# Chafen

1275

#include<bits/stdc++.h>

using namespace std;

const int M = 123456;

struct Edge{

int to,val,next;

Edge(int to=0,int val=0,int next=0):to(to),val(val),next(next){}

}edge[M];

int size;

int head[M];

int n;

int t[30],r[30];

int dist[M],vis[M],inq[M];

inline int addedge(int A,int B,int C){

//printf("%d %d %d\n",A,B,C);

edge[++size]=Edge(B,C,head[A]);

head[A]=size;

return 0;

}

int init(){

size=0;

for(int i=0;i<=24;i++) {

dist[i]=-12345678;

head[i]=-1;

}

dist[0]=0;

return 0;

}

queue<int> q;

int start=0;

int build(int sum){

init();

for(int i=1;i<=8;i++){

addedge(16+i,i,t[i]-sum);

}

for(int i=9;i<=24;i++){

addedge(i-8,i,t[i]);

}

for(int i=1;i<=24;i++){

addedge(i-1,i,0);

addedge(i,i-1,-r[i]);

}

addedge(0,24,sum);

return 0;

}

int spfa(int sum){

build(sum);

while(!q.empty()) q.pop();

for(int i=0;i<=24;i++) vis[i]=0,inq[i]=0;

start=0;

q.push(start);

inq[start]=1;

while(!q.empty()){

int from=q.front();

q.pop();

//printf("A%2d %d %d\n",from,dist[from],sum);

inq[from]=0;

vis[from]++;

if(vis[from]>25){

//printf("FFFFFFFFFFF%d\n",from);

return 0;

}

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

int to=edge[i].to;

if(dist[to]<dist[from]+edge[i].val){

dist[to]=dist[from]+edge[i].val;

if(!inq[to]){

q.push(to);

inq[to]=1;

}

}

}

}

//printf("ABC%d %d\n",sum,dist[24]);

return dist[24]==sum;

}

int ans=12345678;

int binary(){

int left=1,right=n;

while(left<=right){

//printf("%d %d\n",left,right);

int mid=(left+right)/2;

if(spfa(mid)){

ans=min(ans,mid);

right=mid-1;

}

else left=mid+1;

//return 0;

}

return 0;

}

int main(){

freopen("A.in","r",stdin);

int T;

cin>>T;

while(T--){

for(int i=1;i<=24;i++){

cin>>t[i];

r[i]=0;

}

cin>>n;

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

int x;

cin>>x;

x++;

r[x]++;

}

ans=12345678;

binary();

if(ans==12345678) printf("No Solution\n");

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

}

return 0;

}