

# P1926 小书童——刷题大军

## 题目背景

数学是火，点亮物理的灯；物理是灯，照亮化学的路；化学是路，通向生物的坑；生物是坑，埋葬学理的人。文言是火，点亮历史宫灯；历史是灯，照亮社会之路；社会是路，通向哲学大坑；哲学是坑，埋葬文科生。——小A

## 题目描述

小A“刷题”十分猖狂，明目张胆地“刷题”。他现在在小书童里发现了 $n$ 样他喜欢的“题目”，每“题”都有他的需要时间，而老师布置了 $m$ 项作业，每项作业都有它的需要时间及分值，老师规定 $k$ 分以上算及格。小A只剩 $r$ 个单位时间，他想要在及格的基础上更多地“刷题”。

## 输入输出格式

输入格式：

第一行： $n\ m\ k\ r$ 。第二行： $n$ 个数，代表每“题”他的需要时间。第三行： $m$ 个数。表示每项作业它的需要时间。第四行： $m$ 个数。代表每项作业它的分值。

输出格式：

一个数，代表小A能刷几道题

## 输入输出样例

输入样例#1：

1	3	4	20	100
2	15	20	50	
3	10	15	40	40
4	5	5	10	15

输出样例#1：

复制

1	2
---	---

## 说明

没有不能及格的情况

对于100%的数据， $n \leq 10, m \leq 10, k \leq 50, r \leq 150$

```

1 //周子涵
2 #include<bits/stdc++.h>
3 using namespace std;
4 int n,m,r,k,a[15],tt[15],t[15],fs[55],ans;
5 int main(){
6     cin>>n>>m>>k>>r;
7     for(int i=1;i<=n;i++) cin>>tt[i];
8     for(int i=1;i<=m;i++) cin>>t[i];
9     for(int i=1;i<=m;i++) cin>>a[i];
10    for(int i=1;i<=k;i++) fs[i]=250000;//要求为最小值,先赋值为用时很大
11    for(int i=1;i<=m;i++)//01背包,对于这道老师布置的题做或不作选择
12        for(int j=k;j>=a[i];j--)
13            if(fs[j]>=fs[j-a[i]]+t[i]) fs[j]=fs[j-a[i]]+t[i];
14    sort(tt+1,tt+n+1);//排序,为下面做用时少的喜欢的题目做预处理
15    r-=fs[k];//真正刷题的时间
16    int p=0;
17    while(r>=tt[++p]&& p<=n){//从用时短的题开始做,直到没时间了或所有题都做完了
18        r-=tt[p];
19        ans++;
20    }
21    cout<<ans;
22    return 0;
23 }
24

```

```

1 //殷学凯
2 #include<bits/stdc++.h>
3 using namespace std;
4 int n,m,k,r,cnt;
5 int st[100],t[100],f[100],little;
6 bool did[100];
7 void dfs(int fen,int use_time){//fen是分数,use_time是用去的时间;
8     if(fen>k){//如果及格了,不做了;
9         if(little>use_time) little=use_time;//更新最小值;
10        return;
11    }
12    for(int i=1;i<=m;i++){
13        if(did[i]==false){//如果没做过;
14            did[i]=true;//标记成做过的;
15            dfs(fen+f[i],use_time+t[i]);//dfs(分数+所得分数,时间+所用时间);
16            did[i]=false;//回溯;
17        }
18    }
19 }
20 int main() {
21     //输入;
22     cin>>n>>m>>k>>r;
23     little=151;
24     for(int i=1;i<=n;i++) cin>>st[i];
25     for(int i=1;i<=m;i++) cin>>t[i];
26     for(int i=1;i<=m;i++) cin>>f[i];
27     //做题;
28     dfs(0,0);
29     r-=little;
30     sort(st+1,st+n+1);//排序,
31     for(int i=1;i<=n;i++){//从小到大依次做;
32         r-=st[i];//减去时间;

```

```

33         if(r<0){//如果没时间了，就输出；
34             cout<<cnt;
35             return 0;
36         }
37         cnt++;//题数++;
38     }
39     cout<<cnt;//输出；
40     return 0;
41 }
42

```

```

1  //钱嘉欢
2  #include<iostream>
3  #include<algorithm>
4  using namespace std;
5  long long n,m,k,r;//题目上有
6  long long a[15]/*每“题”他的需要时间*/，
7             b[15]/*每项作业它的需要时间*/，
8             c[15]/*每项作业它的分值*/；
9  long long zd1;//最多还剩多少时间
10 long long zd2;//最多可以做几道题
11 long long yq;//用去的时间
12 void dfs(int p,int h,int sy)
13 {
14     if(h==k)
15     {
16         if(sy>zd1) zd1=sy;
17     }
18     for(int i=p+1;i<m;i++)
19     {
20         dfs(i,h+c[i],sy-b[i]);
21     }
22 }
23 int main()
24 {
25     cin>>n>>m>>k>>r;
26     for(int i=0;i<n;i++)
27         cin>>a[i];
28     for(int i=0;i<m;i++)
29         cin>>b[i];
30     for(int i=0;i<m;i++)
31         cin>>c[i];
32     //输入
33     sort(a,a+n);//要找最小的做
34     dfs(-1,0,r);
35     //搜索，找出最少要用多少时间
36     for(int i=0;i<n;i++)
37         if(a[i]<=zd1)
38         {
39             zd1-=a[i];
40             zd2++;
41         }else break;
42     //找出最多可以做几道题
43     cout<<zd2;
44     return 0;
45 }

```

```

1 //李明烨
2 #include<cstdio>
3 #include<iostream>
4 #include<algorithm>
5 using namespace std;
6 struct work{
7     long long t,s;
8 };
9 work w[15];
10 long long pro,wor,pass,w_t,p_t[15],zd=2147483647;
11 void check(long long ti){
12     zd=min(zd,ti);//比较
13     return ;
14 }
15 void dfs(long long now/*现在站在那个作业上*/,long long score/*目前得分*/,long long tim/*目前
    花的时间*/){
16     if(now==wor-1){//如果到了最后一个作业
17         if(score>=pass){//如果及格了
18             check(tim);//比较一下(其实这个函数根本不需要(￣▽￣))
19         }
20         else{
21             if(score+w[now].s>=pass){//如果加上这个点可以及格的话
22                 check(tim+w[now].t);//比较一下
23             }
24         }
25         return ;
26     }
27     if(tim>w_t){
28         return ;
29     }//如果没有那么多时间就不行
30     dfs(now+1/*来到下一个点*/,score+w[now].s/*加上这个点的得分*/,tim+w[now].t/*花掉*/);//枚举
1 : 做这个作业
31     dfs(now+1/*来到下一个点*/,score/*不加*/,tim/*不加*/);//枚举2 : 跳过这个作业
32 }
33 int main(){
34     scanf("%lld%lld%lld%lld",&pro,&wor,&pass,&w_t);
35     for(int i=0;i<pro;i++){
36         scanf("%lld",&p_t[i]);
37     }//输入
38     sort(p_t,p_t+pro);//从小到大排序备用
39     for(int i=0;i<wor;i++){
40         scanf("%lld",&w[i].t);
41     }
42     for(int i=0;i<wor;i++){
43         scanf("%lld",&w[i].s);
44     }//输入
45     dfs(0,0,0);//进行回溯(初始都是0)
46     w_t-=zd;//花掉这些时间
47     for(int i=0;i<pro;i++){
48         if(w_t-p_t[i]<=0){//如果没有那么多的时间了
49             printf("%d\n",i);//就输出
50             return 0;
51         }
52         w_t-=p_t[i];//否则花掉这些时间
53     }
54     printf("%lld\n",pro);//如果一直没有输出(都能刷)就输出所有题

```

```

55     return 0;
56 }
57
58 //思路：用dfs回溯找出做老师题目的最短时间，然后用模拟找出能刷的题
59

```

```

1  //佟书丞
2  // luogu-judger-enable-o2
3  #include<bits/stdc++.h> //万能头文件万岁!!!!
4  using namespace std;
5  int a[10001];
6  int b[10001];
7  int c[10001];
8  int d[10001];
9  int e[10001];
10 int main(){
11     //输入
12     int n,m,k,r;
13     cin>>n>>m>>k>>r;
14     for(int j=0;j<n;j++){
15         cin>>a[j];
16     }
17     for(int j=0;j<m;j++){
18         cin>>b[j];
19     }
20     for(int j=0;j<m;j++){
21         cin>>c[j];
22     }
23     //结束
24     memset(d,9999,sizeof(d));
25     memset(e,0,sizeof(e));
26     //.....准备擂台
27     d[0]=0;
28     for(int j=0;j<m;j++){
29         for(int i=k;i>=c[j];i--){
30             d[i]=min(d[i],d[i-c[j]]+b[j]);
31         }
32     } //只要到达及格就够 ( 记在 c[j] )
33     int l=r-d[k];
34     for(int j=0;j<n;j++){
35         for(int i=l;i>=a[j];i--){
36             e[i]=max(e[i],e[i-a[j]]+1);
37         }
38     } //总time-用过的时间
39     cout<<e[l]<<endl;
40 }
41
42

```

```

1
2 //刘宇鑫 P1926代码：
3 #include<iostream>
4 using namespace std;

```

```

5  int n,m,k,r,tt[11],zt[11],zf[11],minr=10000,ans;
6  bool zy[11]; //存放该作业是否做过
7  void newsj(int sj,int fs)
8  {
9      if(sj>minr&&fs<k) return ;
10     if(fs>=k&&sj<minr)
11     {
12         minr=sj;
13         return ;
14     }
15     for(int i=m;i>=1;i--)
16     {
17         if(r-sj>=zt[i]&&zy[i]==0)
18         {
19             zy[i]=1;
20             newsj(sj+zt[i],fs+zf[i]);
21         }
22     }
23 }
24 int main()
25 {
26     cin>>n>>m>>k>>r;
27     for(int i=1;i<=n;i++)
28         cin>>tt[i];
29     for(int i=1;i<=m;i++)
30         cin>>zt[i];
31     for(int i=1;i<=m;i++)
32         cin>>zf[i];
33     newsj(0,0); //求出做作业及格且时间最短的结果
34     r=r-minr;
35     for(int i=1;i<=n;i++)
36         for(int j=i+1;j<=n;j++)
37             if(tt[i]>tt[j])
38             {
39                 int q=tt[i];
40                 tt[i]=tt[j];
41                 tt[j]=q;
42             } //冒泡排序
43     for(int i=1;i<=n;i++)
44         if(r>=tt[i])
45         {
46             r-=tt[i];
47             ans++;
48         }
49     cout<<ans;
50     return 0;
51 }

```

思路：

先输入，然后用一个函数出做作业及格且时间最短的结果，并将总时间减去最短的时间，然后将题目的时间用冒泡排序从大到小排出来，最后求出可以做的最多的题数