

搜索

[试用Google AdWords CPC](#)  
为您的市场推广获取投资回报 今天就开始扩展新  
·客户!  
[adwords.google.com](#)

[测试装备拍卖](#)  
出价在测试& 测量装备. 质量设备- 拍卖价格!  
·[www.DoveBid.com](#)

[科日新PCI四轴运动控制卡](#)  
编码器输入,三轴联动,伺服反馈等功能 提供VB ,  
·VC动态库函数010-51650651  
[www.krxgk.com](#)

Google 提供的广告

在本网站刊登广告



标题: NOIP2006普及组解题报告（已补充到第四题）

打印 | 推荐 | 订阅 | 收藏

本主题由 [Liangent](#) 于 11月22日 23:13 加入精华

网络虾客

管理员



发表于 11月19日 13:09 [资料](#) [文集](#) [短消息](#)

#1

NOIP2006普及组解题报告（已补充到第四题）

CODE:

[\[Copy to clipboard\]](#)

第一题：自己看题目。  
思路：很多人想到排序，效率大概是O(n^2)吧，其实还有个更简单的方法。用一个1到1000的数组储存每个数的出现情况，t每出现一次，a[t]=a[t]+1，而且，同时操作 if(a[t]>1) 则总数=总数-1。然后从1到1000的循环，如果a[i]≠0，输出，这样就根本不用排序了。大概这个应该是最优算法吧。O(n)的效率。

参考C代码：

CODE:

[\[Copy to clipboard\]](#)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    int a[1001],n,temp,num,i;/*a是记录数组，相当于数轴，与数一一对应，temp是临时存储数据，n和题目一样，记录不相同的数的个数*/
    FILE *in=fopen("random.in","r");
    FILE *out=fopen("random.out","w");
    fscanf(in,"%d",&n);
    num=n;
    memset(a,0,sizeof(a));/*初始化数组为0，pascal有fillchar*/
    for(i=1;i<=n;i++)
    {
        fscanf(in,"%d",&temp);
        a[temp]++;/*记录是哪个数*/
        if(a[temp]>1)
            num--;
    }
    fprintf(out,"%d\n",num);
    for(i=1;i<=1000;i++)
        if(a[i])/*如果a[i]不为0。这样就根本不用排序*/
            fprintf(out,"%d ",i);
    fclose(in);
    fclose(out);
    return 0;
}
```

CODE:

[\[Copy to clipboard\]](#)

第二题：  
和采药一样。但是如果完全照搬在WINXP下不能运行，CENA测的是栈溢出，TC甚至不让编译通过，说too big size（好像是系统限制，当时我就是完全照搬的，还好评测时用的是LINUX，可以运行。）所以要进行优化。回滚，因为这题比较特殊，所以可以用回滚进行优化，将26\*30000压缩到3\*30000甚至2\*30000（我建议前者，保险）。所谓回滚就是到这一个状态储存在前一个状态的空间里。一个 mod 2+1就给解决了。这样就避免了空间的浪费。  
动态转移方程 ,i表示状态，也就是取第几个物品，j表示总钱数  
f[i][j]=max(f[i-1][j-money[i]]+v[i]\*p[i],f[i-1][j])

下面为了大家清楚，我把优化前和优化后的代码贴出来  
优化前：

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    FILE *in=fopen("happy.in","r");
    FILE *out=fopen("happy.out","w");
    long ans[26][30001];/*状态数组*/
    int n,m,v,p;
    long money[26],value[26];
    long i,j;
    int max(int x,int y);
    fscanf(in,"%ld %ld",&n,&m);
    memset(ans,0,sizeof(ans));
    for(i=1;i<=m;i++)
    {
        fscanf(in,"%ld %ld",&v,&p);
        money[i]=v;
        value[i]=v*p;
    }
    fclose(in);
    for(i=0;i<=m;i++)/*边界状态*/
        ans[i][0]=0;
    for(i=0;i<=n;i++)/*边界状态*/
        ans[0][i]=0;
    for(i=1;i<=m;i++)/*寻求最佳方案*/
        for(j=1;j<=n;j++)
            if(j>=money[i])
                ans[i][j]=max(ans[i-1][j-money[i]]+value[i],ans[i-1][j]);
            else ans[i][j]=ans[i-1][j];
    fprintf(out,"%ld\n",ans[m][n]);
    fclose(out);
    return 0;
}
int max(int x,int y)
{
    if(x>y)    return x;
    else return y;
}
```

优化后

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    FILE *in=fopen("happy.in","r");
    FILE *out=fopen("happy.out","w");
    long ans[3][30001];
    int n,m,v,p;
    long money[26],value[26];
    long i,j;
    int max(int x,int y);
    fscanf(in,"%ld %ld",&n,&m);
    memset(ans,0,sizeof(ans));
    for(i=1;i<=m;i++)
    {
        fscanf(in,"%ld %ld",&v,&p);
        money[i]=v;
        value[i]=v*p;
    }
    fclose(in);
    for(i=0;i<=3;i++)
        ans[i][0]=0;
    for(i=0;i<=n;i++)
        ans[0][i]=0;
    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
            if(j>=money[i])
                {if(i%2==1)
                    ans[i%2][j]=max(ans[i%2-1][j-money[i]]+value[i],ans[i%2-1][j]);
                else ans[i%2][j]=max(ans[i%2+1][j-money[i]]+value[i],ans[i%2+1][j]);}
            else {
                if(i%2==1)
                    ans[i%2][j]=ans[i%2-1][j];
                else ans[i%2][j]=ans[i%2+1][j];
            }
    fprintf(out,"%ld\n",ans[m%2][n]);
    fclose(out);
    return 0;
}
```

```
int max(int x,int y)
{
    if(x>y)    return x;
    else return y;
}
```

CODE:

[\[Copy to clipboard\]](#)

第三题：  
没想好，同学用C++写出来了，还是贴代码给大家看吧，请参看代码中的注释。晚一些我会把C的代码写出来。

CODE:

[\[Copy to clipboard\]](#)

```
#include <fstream>
#include <string>

using namespace std;

class JamNum
{ //JamNum类， 记录Jam数字及和该数字相关的属性
public:
    char num[27]; //以C-style字符串记录该数字内容
    char begin; //字符集开始字符
    char end; //字符集结束字符
    int length; //该Jam数字长度

    JamNum(char n[], char bg, char e, int len)
    { //构造函数， 参数为： (C-style字符串表示的内容, 字符集开始字符, 字符集结束字符, 该Jam数字长度)
        strcpy(num, n);
        begin = bg;
        end = e;
        length = len;
    }

    JamNum* getNext()
    { //程序最主要部分， 获取下一个Jam数字， 返回下一个Jam数字的指针
        char next[27]; //记录下一个数字的内容
        int i, j;
        strcpy(next, num);
        if (next[length-1] < end)
        { //如果最后一位没有到字符集的最后一个字符， 那么把最后一位+1， 创建新对象， 返回指针
            next[length-1]++;
            return new JamNum(next, begin, end, length);
        }

        for (i = length - 2; i >=0; i--) //否则往前扫描每一位
            if (next[i] < end - (length - i - 1))
            { //如果该位没有到该位可以出现的最大字符（字符集结束字符 - (长度 - 从左到右的位数 - 1)，
              //即保证最后一位不会超出字符集结束字符）， 那么将该位+1， 后面每位比前一位依次+1
              //（因为后面每位都比前面一位大）， 创建新对象， 返回指针
                next[i]++;
                for (j = i + 1; j < length; j++)
                    next[j] = next[j-1] + 1;
                return new JamNum(next, begin, end, length);
            }

        //如果没有任何一位能满足没有到该位可以出现的最大字符， 即已到最后一个数， 返回length属性为-1的Jam数字对象指针
        return new JamNum("", -1, -1, -1);
    }
};

int main()
{
    ifstream fin("count.in");
    ofstream fout("count.out");

    int bgi, endi, len;
    char bg, end;
    char innum[27];
    JamNum *n;
    int i;

    //读入数据并创建开头的Jam数字对象
    fin >> bgi >> endi >> len;
    bg = bgi + 'a' - 1;
    end = endi + 'a' - 1;
    fin >> innum;
    n = new JamNum(innum, bg, end, len);

    for (i = 0; i < 5; i++)
    { //重复5次获取下一个， 输出。因为最多只会有6个对象， 就不delete了
```

```
        n = n->getNext();
        if (n->length == -1) //如果没有下一个，即新对象的length属性为-1
            break;
        fout << n->num << endl;
    }

    fin.close();
    fout.close();

    return 0;
}
```

CODE:

[Copy to clipboard]

第四题：

很简单的递推（可恶我当时怎么没想到！）设a[1000]为这1000项的集合，a[n]表示第n个数（忽略下标0，从1开始）。第i个完全平方数（就是没有相加，只是一个数的n次方的位置，以1为起始）的位置是：2^(i-1)，（^是乘方符号）。那么第i+1（i同前面一样）个数就等于(2^(i-1)+a[1])，i+2个数就是2^(i-1)+a[2]，依次类推。把前1000项全部求出来。然后输出第n项就可以了。这题比第二题还简单，可我竟然没读懂这题目！由此可见，难度并不是由易到难的。

参考源程序：

CODE:

[Copy to clipboard]

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *in=fopen("sequence.in","r") ;
    FILE *out=fopen("sequence.out","w");
    int k,n,i,j,l,t;
    long a[2000];
    long cf(int x,int y);/*求乘方的函数*/
    fscanf(in,"%ld %ld",&k,&n);
    for(i=0;i<=1050;i++)
        a[i]=0;
    for(i=0;i<=10;i++)
    {
        t=cf(2,i);
        a[t]=cf(k,i);
    }

    for(i=1,j=1;i<=1000;i++)
        if(a[i]!=0)
        {
            l=a[i];
            j=1;
        }
        else if(a[i]==0)
            a[i]=a[j++]+l;
    fprintf(out,"%ld\n",a[n]);
    fclose(in);
    fclose(out);
    return 0;
}


long cf(int x,int y)/*乘方函数*/
{
    int i;
    long q=1;
    if(y==0)
        return 1;
    for(i=1,q=1;i<=y;i++)
        q=q*x;
    return q;
}
```

[引用](#) [回复](#)

[广告] 网上冲浪更快更爽，下载带有 Google 工具栏的 Firefox


<b>游客</b> 125.73.57.x 未注册	发表于 11月19日 15:40	#2
	其它题目呢？ ？ ？ ？ ？	
		<a href="#">编辑</a> <a href="#">引用</a> <a href="#">回复</a>
<b>游客</b> 220.179.206.x 未注册	发表于 11月19日 18:02	#3

那就是排序，桶排序。

编辑 引用 回复 

**victorlee129**

新手上路



UID 5

精华 0

积分 0

帖子 2

阅读权限 10

注册 11月19日


状态 离线

发表于 11月19日 21:18 [资料](#) [短消息](#)

#4

到底其他的呢？

到底其他的呢，阿？！

引用 回复 

**游客** 218.71.229.x

未注册

发表于 11月21日 16:48

#5

happy

**CODE:**

[\[Copy to clipboard\]](#)

```
#include <stdio>
#include <cstring>

int Money, Goods, Ans;
int M[25], C[25];
int Opt[30001];


int main()
{
    freopen("happy.in", "r", stdin);
    freopen("happy.out", "w", stdout);

    scanf("%d%d", &Money, &Goods);
    for (int I = 0; I < Goods; I++)
    {
        scanf("%d%d", &M[I], &C[I]);
        C[I] *= M[I];
    }

    memset(Opt, -1, sizeof(Opt));
    Opt[0] = 0;
    for (int I = 0; I < Goods; I++)
        for (int J = Money; J >= 0; J--)
            if (J + M[I] <= Money)
                if (Opt[J] != -1 && Opt[J] + C[I] > Opt[J + M[I]])
                    Opt[J + M[I]] = Opt[J] + C[I];

    Ans = 0;
    for (int I = 0; I < Money; I++)
        if (Opt[I] > Ans)
            Ans = Opt[I];

    printf("%d\n", Ans);
    return 0;
}
```

编辑 引用 回复 

**游客** 218.71.229.x

未注册

发表于 11月21日 16:48

#6

counting

**CODE:**

[\[Copy to clipboard\]](#)

```
#include <stdio>

int Start, Target, Size;
char S[27];

int Inc()
{
    int I = Size - 1;
    while (I < 0 || S[I] == (I + 97 + Target - Size))
    {
        I--;
    }
    if (I < 0)
        return 0;
    S[I++]++;
    for (; I < Size; I++)
```

游客218.71.229.x

未注册

发表于 11月21日 16:49

#7

sequence

CODE:

[Copy to clipboard]

```
#include <stdio>

int main()
{
    freopen("counting.in", "r", stdin);
    freopen("counting.out", "w", stdout);

    scanf("%d%d%d\n%s", &Start, &Target, &Size, &S);

    for (int I = 0; I < 5; I++)
    {
        if (Inc())
            printf("%s\n", S);
        else
            break;
    }
    return 0;
}
```

编辑

引用

回复

TOP

游客125.73.58.x

未注册

发表于 11月24日 17:48

#8

QUOTE:

原帖由 *Guest* from 220.179.206.x 于 11月19日 18:02 发表  
那就是排序，桶排序。

长见识了。。  
原来这就叫传说中的桶排序。谢谢~

编辑

引用

回复

TOP

czhxdong

新手上路

★

UID 8

精华 0

积分 0

帖子 5

阅读权限 10

注册 11月26日

状态 离线

发表于 11月26日 16:19 [资料](#) [短消息](#)

#9

申请当NOIP版主！！

我是一个编程爱好者，我对程序设计语言有一种说不出的热情！我的目标是开发出比Windows更优秀的操作系统！  
我现在的目标是通过信息学奥林匹克保送上名校深造！！  
我觉得OI人网是一个我见过的最好的NOI/IOI网站，我一定多为你宣传一下，提高OI网的点击！！  
望站长批准！！  
申请人：张旭东

引用

回复


TOP

游客222.209.223.x

未注册

发表于 11月27日 17:09

#10



[编辑](#) [引用](#) [回复](#) [TOP](#)

 [新帖](#)

 [回复](#)

快速回复主题

选项

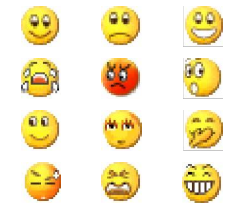
标题

Smilies

- ☐ 禁用 URL 识别
- ☐ 禁用 Smilies
- ☐ 禁用 Discuz! 代码
- ☐ 使用个人签名
- ☐ 接收新回复邮件通知

☐

内容



[更多 Smilies >>](#)

发表帖子

[完成后可按 Ctrl+Enter 发布]

预览帖子

清空内容



当前时区 GMT+8, 现在是 12月3日 16:18  
[桂ICP备06014442号](#)



Powered by **Discuz! 5.0.0** © 2001-2006 Comsenz Inc.  
Processed in 0.066677 second(s), 8 queries

[TOP](#)

[清除 Cookies](#) - [联系我们](#) - [OI人](#) - [Archiver](#) - [WAP](#)