**A Reverse the lights**

#include <bits/stdc++.h>

using namespace std;

const long long inf = 1LL << 61;

const int maxn = 1e4 + 15;

long long dp[maxn];

int n , K , c[maxn];

void up( long long & x , long long v ) { x = min( x , v ) ;}

int main( int argc , char \* argv[] ){

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

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

scanf( "%d" , c + i);

dp[i] = inf;

}

for(int i = 1 ; i <= min( n , K + 1 ) ; ++ i) up( dp[ min( n , i + K ) ] , c[i] );

for(int i = 1 ; i + K + 1 <= n ; ++ i) if( dp[i] != inf ){

up( dp[min( n , i + 2 \* K + 1 )] , dp[i] + c[i + K + 1] );

}

cout << dp[n] << endl;

return 0;

}

**B Forgive her**

#include <bits/stdc++.h>

using namespace std;

struct query{

int index ;

int s ;

int k ;

query( int index , int s , int k ) : index( index ) , s(s) , k(k){}

};

char dp[52][202][2002];

int op[52][202][2002];

vector < query > ask[55];

vector < int > ans[105];

void PreDeal( int n ){

memset( dp , 0 , sizeof( dp ) );

dp[0][0][0] = 1;

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

for(int j = 0 ; j <= 200 ; ++ j)

for(int k = 0 ; k <= 2000 ; ++ k)

if( dp[i][j][k] ){

int f = (i + 1) \* (n - i - 1);

for(int add = 0 ; j + add \* (i + 1) <= 200 && k + add \* f <= 2000 ; ++ add){

int newi = i + 1 , newj = j + add \* ( i + 1 ) , newk = k + add \* f;

//if( !dp[newi][newj][newk]){

dp[newi][newj][newk] = 1 ;

op[newi][newj][newk] = add;

//}

}

}

}

int temp[105];

void getpath( int index , int n , int s , int k ){

if( !dp[n][s][k] ){

return ;

}

int len = n;

while( n ){

temp[n] = op[n][s][k];

s -= temp[n] \* n;

k -= temp[n] \* n \* (len - n);

-- n;

}

reverse( temp + 1 , temp + len + 1 );

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

temp[i] += temp[i - 1];

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

ans[index].push\_back( temp[i] );

}

int main( int argc , char \* argv[] ){

//for(int i = 1 ; i <= 50 ; ++ i)

// PreDeal( i );

//printf("run time is %d\n" , clock() );

int T;

scanf( "%d" , & T );

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

int n , s , k ;

scanf( "%d%d%d" , & n , & s , & k );

ask[n].push\_back( query( i , s , k ) );

}

for(int i = 1 ; i <= 50 ; ++ i)

if( ask[i].size() ){

PreDeal( i );

for(auto & it : ask[i])

getpath( it.index , i , it.s , it.k );

}

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

if( ans[i].size() == 0 )

puts("-1");

else{

for(int j = 0 ; j < ans[i].size() ; ++ j){

if( j )

putchar( 32 );

printf( "%d" , ans[i][j] );

}

puts("");

}

return 0;

}

**C**  **咸鱼魔法记**

#include <bits/stdc++.h>

using namespace std;

const int maxn = 3e5 + 15;

int N , K , a[maxn] , sum[maxn] , ans[maxn];

int main(int argc,char \*argv[]){

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

for(int i = 1 ; i <= N ; ++ i) scanf("%d",a+i);

for(int i = 1 ; i <= N ; ++ i) sum[i]=sum[i-1]+(a[i]^1);

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

if( K < ( a[i] ^ 1 ) ){

ans[i] = i - 1;

continue;

}

int l = i , r = N;

while( l < r ){

int mid = l + r + 1 >> 1;

if( sum[mid] - sum[i - 1] <= K ) l = mid;

else r = mid - 1;

}

ans[i] = l;

}

int mx = 0 , have = K ;

for(int i = 1 ; i <= N ; ++ i) if( mx == 0 || ans[i] - i + 1 >= ans[mx] - mx + 1 ) mx = i;

for(int i = mx ; i <= N && have ; ++ i) if(a[i] == 0) a[i] = 1 , -- have;

printf("%d\n" , max( 0 , ans[mx] - mx + 1 ) );

//for(int i = 1 ; i <= N ; ++ i) printf("%d " , a[i]);printf("\n");

return 0;

}

**D咸鱼商店**

#include<bits/stdc++.h>

using namespace std;

const int maxn = 1010;

int n,m,K;

pair<int,int> A[maxn];

long long f[maxn];

void Work(){

for(int i=n;i;i--){

for(int j=m;j>=A[i].second;j--)f[j]=max(f[j],f[j-A[i].second]+A[i].first);

if(f[m]>=K){

printf("%d\n",A[i].first);

return;

}

}

puts("-1");

}

void Init(){

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

for(int i=1;i<=n;i++)scanf("%d%d",&A[i].second,&A[i].first),A[i].second=-A[i].second;

sort(A+1,A+n+1);

for(int i=1;i<=n;i++)A[i].second=-A[i].second;

}

int main(){

Init();

Work();

return 0;

}

**E咸鱼旅行**

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 15;

struct edge{

int v , c , nxt;

}e[maxn \* 10];

int head[maxn] , tot , N , M , S , T , vis[maxn];

void link( int u , int v , int c ){

e[tot].v=v,e[tot].c=c,e[tot].nxt=head[u],head[u]=tot++;

}

queue < int > Q;

bool bfs( int limit ){

Q.push( S );

memset( vis , 0 , sizeof( vis ) );

vis[S] = 0;

while(!Q.empty()){

int x = Q.front() ; Q.pop() ;

for( int i = head[x] ; ~i ; i = e[i].nxt ){

int v = e[i].v;

if( e[i].c <= limit && !vis[v] ){

vis[v] = 1;

Q.push( v );

}

}

}

return vis[T] == 1;

}

int main( int argc , char \* argv[] ){

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

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

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

int u , v , c ;

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

link( u , v , c );

link( v , u , c );

}

scanf("%d%d",&S,&T);

if( !bfs( 1e9 + 1 ) ){

printf("-1\n");

return 0;

}

int l = 0 , r = 1e9;

while( l < r ){

int mid = l + r >> 1;

if( bfs( mid ) ) r = mid;

else l = mid + 1;

}

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

return 0;

}

**F咸鱼文章**

#include<bits/stdc++.h>

using namespace std;

stack<char> Q;

char c;

int main(){

while((c = getchar()) != EOF){

if(c==' '||c=='\n'){

while(!Q.empty()){

cout<<Q.top();

Q.pop();

}

cout<<c;

}else{

Q.push(c);

}

}

while(!Q.empty()){

cout<<Q.top();

Q.pop();

}

}

**G咸鱼拷问**

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 50;

struct Sgtree{

struct treenode{

int l , r , maxv , minv;

void Update( int x ){

maxv = minv = x;

}

}tree[maxn << 2];

void build( int l , int r , int o ){

tree[o].maxv = -1e9 , tree[o].minv = 1e9 , tree[o].l = l , tree[o].r = r;

if( r > l ){

int mid = l + r >> 1;

build( l , mid , o << 1 );

build( mid + 1 , r , o << 1 | 1 );

}

}

void Push\_up( int o ){

tree[o].maxv = max( tree[o << 1].maxv , tree[o << 1 | 1].maxv );

tree[o].minv = min( tree[o << 1].minv , tree[o << 1 | 1].minv );

}

void Modify( int x , int y , int o ){

int l = tree[o].l , r = tree[o].r;

if( l == r ) tree[o].Update( y );

else{

int mid = l + r >> 1;

if( x <= mid ) Modify( x , y , o << 1 );

else Modify( x , y , o << 1 | 1 );

Push\_up( o );

}

}

int AskMx( int ql , int qr , int o ){

int l = tree[o].l , r = tree[o].r;

if( ql <= l && r <= qr ) return tree[o].maxv;

else{

int mx = -1e9 , mid = l + r >> 1;

if( ql <= mid ) mx = max( mx , AskMx( ql , qr , o << 1 ) );

if( qr > mid ) mx = max( mx , AskMx( ql , qr , o << 1 | 1 ) );

return mx;

}

}

int AskMi( int ql , int qr , int o ){

int l = tree[o].l , r = tree[o].r;

if( ql <= l && r <= qr ) return tree[o].minv;

else{

int mi = 1e9 , mid = l + r >> 1;

if( ql <= mid ) mi = min( mi , AskMi( ql , qr , o << 1 ) );

if( qr > mid ) mi = min( mi , AskMi( ql , qr , o << 1 | 1 ) );

return mi;

}

}

}Sgtree;

int N ;

int main( int argc , char \* argv[] ){

scanf("%d",&N);

Sgtree.build( 1 , N , 1 );

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

int x ;

scanf("%d",&x);

Sgtree.Modify( i , x , 1 );

}

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

int x;

scanf("%d",&x);

printf("%I64d\n" , 1LL \* Sgtree.AskMx(i - x + 1 , i , 1 ) \* Sgtree.AskMi( i - x + 1 , i , 1 ) );

}

return 0;

}

**H喵哈哈村的战斗魔法师丶坏坏い月**

#include<bits/stdc++.h>

using namespace std;

const int maxn = 1e6+7;

int n,m,num,belong[maxn],block,l[maxn],r[maxn];

long long a[maxn],Max[maxn],Upd[maxn];

inline long long read()

{

long long x=0,f=1;char ch=getchar();

while(ch<'0'||ch>'9'){if(ch=='-')f=-1;ch=getchar();}

while(ch>='0'&&ch<='9'){x=x\*10+ch-'0';ch=getchar();}

return x\*f;

}

void build(){

block=sqrt(n);

num=n/block;if(n%block)num++;

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

l[i]=(i-1)\*block+1,r[i]=i\*block;

r[num]=n;

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

belong[i]=(i-1)/block+1;

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

Max[i]=0;

Upd[i]=0;

for(int j=l[i];j<=r[i];j++){

Max[i]=max(Max[i],a[j]);

}

}

}

void update(int L,int R,long long k){

if(belong[L]==belong[R]){

for(int i=L;i<=R;i++){

a[i]+=k;

Max[belong[L]]=max(Max[belong[i]],a[i]);

}

}else{

for(int i=L;i<=r[belong[L]];i++){

a[i]+=k;

Max[belong[i]]=max(Max[belong[i]],a[i]);

}

for(int i=belong[L]+1;i<belong[R];i++)

Upd[i]+=k;

for(int i=l[belong[R]];i<=R;i++){

a[i]+=k;

Max[belong[i]]=max(Max[belong[i]],a[i]);

}

}

}

void query(int L,int R,long long k){

int ans = -1;

if(belong[L]==belong[R]){

for(int i=L;i<=R;i++){

if(a[i]+Upd[belong[L]]>=k){

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

return;

}

}

}else{

for(int i=L;i<=r[belong[L]];i++){

if(a[i]+Upd[belong[L]]>=k){

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

return;

}

}

for(int i=belong[L]+1;i<belong[R];i++){

if(Max[i]+Upd[i]>=k){

for(int j=l[i];j<=r[i];j++){

if(a[j]+Upd[belong[j]]>=k){

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

return;

}

}

}

}

for(int i=l[belong[R]];i<=R;i++){

if(a[i]+Upd[belong[R]]>=k){

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

return;

}

}

}

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

}

void solve(){

long long Ans = 0;

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

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

a[i]=read();

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

int op,l,r;

long long v;

op=read(),l=read(),r=read(),v=read();

if(op==2){

update(l,r,v);

}else{

query(l,r,v);

}

}

}

int main()

{

int t;t=read();

while(t--){

solve();

}

}

**I喵哈哈村的魔法大师╳灬兲笙疯癫°月**

#include <cstring>

#include <algorithm>

#include <cstdio>

#include <stack>

#include <queue>

using namespace std;

const int INF = 0x3f3f3f3f;

const int MAXN = 505;

const int MAXM = MAXN\*MAXN;

const int MAXK = 505;

int n, m, k;

int a[MAXK], c[MAXK];

namespace Graph {

int head[MAXN], nxt[MAXM], to[MAXM], eidx;

void init() {

eidx = 0;

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

}

void adde( int u, int v ) {

to[eidx] = v, nxt[eidx] = head[u], head[u] = eidx++;

}

}

void input() {

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

Graph::init();

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

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

Graph::adde(u,v);

}

for( int i = 0; i < k; ++i ) scanf( "%d%d", a+i, c+i );

}

namespace SCC {

int dfn[MAXN], low[MAXN], dfn\_cnt;

int sccno[MAXN], scc\_cnt;

stack<int> stk;

void dfs( int u ) {

using namespace Graph;

dfn[u] = low[u] = ++dfn\_cnt;

stk.push(u);

for( int i = head[u]; ~i; i = nxt[i] ) {

int v = to[i];

if( !dfn[v] ) {

dfs(v);

low[u] = min( low[u], low[v] );

} else if( !sccno[v] ) {

low[u] = min( low[u], dfn[v] );

}

}

if( low[u] == dfn[u] ) {

++scc\_cnt;

while(1) {

int nod = stk.top(); stk.pop();

sccno[nod] = scc\_cnt;

if( nod == u ) break;

}

}

}

void solve() {

dfn\_cnt = scc\_cnt = 0;

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

dfn[i] = sccno[i] = 0;

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

if( !dfn[i] ) dfs(i);

}

}

using SCC::scc\_cnt;

using SCC::sccno;

namespace Dinic {

const int MAXV = 2\*MAXN;

const int MAXE = MAXN\*2 + MAXN\*MAXN;

struct Edge {

int u, v, c, f;

Edge(){}

Edge( int u, int v, int c, int f ):

u(u),v(v),c(c),f(f){}

};

int n, m, s, t;

int head[MAXV], nxt[MAXE<<1];

Edge edge[MAXE<<1];

void init( int n2 ) {

n = n2, m = 0;

for( int i = 0; i < n; ++i ) head[i] = -1;

}

void adde( int u, int v, int c ) {

edge[m] = Edge(u,v,c,0);

nxt[m] = head[u], head[u] = m++;

edge[m] = Edge(v,u,0,0);

nxt[m] = head[v], head[v] = m++;

}

queue<int> bfsq;

int cur[MAXV], dis[MAXV];

bool bfs() {

for( int i = 0; i < n; ++i ) dis[i] = INF;

dis[s] = 0, bfsq.push(s);

while( !bfsq.empty() ) {

int u = bfsq.front(); bfsq.pop();

for( int i = head[u]; ~i; i = nxt[i] ) {

Edge &e = edge[i];

if( e.c > e.f && dis[e.v] == INF ) {

dis[e.v] = dis[u] + 1;

bfsq.push(e.v);

}

}

}

return dis[t] != INF;

}

int dfs( int u, int res ) {

if( u == t || res == 0 ) return res;

int flow = 0;

for( int &i = cur[u]; ~i; i = nxt[i] ) {

Edge &e = edge[i];

if( e.c > e.f && dis[e.v] == dis[u] + 1 ) {

int f = dfs( e.v, min( e.c-e.f, res ) );

flow += f, res -= f;

e.f += f, edge[i^1].f -= f;

if( !res ) break;

}

}

return flow;

}

int solve( int s2, int t2 ) {

s = s2, t = t2;

int flow = 0;

while( bfs() ) {

for( int i = 0; i < n; ++i ) cur[i] = head[i];

flow += dfs(s, INF);

}

return flow;

}

}

bool g[MAXN][MAXN] = {0};

void floyd() {

for( int k = 1; k <= scc\_cnt; ++k )

for( int i = 1; i <= scc\_cnt; ++i )

for( int j = 1; j <= scc\_cnt; ++j )

g[i][j] |= g[i][k] && g[k][j];

}

int prelude() {

using namespace Graph;

SCC::solve();

for( int i = 1; i <= scc\_cnt; ++i )

for( int j = 1; j <= scc\_cnt; ++j )

g[i][j] = i == j;

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

for( int i = head[u]; ~i; i = nxt[i] ) {

int v = to[i];

g[sccno[u]][sccno[v]] = true;

}

}

floyd();

int s = scc\_cnt\*2+1, t = s+1;

Dinic::init(t+1);

for( int i = 1; i <= scc\_cnt; ++i ) {

Dinic::adde(s, i, 1);

Dinic::adde(i+scc\_cnt, t, 1);

}

for( int i = 1; i <= scc\_cnt; ++i )

for( int j = 1; j <= scc\_cnt; ++j )

if( i != j && g[i][j] )

Dinic::adde(i, j+scc\_cnt, 1);

return scc\_cnt - Dinic::solve(s,t);

}

int dp[MAXN];

void solve( int cnt ) {

int sum = 0;

for( int i = 0; i < k; ++i ) sum += c[i];

if( sum < cnt ) {

puts("-1");

return;

}

for( int i = 0; i <= cnt; ++i ) dp[i] = INF;

dp[0] = 0;

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

for( int j = cnt; j >= 0; --j ) {

if( c[i] >= j ) dp[j] = min( dp[j], a[i] );

else dp[j] = min( dp[j], dp[j-c[i]] + a[i] );

}

}

printf( "%d\n", dp[cnt] );

}

int main() {

int T; scanf( "%d", &T );

while( T-- ) {

input(), solve( prelude() );

}

return 0;

}

**J喵哈哈村的几何大师╰☆莣メ誋こ月**

#include<bits/stdc++.h>

using namespace std;

const double pi = acos(-1.0);

int main(){

freopen("1.in","r",stdin);

freopen("1.out","w",stdout);

int t;

scanf("%d",&t);

while(t--){

double a,b,c,d;

scanf("%lf%lf%lf%lf",&a,&b,&c,&d);

double f = 180 - a - b - c - d;

double CD = sin((c+d)\*pi/180.0) / sin(f\*pi/180.0);

double D = sin(d\*pi/180.0) / sin((180-a-b-d)\*pi/180.0);

double B = sin(b\*pi/180.0) / sin((180-c-d-b)\*pi/180.0);

double X1 = CD - D;

double X2 = CD - B;

double X3 = sqrt(X1\*X1+X2\*X2-2\*X1\*X2\*cos(f\*pi/180.0));

double x4 = acos((X3\*X3+X2\*X2-X1\*X1)/(2.0\*X3\*X2));

double X4 = x4\*180.0/pi;

double h = 180 - b - c - d;

double ans = 180 - h - X4;

printf("%.2f\n",ans);

}

}