# 算法组选拔赛题解

对于题目的修正：

1. 比赛时B题的题目在几个地方描述有误，可能影响了参赛选手的思路，还请见谅。（最终的题目和数据也将和题解一起发布）
2. D题在描述中没有提及是有几副扑克（出题人默认是一副），但比赛时在提示前就有选手考虑到了这点并AC了，然后在比赛还有1小时结束时judge给出了一副牌的提示。

B题没有人AC，D题提示是对全场的，所以比赛仍然是公平的，成绩不改变。

## A） Harry Lee’s Lunch

* 题意：先读入三个数M,R,D;如果M>=R+D,就输出R+D；如果M<R+D&&M是偶数，输出R; 如果M<R+D&&M是偶数，输出D;
* 思路：按照题意模拟就ＯＫ。

/\*

## Title:算法组选拔赛- A） Harry Lee’s Lunch

## Author: Plato

\*/

#include <cstdio>

#include <iostream>

#include <fstream>

#include <cstring>

#include <vector>

using namespace std;

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int T;

scanf("%d",&T);

while (T--)

{

int M,R,D;

scanf("%d%d%d",&M,&R,&D);

if (M >= R+D)

printf("%d\n",R+D);

else

{

if (M%2)

printf("%d\n",D);

else

printf("%d\n",R);

}

}

return 0;

}

## B）正太的上课计划

* 题意：有N节课，给出每节课的base、add花费，求在精力值为M时，能上课的最多数目。总花费的计算公式 sum{ base[i] + add[i]\*(M-1) }
* 思路：枚举+贪心。由计算公式可以看出实际每节课的花费与选课的总数M有关，并且M的范围比较小（M<=100），所以可以去枚举M。然后算出每节课的花费，贪心地取最少的那些课，验证M是否可以达成。

/\*

Title: 算法组选拔赛- A） Harry Lee’s Lunch

Author: Plato

\*/

#include <cstdio>

#include <iostream>

#include <fstream>

#include <cstring>

#include <vector>

#include <algorithm>

using namespace std;

int base[100010],add[100010],c[100010];

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int T;

scanf("%d",&T);

while (T--)

{

int N,total;

scanf("%d%d",&N,&total);

for (int i = 1;i <= N;i++) scanf("%d",&base[i]);

for (int i = 1;i <= N;i++) scanf("%d",&add[i]);

int k;

for (k = 1;k <= N;k++)

{

for (int i = 1;i <= N;i++)

c[i] = base[i] + add[i]\*(k-1);

sort(c+1,c+N+1);

int sum = 0;

for (int i = 1;i <= k;i++)

sum += c[i];

if (sum > total) break;

}

printf("%d\n",k-1);

}

return 0;

}

## C）教室调度方案

* 题意：三个部门各有A、B、C人，三个教室的座位数X、Y、Z；求一种分配的方案能使得有最多的人有座位。
* 思路：贪心。其实就是把人数最多的部门分到最大的教室，次大依次类推。

/\*

title: 算法组选拔赛 – C）教室调度方案

author: Plato

\*/

#include <cstdio>

#include <iostream>

#include <fstream>

#include <cstring>

#include <vector>

#include <algorithm>

using namespace std;

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int T;

scanf("%d",&T);

while (T--)

{

int a[10];

int b[10];

for (int i = 1;i <= 3;i++)

{

scanf("%d",&a[i]);

}

for (int i = 1;i <= 3;i++)

{

scanf("%d",&b[i]);

}

sort(a+1,a+4);

sort(b+1,b+4);

int ans = 0;

for (int i = 1;i <= 3;i++)

{

if (a[i] < b[i])

ans += a[i];

else

ans += b[i];

}

printf("%d\n",ans);

}

return 0;

}

## D）三基友打扑克

* 题意：根据某个计分规则，已知两个人的三张手牌，和第三个人的前两张手牌，求在使第三个人获胜的情况下，第三个人的最小手牌（不存在就输出impossible）。
* 思路：模拟+枚举。先计算出第一人和第二人的手牌得分，然后枚举A-K，计算出第三人的得分，如果高于前两人就输出答案。没有找到就输出impossible。需要注意的就是：一副扑克同一面值只有4张，需要统计每个面值的使用情况，再进行枚举。

/\*

title: 算法组选拔赛 –D）三基友打扑克

author: Plato

\*/

#include <cstdio>

#include <iostream>

#include <fstream>

#include <cstring>

#include <vector>

#include <algorithm>

using namespace std;

int used[19];

int count(int b[10])

{

int a[10] = {0};

for (int i = 1;i <= 3;i++)

a[i] = b[i];

sort(a+1,a+4);

if (a[1] == a[2] && a[2] == a[3])

{

return (a[1] + a[2] + a[3])\*3;

}

int sum = 0;

for (int i = 1;i <= 3;i++)

{

sum += a[i];

if(a[i] == a[i-1] || a[i] == a[i+1]) sum += a[i];

}

return sum;

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int T;

scanf("%d",&T);

while (T--)

{

memset(used,0,sizeof(used));

int b[10] = {0},m[10] = {0},s[10] = {0};

int BB,MM,SS;

scanf("%d%d",&b[1],&b[2]);

for (int i = 1;i <= 2;i++)

used[b[i]]++;

scanf("%d%d%d",&m[1],&m[2],&m[3]);

for (int i = 1;i <= 3;i++)

used[m[i]]++;

MM = count(m);

scanf("%d%d%d",&s[1],&s[2],&s[3]);

for (int i = 1;i <= 3;i++)

used[s[i]]++;

SS = count(s);

int maxx;

if (MM > SS)

maxx = MM;

else

maxx = SS;

int k;

for (k = 1;k <= 13;k++) if (used[k] < 4)

{

b[3] = k;

BB = count(b);

if (BB > maxx) break;

}

if(k != 14)

printf("%d\n",k);

else

printf("Impossible\n");

}

return 0;

}

## E）BigBenBird与妹子

* 题意：给出一些带权值的边，让你统计从0点出发到其他可到达点的单源最短路径之和。
* 思路： 就是一个单源最短路径的模版题，用spfa或者dijkstra什么的都行，判断一下加起来就可以了。

/\*

title: 算法组选拔赛- E) BigBenBird与妹子

author: ZH

\*/

#include <stdio.h>

#include <string.h>

#include <queue>

#define MAXN 100005

#define MAXM 500005

using namespace std;

struct Node

{

int v,w,next;

}edge[MAXM];

int e,head[MAXN],dis[MAXN],vis[MAXN];

int n,m;

const int INF=1000000000;

void addedge(int u,int v,int w)

{

edge[e].v=v;

edge[e].w=w;

edge[e].next=head[u];

head[u]=e++;

}

void spfa(int s)

{

for(int i=0;i<=n;i++)

{

dis[i]=INF;

vis[i]=0;

}

queue<int> q;

q.push(s);

dis[s]=0;

while(!q.empty())

{

int u=q.front();

q.pop();

vis[u]=false;

for(int i=head[u];i!=-1;i=edge[i].next)

{

int v=edge[i].v;

if(dis[v]>dis[u]+edge[i].w)

{

dis[v]=dis[u]+edge[i].w;

if(!vis[v])

{

vis[v]=true;

q.push(v);

}

}

}

}

}

int main()

{

//freopen("input.txt","r",stdin);

int t;

scanf("%d",&t);

while(t--)

{

memset(head,-1,sizeof(head));

scanf("%d%d",&n,&m);

for(int i=0;i<m;i++)

{

int u,v,w;

scanf("%d%d%d",&u,&v,&w);

addedge(u,v,w);

}

int ans=0;

spfa(0);

for(int i=1;i<=n;i++)

if(dis[i]!=INF)

ans+=dis[i];

printf("%d\n",ans);

}

}

## F) 挑剔的琪琪

* 题意：有一长度为N的序列，求其中最长的连续一段(要求在这个区间上的最大值-最小值<=M)。
* 思路：1.首先想到的思路就是枚举所有的区间，但是N的范围<=10^6,枚举所有区间肯定超时。

2.借用下枚举的思路，先枚举区间的左断点i:1->N，再枚举右端点j:i->N，可以发现一些规律：a）左端点固定时，当右端点移动到某个位置后max-min>M了，后面的区间也没有必要去检查了，称这个位置为右端点的结束位置。b）左端点移动一个位置后，右端点的结束位置一定>=上一次的结束位置。

新的想法就是基于上面发现的规律，枚举左端点i:1->N,右端点j:从上次的结束位置->N/不满足条件。

这里需要多次查找区间[i,j]里的最大值和最小值，直接枚举最值时间上也会超时，需要用单调队列O(1)/线段树O(logN)来快速找到区间的最值。有兴趣的同学可以自己去学习下。下面的代码是基于单调队列实现的。

/\*

title: 算法组选拔赛- F) 挑剔的琪琪

author: Plato

\*/

#include <cstdio>

#include <iostream>

#include <fstream>

#include <cstring>

#include <vector>

using namespace std;

#define MAX(a,b) ((a)>(b)?(a):(b))

#define MIN(a,b) ((a)<(b)?(a):(b))

int N,M;

int d[1000010];

void solve()

{

int ans = 0;

static int qmax[1000010],qmin[1000010];

int front1,front2,rear1,rear2;

front1 = front2 = 0;

rear1 = rear2 = 0;

int i,j;

for (i = 1,j = 1;j <= N;j++)

{

while (front1 < rear1 && d[qmax[rear1-1]] <= d[j]) rear1--;

qmax[rear1++] = j;

while (front2 < rear2 && d[qmin[rear2-1]] >= d[j]) rear2--;

qmin[rear2++] = j;

if (d[qmax[front1]] - d[qmin[front2]] > M)

{

ans = MAX(ans,j - i);

while (d[qmax[front1]] - d[qmin[front2]] > M)

{

i = MIN(qmax[front1],qmin[front2]) + 1;

while (front1 < rear1 && qmax[front1] < i) front1++;

while (front2 < rear2 && qmin[front2] < i) front2++;

}

}

}

ans = MAX(ans,j - i);

printf("%d\n",ans);

}

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int T;

scanf("%d",&T);

while(T--)

{

scanf("%d%d",&N,&M);

for (int i = 1;i <= N;i++)

scanf("%d",&d[i]);

solve();

}

return 0;

}

## G）MVP

* 题意：给出M件商品他们的花费Ｃ和价值Ｖ，商品分为两类：唯一的（只能购买一件），可随意购买。求在总费用为Ｎ时，所能购买的商品的最大总价值。
* 思路：动态规划－背包。很明显的一个背包了，商品唯一时就是０/1背包，不唯一时就是完全背包。(PS:数据很弱，完全背包枚举就OK了，所以难度也并不高)。

/\*

title: 算法组选拔赛-G）MVP

author : Plato

\*/

#include <cstdio>

#include <iostream>

#include <fstream>

#include <cstring>

#include <vector>

using namespace std;

#define MAX(a,b) (a)>(b)?(a):(b)

int main()

{

//freopen("input.txt","r",stdin);

//freopen("output.txt","w",stdout);

int T;

scanf("%d",&T);

while (T--)

{

static int c[100009],a[100009];

static bool u[100009];

int N,M;

char s[10];

scanf("%d%d",&N,&M);

for (int i = 1;i <= M;i++)

{

scanf("%d%d%s",&c[i],&a[i],s);

if (s[0] == 'U')

u[i] = true;

else

u[i] = false;

}

static int f[100009];

memset(f,0,sizeof(f));

for (int i = 1;i <= M;i++)

{

for (int j = N;j >= 1;j--)

{

if (u[i])

{

if (j - c[i] >= 0) f[j] = MAX(f[j],f[j - c[i]] + a[i]);

}

else

{

for (int m = j/c[i];m >= 1;m--)

f[j] = MAX(f[j],f[j - c[i]\*m] + a[i]\*m);

}

}

}

int maxx = 0;

for (int i = 1;i <= N;i++)

{

if (f[i] > maxx) maxx = f[i];

}

printf("%d\n",maxx);

}

return 0;

}