# Abc

1057

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2005;

int n, m;

struct node {

int x, y, left;

const bool operator > (const node b) {

return this->left > b.left;

}

};

stack<node> s;

int c[maxn][maxn], x[maxn][maxn], y[maxn][maxn], up[maxn];

int ans1 = 0, ans2 = 0;

void solve1() {

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

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

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

node a = (node){i, j, x[i][j]};

node b =(node){0, j, 0};

if(!s.empty()){

b = s.top();

while(b > a) {

s.pop();

ans2 = max(ans2, (a.x - up[b.x]) \* b.left);

ans1 = max(ans1, min(b.left, a.x - up[b.x]) \* min(b.left, a.x-up[b.x]));

if(s.empty()){

b = (node){0, j, 0};

break;

}

b = s.top();

}

}

up[a.x] = b.x+1;

s.push(a);

}

while(!s.empty()) {

node a = s.top();

s.pop();

ans2 = max(ans2, (n+1 - up[a.x]) \* a.left);

ans1 = max(ans1, min(a.left, n - up[a.x] + 1) \* min(a.left, n-up[a.x]+1));

}

}

}

void solve2() {

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

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

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

node a = (node){i, j, y[i][j]};

node b =(node){0, j, 0};

if(!s.empty()){

b = s.top();

while(b > a) {

s.pop();

ans2 = max(ans2, (a.x - up[b.x]) \* b.left);

ans1 = max(ans1, min(b.left, a.x - up[b.x]) \* min(b.left, a.x-up[b.x]));

if(s.empty()){

b = (node){0, j, 0};

break;

}

b = s.top();

}

}

up[a.x] = b.x+1;

s.push(a);

}

while(!s.empty()) {

node a = s.top();

s.pop();

ans2 = max(ans2, (n - up[a.x] + 1) \* a.left);

ans1 = max(ans1, (min(a.left, (n-up[a.x]+1)) \* min(a.left, (n-up[a.x]+1))));

}

}

}

int main() {

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

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

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

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

if((i+j)%2==1) {

c[i][j] ^= 1;

}

x[i][j] = c[i][j] == 1?x[i][j-1]+1:0;

y[i][j] = c[i][j] == 0?y[i][j-1]+1:0;

}

}

solve1();

solve2();

printf("%d\n%d", ans1, ans2);

}

1082

#include <iostream>

#define lowbit(i) (i & (-i))

using namespace std;

const int Nmax = 200100;

int N, Q;

long long delta[Nmax]; // delta的前缀和

long long deltai[Nmax]; // delta \* i的前缀和

long long sum[Nmax]; // 原始前缀和

long long Query(long long \*array, int pos){

long long temp = 0ll;

while(pos > 0)

{

temp += array[pos];

pos -= lowbit(pos);

}

return temp;

}

void Update(long long \*array, int pos, int x){

while(pos <= N){

array[pos] += x;

pos += lowbit(pos);

}

}

int main(){

N = readint();

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

int x = readint();

sum[i] = sum[i - 1] + x;

}

Q = readint();

while(Q--){

int sign = readint();

if(sign == 1) {// 修改：把[l, r]区间均加上x

int l = readint(), r = readint(), x = readint();

Update(delta, l, x);

Update(delta, r+1, -x);

Update(deltai, l, x \* l);

Update(deltai, r+1, -x \* (r+1));

}

else {// 查询：[l, r]区间和

int l = readint(), r = readint();

long long suml = sum[l - 1] + l \* Query(delta, l - 1) - Query(deltai, l - 1);

long long sumr = sum[r] + (r + 1) \* Query(delta, r) - Query(deltai, r);

printf("%lld\n", sumr - suml);

}

}

return 0;

}

1830

#include<bits/stdc++.h>

using namespace std;

int s[35],e[35],a[35][35],n;

int gauss(){

int u,c;

for(u=1,c=1;u<=n&&c<=n;c++){

int k=0;

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

if(a[i][c]){k=i; break;}

}

if(a[k][c]){

for(int j=1;j<=n+1;j++) swap(a[k][j],a[u][j]);

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

if(a[i][c]){

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

a[i][j]^=a[u][j];

}

}

}

u++;

}

}

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

if(a[i][n+1]) return -1;

}

return 1<<(n-u+1);

}

int main(){

int T;

scanf("%d",&T);

while(T--){

scanf("%d",&n);

memset(a,0,sizeof(a));

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

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

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

scanf("%d",&e[j]);

if(s[j]!=e[j])

a[j][n+1]=1;

a[j][j]=1;

}

int x,y;

while(scanf("%d%d",&x,&y)&&x+y){a[y][x]=1;}

int ans=gauss();

if(ans==-1)

printf("Oh,it's impossible~!!\n");

else

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

}

return 0;

}

3584

#include<iostream>

using namespace std;

int c[105][105][105];

int n;

int lowbit(int x){ return x&-x;}

int update(int x,int y,int z,int k){

for(int i=x; i<=n; i+=lowbit(i))

for(int j=y; j<=n; j+=lowbit(j))

for(int l=z; l<=n; l+=lowbit(l))

c[i][j][l]+=k;

return 0;

}

int sum(int x,int y,int z){

int sum=0;

for(int i=x; i>0; i-=lowbit(i))

for(int j=y; j>0; j-=lowbit(j))

for(int k=z; k>0; k-=lowbit(k))

sum+=c[i][j][k];

return sum;

}

int main(){

int m;

int x2,x1,y2,y1,z2,z1,k;

while(~scanf("%d %d",&n,&m)){

memset(c,0,sizeof(c));

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

scanf("%d %d %d %d",&k,&x1,&y1,&z1);

if(k==0)

printf("%d\n",sum(x1,y1,z1)%2);

else{

scanf("%d %d %d",&x2,&y2,&z2);

update(x1,y1,z1,1);

update(x1,y1,z2+1,1);

update(x1,y2+1,z1,1);

update(x1,y2+1,z2+1,1);

update(x2+1,y1,z1,1);

update(x2+1,y1,z2+1,1);

update(x2+1,y2+1,z1,1);

update(x2+1,y2+1,z2+1,1);

}

}

}

return 0;

}

5534

#include<bits/stdc++.h>

using namespace std;

const int M = 5000+123;

long long a[M];

long long f[2][M\*2];

struct pos{

int x,y;

}P[M\*M];

int sz;

int bin[20];

int main(){

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

bin[0]=0;

bin[1]=1;

for(int i=2;i<18;i++) bin[i]=bin[i-1]\*2;

int T;

cin>>T;

while(T--){

int n;

cin>>n;

for(int i=1;i<n;i++)cin>>a[i];

for(int j=0;j<=n;j++){

f[0][j]=-123456789;

f[1][j]=-123456789;

}

sz=0;

f[0][0]=n\*a[1];

int flag=0;

for(int i=1;i<=n-2;i++){

int K=(n-2)/i;//拆分一个大小为K的背包

int j=0;

for(int j=1;;j++){

if(bin[j]>K)break;

sz++;

P[sz].x=bin[j];

P[sz].y=i;

K-=bin[j];

}

if(K){

sz++;

P[sz].x=K;

P[sz].y=i;

}

}

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

flag=flag^1;

for(int j=0;j<=n-2;j++){

f[flag][j]=f[flag^1][j];

if(j-P[i].x\*P[i].y<0) continue;

//printf("HH%d %d %d %d %d %d\n",f[flag][j],i,j,P[i].x,P[i].y,j);

f[flag][j]=max(f[flag][j],f[flag^1][j-P[i].x\*P[i].y]+P[i].x\*(a[P[i].y+1]-a[1]));

}

}

printf("%lld\n",f[flag][n-2]);

}

return 0;

}