

第8章 算法设计初体验

第85课 贪婪技术

《信息学奥赛一本通·编程启蒙 C++版》

所谓贪心（贪婪）算法就是在对问题求解时，总是做出当前看来是最好的选择。也就是说贪心算法并不从整体最优上考虑问题，算法得到的是某种意义上的局部最优解。而局部的最优解叠加在一起便构成了问题的整体最优解，或者近似最优解。正是因为贪心算法思想简单，且算法效率较高，所以在一些问题的解决上有着明显的优势。

虽然贪心算法简单且易于实现，但是理论上在使用贪心算法解决问题之前，要对问题本身进行深入透彻的分析和证明，以确保使用贪心算法可以得到整体最优解。

【例 85.1】 金银岛

【题目描述】

金银岛上有 s 种金属, 每种金属重量不同, 分别为 n_1, n_2, \dots, n_s , 同时每种金属的价值也不同, 分别为 v_1, v_2, \dots, v_s 。一次带走的金属重量总和不能超过 w , 问最多能带走价值多少的金属?

注意: 金属是可以被任意分割的, 金属的价值和重量成正比。

【输入格式】

第 1 行是测试数据的组数 k , 后面跟着 k 组输入。

每组测试数据占 3 行, 第 1 行是一个正整数 $w (1 < w \leq 10000)$, 表示口袋承重上限。第 2 行是一个正整数 $s (1 < s < 100)$, 表示金属种类。第 3 行有 $2s$ 个正整数, 分别为 $n_1, v_1, n_2, v_2, \dots, n_s, v_s$, 分别为第一种, 第二种, \dots 第 s 种金属的总重量和总价值 ($1 \leq n_i \leq 10000, 1 \leq v_i \leq 10000$)。

【输出格式】

k 行, 每行输出对应一个输入。输出应精确到小数点后 2 位。

【样例输入】

```
2
50
4
10 100 50 30 7 34 87 100
10000
5
1 43 43 323 35 45 43 54 87 43
```

【样例输出】

```
171.93
508.00
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. struct coin{
4.     int m,v;
5. }a[105];
6. int k,w,s,c;
7. double ans;
8. bool cmp(coin x,coin y){
9.     return x.v*y.m>y.v*x.m;
10.}
11.void solve(){
12.    ans=0;
13.    cin>>w>>s;
14.    c=w;
15.    for(int j=1;j<=s;j++){
16.        cin>>a[j].m>>a[j].v;
17.    }
18.    sort(a+1,a+s+1,cmp);
19.    int cnt=0;
20.    bool flag=false;
```

```
21.    for(int j=1;j<=s;j++){
22.        if(a[j].m>c){
23.            cnt=j;
24.            flag=true;
25.            break;
26.        }
27.        c-=a[j].m;
28.        ans+=a[j].v;
29.    }
30.    if(flag){
31.        ans+=1.0*c*a[cnt].v/a[cnt].m;
32.    }
33.    printf("%.2lf\n",ans);
34.}
35.int main(){
36.    cin>>k;
37.    for(int i=1;i<=k;i++){
38.        solve();
39.    }
40.    return 0;
41.}
```

【例 85.2】 区间调度问题

【题目描述】

学校在最近几天有 n 个活动，这些活动都需要使用学校的大礼堂，在同一时间，礼堂只能被一个活动使用。由于有些活动时间上有冲突，学校办公室人员只好让一些活动放弃使用礼堂而使用其他教室。

现在给出 n 个活动使用礼堂的起始时间 $begin_i$ 和结束时间 end_i ($begin_i < end_i$)，请你帮助办公室人员安排一些活动来使用礼堂，要求安排的活动尽量多。

【输入格式】

第行一个整数 n ($n \leq 1000$)；

接下来的 n 行，每行两个整数，第一个 $begin_i$ ，第二个是 end_i ($begin_i < end_i \leq 32767$)。

【输出格式】

输出最多能安排的活动个数。

【样例输入】

```
11
3 5
1 4
12 14
8 12
0 6
8 11
6 10
5 7
3 8
5 9
2 13
```

【样例输出】

```
4
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. struct node{
4.     int start,end;
5.     bool operator< (node x){
6.         return end<x.end;
7.     }
8. }a[1005];
9. int n,ans,finsh;
10.int main(){
11.     cin>>n;
12.     for(int i=1;i<=n;i++){
13.         cin>>a[i].start>>a[i].end;
14.     }
15.     sort(a+1,a+n+1);
16.     for(int i=1;i<=n;i++){
17.         if(finsh<=a[i].start){
18.             ans++;
19.             finsh=a[i].end;
20.         }
21.     }
22.     cout<<ans;
23.     return 0;
24.}
```

【例 85.3】 过河问题

【题目描述】

有 n 个人一条船，所有人要从左岸渡船到右岸，第 i 个人过河时间是 a_i 。船上最多坐两个人，船从右岸回到左岸时必须保证船上有人。每次船的行驶时间等于船上两人中过河时间的较大值。

问过河所需最短时间。

【输入格式】

输入 t 组数据，每组数据第 1 行输入 n ，第 2 行输入 n 个数，表示每个人过河的时间。

【输出格式】

输出 t 行数据，每行 1 个数，表示每组过河最少时间。

【样例输入】

1

4

1 2 5 10

【样例输出】

17

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n, T;
4. int a[1010];
5. int main(){
6.     cin>>T;
7.     while(T-->0)
8.     {
9.         cin >> n;
10.        for (int i = 1; i <= n; i ++> cin >> a[i];
11.        sort(a + 1, a + 1 + n);
12.        int ans = 0;
13.        while(n > 3){
14.            int sum1 = a[1] * 2 + a[n] + a[n - 1];
15.            int sum2 = a[2] * 2 + a[1] + a[n];
16.            ans += min(sum1, sum2);
17.            n -= 2;
18.        }
19.        if(n == 1) ans += a[1];
20.        else if(n == 2) ans += a[2];
21.        else ans += a[1] + a[2] + a[3];
22.        cout << ans << endl;
23.    }
24.    return 0;
25.}
```

练 85.1 [NOIP2007 普及组] 纪念品分组

【题目描述】

元旦快到了，校学生会让乐乐负责新年晚会的纪念品发放工作。为使得参加晚会的同学所获得的纪念品价值相对均衡，他要把购来的纪念品根据价格进行分组，但每组最多只能包括两件纪念品，并且每组纪念品的价格之和不能超过一个给定的整数。为了保证在尽量短的时间内发完所有纪念品，乐乐希望分组的数目最少。

你的任务是写一个程序，找出所有分组方案中分组数最少的一种，输出最少的分组数目。

【输入格式】

共 $n+2$ 行：

第一行包括一个整数 w ，为每组纪念品价格之和的上限。

第二行为一个整数 n ，表示购来的纪念品的总件数 G 。

第 $3\sim n+2$ 行每行包含一个正整数 P 表示所对应纪念品的价格。

50%的数据满足： $1\leq n\leq 15$ 。

100%的数据满足： $1<n\leq 3\times 10^4, 80<w\leq 200, 5\leq P_i\leq w$ 。

【输出格式】

一个整数，即最少的分组数目。

【样例输入】

100

9

90

20

20

30

50

60

70

80

90

【样例输出】

6

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int W,ans=0;
4. int n,a[30001];
5. int l,r,i;
6. int main()
7. {
8.     scanf("%d%d",&W,&n);
9.     for(i=1;i<=n;i++)
10.         scanf("%d",&a[i]);
11.     sort(a+1,a+n+1);
12.     l=1; r=n;
13.     while(l<=r)//一定要有等号。
14.     {
15.         if(a[l]+a[r]<=W) //一定要有等号。
16.             l++,r--,ans++;
17.         else
18.             r--,ans++; //贪心过程
19.     }
20.     printf("%d",ans);
21.     return 0;
22. }
```

练 85.2 排队接水

【题目描述】

有 n 个人在一个水龙头前排队接水，假如每个人接水的时间为 T_i ，请编程找出这 n 个人排队的一种顺序，使得 n 个人的平均等待时间最小。

【输入格式】

共两行，第一行为 $n(1 \leq n \leq 1000)$ ；第二行分别表示第 1 个人到第 n 个人每人的接水时间 T_1, T_2, \dots, T_n ，每个数据之间有 1 个空格。保证 T_i 互不相同。

【输出格式】

有两行，第一行为一种排队顺序，即 1 到 n 的一种排列；第二行为这种排列方案下的平均等待时间(输出结果精确到小数点后两位)。

【样例输入】

```
10  
56 12 1 99 1000 234 33 55 99 812
```

【样例输出】

```
3 2 7 8 1 4 9 6 10 5  
291.90
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. struct node{
4.     int b,num;
5. };
6. node a[1010];
7. int n,i,j;
8. double ans;
9. bool cmp(node x,node y){
10.     return x.b<y.b;
11.}
12.int main(){
13.     cin>>n;
```

```
14.     for(i=1;i<=n;i++){
15.         cin>>a[i].b;
16.         a[i].num=i;
17.     }
18.     sort(a+1,a+n+1,cmp);
19.     for(i=1;i<=n;i++) cout<<a[i].num<<" ";
20.     cout<<endl;
21.     for(j=n-1;j>=1;j--){
22.         i=n-j;
23.         ans+=a[i].b*j;
24.     }
25.     printf("%.2lf",ans/n);
26.     return 0;
27.}
```

练 85.3 删数问题 (Noip1994)

【题目描述】

输入一个高精度的正整数 n ，去掉其中任意 s 个数字后剩下的数字按原左右次序组成一个新的正整数。编程对给定的 n 和 s ，寻找一种方案使得剩下的数字组成的新数最小。

输出新的正整数。（ n 不超过 240 位）

输入数据均不需判错。

【输入格式】

共两行，第一行为 n ；第二行 s 。

【输出格式】

最后剩下的最小数

【样例输入】

175438

4

【样例输出】

13

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. string s;
4. int k,len;
5. int main(){
6.     cin>>s>>k;
7.     len=s.size();
8.     for(int i=1;i<=k;i++){
9.         for(int j=0;j<len;j++){
10.            if(s[j]>s[j+1]){
11.                for(int l=j;l<len;l++)
12.                    s[l]=s[l+1];
13.                break;
14.            }
15.        }
16.        len--;
17.    }
18.    int x=0;
19.    while(x<len-1 && s[x]=='0') x++;
20.    for(int i=x;i<len;i++) cout<<s[i];
21.    return 0;
22.}
```

谢谢！

—