用于你的底层函数层

stdlib头文件

- Double atof (const char * p)
- int atoi (const char * p) // 功能同sscanf()函数
- int system (const char * str) 例如 system(“pause”)

输入结束标志：

  - gets != NULL
- 返回值 ：它收到的字符串的地址
- 表示gets函数没有获得任意数据之前如果遇到文件结尾标志将返回NULL。可以这样测试一下，运行程序，不输入任何字符，直接按ctrl+z。。
  - scanf (“%d%d”, &a, &b)==2
- 返回值：正确按指定格式输入变量的个数；也即能正确接收到值的变量个数;
- 当scanf函数的第n个变量格式不正确时，返回值为n-1.
  - getchar() != EOF
- getchar的返回值为该字符的ASCII码值(int)

- Axasx
- Asxasxsa

调试技巧：

- watch窗口中，在整形变量后面加上",c"可以显示该变量对应的ASCII字符;
- 'v',d就是显示字符'v'对应的十进制ASCII码值是118;
- 'v'，x显示的是对应的十六进制的ASCII码值;
- watch窗口中，把指针当成数组看，只要在指针名后面加上一个长度，就可以想看数组一样看到对应的数据了。比如我上面的a,4。那么如果一个指针指向的数据很大，比如一个整形指针a是指向一个1000个整数的大块内存，我只想看看最后4个数据，要怎样呢？那就 (a+996),4 呗。从第996个数据开始，看4个~;
- 在watch窗口中察看错误原因，只需要在错误之后面频上",hr"就可以了。比如我上面的 dwError,hr 和 2, hr 都能够显示错误消息;
  - 设置高级断点
- 条件
- 执行次数
  - Asdasdad

数据类型及其范围

1. int(32位) -(2^31) ——(2^31-1)

a. 即-2147483648(min)——2147483647(max)
b. 11111111 11111111 11111111 11111111 ------- -1
c. 00000000 00000000 00000000 00000000 ------- 0
d. 01111111 11111111 11111111 11111111 ------- max
e. 10000000 00000000 00000000 00000000 ------- min
f. Min = max + 1
g. -1 + 1 = 0

2. unsigned int(32 位) 0——(2^32-1) 即 0——4294967295 %u 输出
3. float –(2^128) ——(2^128) 即 -3.40*10^38——3.40*10^38

a. float 4字节 共32bits, 1bit(符号)，8bits(指数),23bits(尾数), 精确到6位小数,不安全

4. double –(2^1024) ——(2^1024) 即 -1.79*10^308——1.79*10^308ACASC

a. double 8字节 共64bits, 1bit(符号)，11bits(指数),52bits(尾数), 精确到15位小数

5. long long(unsigned) 8Bytes 2^64-1 = 18446744073709551615(20位) %llu
6. long long 8Bytes 2^63-1 = 9223372036854775807(19位) %lld
7. Scvsdvcdsvv
8. Sccsdv

位运算（6种）

1. &（且） 用法:对指定位进行清零 或 保留。有1的保留 。（来自 http://www.cnblogs.com/911/archive/2008/05/20/1203477.html）
2. |（或） 用法:把指定位变成1。 有1的位就变成1了。
3. ^（异或） 即不相同值为1. 用1来异或，实现指定位的1和0颠倒；用0异或，保留原值；

a. 交换两个变量值（不用临时变量）。

i. a = a∧b;
ii. b = b∧a;
iii. a = a∧b;

4. ~（取反运算） 用法：求反码。
5. <<(左移) 用法：将该数乘以2的n次方（高位不溢出1的前提下）。 注意 long long t = 1 << 40, 不会得到想要的结果! 因为移位运算默认返回的类型是int.
6. >>(右移) 用法：与左移相反。

a. 有符号数：设int 4个字节
b. 最高位(即符号位不变)
c. 负数右移　高位补一　int a = -7(即11111111 11111111 11111111 11111001); a = a >> 2; 结果 a = -2(即11111111 11111111 11111111 11111110)
d. 正数右移　高位补零　int a = 7(即00000000 00000000 00000000 00000111); a = a >> 2;　结果 a = 1 (即00000000 00000000 00000000 00000001)
e. 小用法：　a & = b相当于 a = a & b
f. a << =2相当于a = a << 2
7. asdsafd
8. asdasdc
9. sfewsgfre
10. wsgfref