

第8章 算法设计初体验

第8.3课 减治法

《信息学奥赛一本通·编程启蒙 C++版》

减治法，即减而治之（Decrease-and-conquer）为求解一个大规模的问题，可以将其划分为两个子问题：其一平凡，另一规模缩减，分别求其子问题，由子问题的解，得到原问题的解。

一、基本思路

〈1〉将原问题的实例转化为规模更小的实例

〈2〉对规模较小的实例求解

〈3〉将较小的实例的解扩展到原实例

一旦建立原问题给定实例的解和问题规模更小的实例解的某种关系，我们就能够自顶向下或者自底向上地使用这种关系。自顶向下可以用递归方法实现；自底向上可以用迭代实现，这种实现方式经常被称为增量法。

二、减治法的变化形式

〈1〉减去一个常量（一般是 1）

〈2〉减去一个常量因子（一般是一半）

〈3〉减可变规模

【例 83.1】 奥利凡德

【题目描述】

哈利波特在与伏地魔的战斗中毁坏了自己的魔杖，于是他决定去奥利凡德的魔杖店买个新的。他在店里看到 n 个魔杖和 n 个盒子，每个魔杖的长度为 $x_1, x_2, x_3, \dots, x_n$ ，每个盒子的长度为 $y_1, y_2, y_3, \dots, y_n$ 。一个长度为 X 的魔杖能放进长度为 Y 的盒子里只有满足 $X \leq Y$ 。

哈利想知道他能否把所有魔杖都放进盒子里，并且每个盒子只能放一根魔杖请你帮他解决这个问题。

【输入格式】

第一行一个整数 n ($1 \leq n \leq 100$)，表示魔杖的数量。

第二行 n 个整数 $1 \leq x_i \leq 100$ 表示每根魔杖的长度。

第三行 n 个整数 $1 \leq y_i \leq 100$ 表示每个盒子的长度。

【输出格式】

如果哈利能把所有魔杖放进盒子里，输出“DA”，否则输出“NE”。（克罗地亚语的“yes”和“no”）。

【样例输入】

```
3
7 9 5
6 13 10
```

【样例输出】

```
DA
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,a[105],b[105];
4. //插入排序
5. void insertion_sort(int A[]){
6.     int temp,i,j;
7.     for(i=1;i<n;i++)
8.         if(A[i]<A[i-1]){
9.             temp=A[i];
10.            for(i=i-1;i>=0&&A[j]>temp;--j)A[j+1]=A[j];
11.            A[j+1]=temp;
12.        }
13.}
14. //选择排序
15. void selection_sort(int A[]){
16.     for(int i = 0; i < n-1; ++i){
17.         int min_id = i;
18.         for(int j = i; j < n; ++j)
19.             if (A[j] < A[min_id]) min_id=j;
20.         if(min_id != i) swap(A[i],A[min_id]);
21.     }
22.}
```

```
23. //冒泡排序
24. void Bubble_sort(int A[]){
25.     for(int i = 1; i < n; ++i)
26.         for(int j = i; j >=0; j--){
27.             if(A[j]<A[j-1]) swap(A[j],A[j-1]);
28.         }
29.}
30. int main(){
31.     cin>>n;
32.     for(int i=0;i<n;i++) cin>>a[i];
33.     for(int i=0;i<n;i++) cin>>b[i];
34.     insertion_sort(a);
35.     selection_sort(b); //Bubble_sort(b);
36.     for(int i=0;i<n;i++){
37.         if(a[i]>b[i]){
38.             cout<<"NE";
39.             return 0;
40.         }
41.     }
42.     cout<<"DA";
43.     return 0;
44.}
```

【例 83.2】二分查找

【题目描述】

对有序数组进行二分查找，是一种性能卓越的算法：目标是在有序数组 $a[]$ 中查找是否有 $a[k]=key$ ，若有返回 k ，否则返回 -1 。

* 若有多个 $a[k]=key$ ，请输出最小的 k

* 下标 k 从 1 开始计数

【输入格式】

第一行包含 1 个整数 N ，代表数组长度

第二行包含 N 个 ($N \leq 30000$) int 范围内的整数，保证升序，空格隔开。

第三行包含 1 个整数 T ，代表有 T 组询问

第四行包含 T 个 ($T \leq 30000$) int 范围内的整数，表示每组询问需要查找的 key 值，空格隔开。

【输出格式】

一行，对于每组询问输出结果 (k or -1)，空格隔开。

【样例输入】

```
3
-2 -2 1
3
-2 -2 -1
```

【样例输出】

```
1 1 -1
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,a[30005],t,x,l,r,mid;
4. int main()
5. {
6.     scanf("%d",&n);
7.     for(int i=1;i<=n;i++) scanf("%d",&a[i]);
8.     scanf("%d",&t);
9.     for(int i=1;i<=t;i++) {
10.         scanf("%d",&x);
11.         l=0,r=n;//半开半闭区间
12.         while(r-l>1){
13.             mid=(l+r)/2;
14.             if(a[mid]>=x) r=mid;
15.             else l=mid;
16.         }
17.         if(a[r]==x) cout<<r<<' ';
18.         else cout<<-1<<' ';
19.     }
20.     return 0;
21. }
```

【例 83.3】假币问题

【题目描述】

在 n 枚外观相同的硬币中寻找一枚假币，有一架没有刻度的天平但是能够显示两边的重量是否相等，如果相等，天平就不会倾斜，如果不相等，重的一边就会倾斜。设计一个有效的算法来找出这枚假币。假设这枚假币比真币要轻。

【输入格式】

输入有多行，每一行的值为硬币的数目 n ， $1 \leq n \leq 2303$ ，输入 0 程序结束。

【输出格式】

最少要称多少次一定能把那个假币找出来。输出对应输入行数。

【样例输入】

16

0

【样例输出】

3

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int fact(int n)
4. {
5.     int count=0;
6.     if(n==1) return 0;
7.     if(n==2||n==3) return 1;
8.     while(n>3){
9.         if(n%3==0) n/=3;
10.        else n=(n/3)+1;
11.        count++;
12.    }
13.    return count+1;
14.}
15.int main()
16.{
17.    int n;
18.    while(scanf("%d",&n)&&n!=0){
19.        printf("%d\n",fact(n));
20.    }
21.    return 0;
22.}
```

【例 83.4】最大公约数和最小公倍数

【题目描述】

输入两个正整数 x_0, y_0 ，求出满足下列条件的 P, Q 的个数：

1. P, Q 是正整数。
2. 要求 P, Q 以 x_0 为最大公约数，以 y_0 为最小公倍数。

试求：满足条件的所有可能的 P, Q 的个数。

【输入格式】

一行两个正整数 x_0, y_0 。

【输出格式】

一行一个数，表示求出满足条件的 P, Q 的个数。

【样例输入】

3 60

【样例输出】

4

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. typedef long long ll;
4. ll m,n,ans,flag;
5. ll gcd(ll x,ll y)
6. {
7.     if(y==0) {return x;}
8.     return gcd(y,x%y);
9. }
10.int main()
11.{
12.    cin>>n>>m;
13.    for(ll p=1;p<=sqrt(m*n);p++)
14.        if((n*m)%p==0&&gcd(p,n*m/p)==n) //Q=n*m/p
15.            ans+=2;
16.    if(m==n) ans--;
17.    cout<<ans;
18.    return 0;
19.}
```

练 83.1 压缩歌曲

【题目描述】

小明的电脑里面有一些歌。现在他需要把这些歌装进一个硬盘里面。

硬盘大小有限，由于小明下载的都是无损版本，每首歌的占用空间比较大，硬盘不一定装得下，然后他需要压缩其中的一部分歌曲这样他才能将尽可能多的歌曲装进他的硬盘里。

但是他想尽量压缩的歌曲数量尽量少，他不知道该怎么办，就来找你帮忙了。

【输入格式】

输入的第一行包含两个整数 n 和 m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^9$), 分别表示小明电脑里面歌曲的个数和他的硬盘大小 (单位: 字节)。

然后输入 n 行，每一行两个整数 a_i 和 b_i ($1 \leq b_i < a_i \leq 10^9$), 分别表示第 i 首歌曲原本的大小和被压缩后的大小 (单位: 字节)。

【输出格式】

输出只有一个整数，小明至少需要压缩的歌曲的数量。

如果所有的歌曲都压缩的硬盘还是装不下，输出 -1

【样例输入】

4 21

10 8

7 4

3 1

5 4

【样例输出】

2

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. long long n,sum,a[100005],x,y,s=0,o=0;
4. int main(){
5.     cin>>n>>sum;
6.     for(int i=0;i<n;i++){
7.         scanf("%lld%lld",&x,&y);
8.         a[i]=x-y;
9.         s+=x;
10.    }
11.    sort(a,a+n);
12.    if(s<=sum){
13.        cout<<0;
14.        return 0;
15.    }
16.    o++;
```

```
17.    for(int i=n-1;i>=0;i--){
18.        if(s-a[i]<=sum){
19.            cout<<o;
20.            return 0;
21.        }else{
22.            s-=a[i];
23.            o++;
24.        }
25.    }
26.    cout<<-1;
27.    return 0;
28.}
```

练 83.2 汽水瓶

【题目描述】

有这样一道智力题：“某商店规定：三个空汽水瓶可以换一瓶汽水。小张手上有十个空汽水瓶，她最多可以换多少瓶汽水喝？”答案是 5 瓶，方法如下：先用 9 个空瓶子换 3 瓶汽水，喝掉 3 瓶满的，喝完以后 4 个空瓶子，用 3 个再换一瓶，喝掉这瓶满的，这时候剩 2 个空瓶子。然后你让老板先借给你一瓶汽水，喝掉这瓶满的，喝完以后用 3 个空瓶子换一瓶满的还给老板。如果小张手上有 n 个空汽水瓶，最多可以换多少瓶汽水喝？

【输入格式】

每个数据占一行，仅包含一个正整数 $n(1 \leq n \leq 100)$ ，表示小张手上的空汽水瓶数。 $n=0$ 表示输入结束，你的程序不应当处理这一行。

【输出格式】

对于每组测试数据，输出一行，表示最多可以喝的汽水瓶数。如果一瓶也喝不到，输出 0。

【样例输入】

```
3
10
81
0
```

【样例输出】

```
1
5
40
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. int f(int x){
5.     if(x<2)return 0;
6.     if(x==2)return 1;
7.     return x/3+f(x/3+x%3);
8. }
9. int main(){
10.    while(cin>>n){
11.        if(n==0)break;
12.        cout<<f(n)<<endl;
13.    }
14.    return 0;
15.}
```

练 83.3 特殊的集合

【题目描述】

集合 $A=\{1,2,\dots,n\}$, 集合 B 为 A 子集, 对于 B 中任意一个元素 x , $2x$ 均不在集合 B 中。

求集合 B 最多能有多少个元素。

【输入格式】

一个整数 $n(1 \leq n \leq 10^9)$

【输出格式】

以一个整数, 表示集合 B 最多的元素个数。

【样例输入】

100

【样例输出】

67

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. int f(int x){
5.     if(x==0) return 0;
6.     int mid=x/2;
7.     return f(mid/2)+x-mid;
8. }
9. int main(){
10.     cin>>n;
11.     cout<<f(n);
12.     return 0;
13.}
```

练 83.4 分解因数

【题目描述】

给出一个正整数 a ，要求分解成若干个正整数的乘积，即 $a = a_1 \times a_2 \times a_3 \times \dots \times a_n$ ，并且 $1 < a_1 \leq a_2 \leq a_3 \leq \dots \leq a_n$ ，问这样的分解的方案种数有多少。注意到 $a = a$ 也是一种分解。

【输入格式】

第 1 行是测试数据的组数 n ($1 \leq n \leq 10$)，后面跟着 n 行输入。每组测试数据占 1 行，包括一个正整数 a ($1 < a < 32768$)。

【输出格式】

n 行，每行输出对应一个输入。输出应是一个正整数，指明满足要求的分解的方案种数。

【样例输入】

2
2
20

【样例输出】

1
4

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int t,n,ans;
4. int f(int x,int y){
5.     for(int i=y;i<=x/i;i++)
6.         if(x%i==0){
7.             ans++;
8.             f(x/i,i);
9.         }
10.    return ans;
11.}
12.int main(){
13.    scanf("%d",&t);
14.    while(t--){
15.        ans=0;
16.        scanf("%d",&n);
17.        printf("%d\n",f(n,2)+1);
18.    }
19.    return 0;
20.}
```

练 83.5 二分查找 2

【题目描述】

对有序数组进行二分查找，是一种性能卓越的算法：目标是在有序数组 $a[]$ 中查找是否有 $a[k]=key$ ，若有返回 k ，否则返回 -1 。

* 若有多个 $a[k]=key$ ，请输出最大的 k

* 下标 k 从 1 开始计数

【输入格式】

第一行包含 1 个整数 N ，代表数组长度

第二行包含 N 个 ($N \leq 30000$) int 范围内的整数，保证升序，空格隔开。

第三行包含 1 个整数 T ，代表有 T 组询问

第四行包含 T 个 ($T \leq 30000$) int 范围内的整数，表示每组询问需要查找的 key 值，空格隔开。

【输出格式】

一行，对于每组询问输出结果 (k or -1)，空格隔开。

【样例输入】

```
4
-2 -2 -2 1
3
-2 -2 -1
```

【样例输出】

```
3 3 -1
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,a[30005],t,x,l,r,mid;
4. int main()
5. {
6.     scanf("%d",&n);
7.     for(int i=1;i<=n;i++) scanf("%d",&a[i]);
8.     scanf("%d",&t);
9.     for(int i=1;i<=t;i++) {
10.         scanf("%d",&x);
11.         l=1,r=n+1; // 半开半闭区间
12.         while(r-l>1){
13.             mid=(l+r)/2;
14.             if(a[mid]<=x) l=mid;
15.             else r=mid;
16.         }
17.         if(a[l]==x) cout<<l<<' ';
18.         else cout<<-1<<' ';
19.     }
20.     return 0;
21. }
```

谢谢！