**枚举和分治**

一、画家问题

#include <iostream>

#include <string.h>

using namespace std;

int a[20][20] = { 0 };//yellow==1;white==0;

int temp[20][20];

int n;

int cc = 99999;

int minn = 99999;

int getbit(int i, int j)

{

return (i >> j) & 1;

}

void change(int i, int j)

{

//cc++;

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

if(i>0)temp[i-1][j] = 1 - temp[i-1][j];

if(i<n-1)temp[i+1][j] = 1 - temp[i+1][j];

if(j>0)temp[i][j-1] = 1 - temp[i][j-1];

if(j<n-1)temp[i][j+1] = 1 - temp[i][j+1];

}

int main()

{

cin >> n;

char c;

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

{

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

{

cin >> c;

if (c == 'w')a[i][j] = 0;

else a[i][j] = 1;

}

}

for (int i = 0; i < (1<<n); i++)

{

int flag = 1;

memcpy(temp, a, sizeof(a));

cc = 0;

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

{

if (getbit(i, j) == 1)

{

change(0, j);

cc++;

}

}

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

{

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

{

if (temp[k - 1][m] == 0)

{

change(k, m);

cc++;

}

}

}

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

{

if (temp[n - 1][j] == 0)

{

flag = 0;

break;

}

}

if (flag == 1)

{

if (cc < minn)

{

minn = cc;

}

}

cc = 99999;

}

if (minn == 99999)cout << "inf";

else cout << minn;

}

二、Sophie开关灯

#include <iostream>

using namespace std;

int main() {

int n, m, ca;

cin >> n >> m;

int a[n + 3] = {0}, b[n + 3] = {0}, t[n + 3];

int x = 0, y = 0, num = 1;

t[0] = 0;

t[n + 1] = m;

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

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

if(i % 2 == 1) x += t[i] - t[i - 1];

else y += t[i] - t[i - 1];

}

b[0] = y;

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

if(i % 2 == 1) {

a[i] = a[i - 1] + t[i] - t[i - 1];

b[i] = b[i - 1];

}

else {

a[i] = a[i - 1];

b[i] = b[i - 1] - t[i] + t[i - 1];

}

}

int maxn = x;

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

maxn = max(maxn, a[i] + b[i] - 1);

}

cout << maxn << endl;

return 0;

}

三、Sophie吃甘蔗

**#include** <iostream>

**#include** <vector>

**using** **namespace** std;

long long N, L;

vector<long long>a;

vector<long long>total;

int **main**()

{

cin >> N >> L;

long long t;

**while** (cin >> t)

{

a.**push\_back**(t);

}

**for** (long long i = 0; i <= L; i++)

{

long long temp = 0;

**for** (long long j = N - 1;j>= N-(L-i); j--)

{

temp += a[j];

}

**for** (long long k = 0; k < i; k++)

{

temp += a[k];

}

total.**push\_back**(temp);

*//cout << temp<<endl;*

}

long long max = total[0];

**for** (long long tt : total)

{

max = tt > max ? tt : max;

}

cout << max;

}

四、最大元素

**#include** <iostream>

**#include** <vector>

**#include** <algorithm>

**#include** <iomanip>

**using** **namespace** std;

int **maxv**(**const** vector<int>& v)

{

int ans = v[0];

**for** (int i = 1; i < v.**size**(); i++)

{

ans = v[i] > ans ? v[i] : ans;

}

**return** ans;

}

int **main**()

{

vector<int>h;

int t;

**while** (cin >> t)

{

h.**push\_back**(t);

}

cout << **maxv**(h);

}

五、数据清洗

**#include** <iostream>

**#include** <vector>

**#include** <algorithm>

**#include** <iomanip>

**using** **namespace** std;

int **main**()

{

int m, n;

cin >> m >> n;

vector<int>x;

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

{

int t;

cin >> t;

x.**push\_back**(t);

}

**sort**(x.**begin**(), x.**end**());

**for** (int j : x)

{

cerr << j << " ";

}

cerr << endl;

int a=0;

**for** (int i = m; i < n - m; i++)

{

a += x[i];

cerr << x[i]<<" ";

}

cerr << endl;

cout << fixed<<**setprecision**(2)<<(double)a / (n - 2 \* m);

}

六、数组合并

**#include** <iostream>

**#include** <vector>

**#include** <algorithm>

**using** **namespace** std;

int **main**()

{

int a, b;

cin >> a>>b;

vector<int>**x**(a);

vector<int>**y**(b);

**for**(int i=0;i<a;i++)

{

int temp;

cin >> temp;

x.**push\_back**(temp);

}

**for** (int i = 0; i < b; i++)

{

int temp;

cin >> temp;

y.**push\_back**(temp);

}

vector<int>c;

**merge**(x.**begin**(), x.**end**(), y.**begin**(), y.**end**(),**back\_inserter**(c));

**for** (int t = 0; t < c.**size**(); t++)

{

cerr<< c[t] << " ";

}

cerr << endl;

vector<int>d;

**unique\_copy**(c.**begin**(), c.**end**(), **back\_inserter**(d));

**for** (int t=1;t<d.**size**()-1;t++)

{

cout << d[t] << " ";

}

cout << d[d.**size**() - 1];

}

**回溯**

一、迷宫问题

/\*

#include <iostream>

#include <vector>

using namespace std;

struct node

{

int x, y;

node(int x\_, int y\_) :x{ x\_ },y{y\_}{}

};

int maze[5][5];

vector<node>minroute;

vector<node>route;

int minnum = 30;

int dx[4] = { 0,1,0,-1 };

int dy[4] = { 1,0,-1,0 };

void bfs(int x, int y)

{

route.push\_back(node(x, y));

if (x == 4 && y == 4)

{

if (route.size() < minnum)

{

minroute = route;

minnum = route.size();

}

return;

}

maze[x][y] = 1;

for (int i = 0; i < 4; i++)

{

int xx = x + dx[i];

int yy = y + dy[i];

if (xx >= 0 && xx <= 4 && yy >= 0 && yy <= 4 && maze[xx][yy] == 0)

{

bfs(xx, yy);

}

}

route.pop\_back();

maze[x][y] = 0;

}

int main()

{

minroute.clear();

route.clear();

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

cin >> maze[i][j];

}

}

bfs(0, 0);

for (auto& p : minroute)

{

printf("(%d, %d)\n", p.x, p.y);

}

return 0;

}

\*/

#include <iostream>

#include <vector>

#include <queue>

#include <stack>

using namespace std;

int maze[5][5];

int pre[5][5];

struct node

{

int x, y;

node(int x\_, int y\_) :x{ x\_ }, y{ y\_ }{}

};

int dx[4] = { 0,1,0,-1 };

int dy[4] = { 1,0,-1,0 };

int main()

{

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

cin >> maze[i][j];

}

}

queue<node>que;

node start = { 0,0 };

que.push(start);

while (!que.empty())

{

node temp = que.front();

que.pop();

if (temp.x == 4 && temp.y == 4)

{

break;

}

for (int i = 0; i < 4; i++)

{

int xx = temp.x + dx[i];

int yy = temp.y + dy[i];

if (xx >= 0 && xx <= 4 && yy >= 0 && yy <= 4 && maze[xx][yy] == 0)

{

node tt = { xx,yy };

que.push(tt);

pre[xx][yy] = i;

}

}

}

stack<node>ss;

node p = { 4,4 };

while (p.x != 0 || p.y != 0)

{

ss.push(p);

node temp = { p.x - dx[pre[p.x][p.y]],p.y - dy[pre[p.x][p.y]] };

p = temp;

}

printf("(0, 0)\n");

while (!ss.empty())

{

p = ss.top();

printf("(%d, %d)\n", p.x, p.y);

ss.pop();

}

}

二、棋盘问题

**#include** <iostream>

**using** **namespace** std;

int tot = 0;

int visited[10] = { 0 };

int cur = 0;

int n, t;

char x[10][10];

void **f**(int h)

{

**if** (cur == t)

{

tot++;

**return**;

}

**if** (h >= n)**return**;

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

{

**if** (visited[i] != 1 && x[h][i] == '#')

{

visited[i] = 1;

cur++;

**f**(h + 1);

visited[i] = 0;

cur--;

}

}

**f**(h + 1);

}

int **main**()

{

**while**(cin>>n>>t)

{

**if** (n == -1)**break**;

**else**

{

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

{

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

{

cin >> x[i][j];

}

}

**f**(0);

**for** (int i = 0; i < 10; i++)

{

visited[i] = 0;

}

cout << tot<<endl;

tot = 0;

cur = 0;

}

}

}

三、神奇的口袋

**#include**<bits/stdc++.h>

**using** **namespace** std;

**const** int nl=404;

**const** int mod=10000;

int a[nl];

int f[nl];

int **main**(){

int n;

cin>>n;

f[0]=1;

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

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

}

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

**for**(int j=400;j>=a[i];j--){

f[j]+=f[j-a[i]];

f[j]%=mod;

}

}

cout<<f[400];

}

**#include** <iostream>

**#include**<cstring>

**using** **namespace** std;

int n;

int vol[205];

int total[205][405];

int **f**(int num, int remain)

{

**if** (remain == 0)**return** 1;

**if** (num >= n)**return** 0;

**if** (total[num][remain] != -1)**return** total[num][remain];

**if** (remain < vol[num])**return** **f**(num + 1, remain);

**else**

{

total[num][remain] = (**f**(num + 1, remain - vol[num]) + **f**(num + 1, remain)) % 10000;

}

**return** total[num][remain];

}

int **main**()

{

cin >> n;

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

{

cin >> vol[i];

}

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

{

**for** (int j = 0; j < 405; j++)

{

total[i][j] = -1;

}

}

cout << **f**(0, 400);

}

四、数字方格

**#include** <iostream>

**using** **namespace** std;

int **main**()

{

int n,max=0;

cin >> n;

**for** (int i = n; i >= 0; i--)

{

**for** (int j = n; j >= 0; j--)

{

**for** (int k = n; k >= 0; k--)

{

**if** ((i + j) % 2 == 0 && (j + k) % 3 == 0 && (i + j + k) % 5 == 0)

{

max= (i + j + k)>max? (i + j + k):max;

}

}

}

}

cout << max;

}

**深搜**

一、硬币找零

**#include** <iostream>

**#include** <algorithm>

**#include** <cstring>

**using** **namespace** std;

int a[55],d[100005];

int n,tar;

int **main**()

{

cin >> n>>tar;

**memset**(d, 10000, **sizeof**(d));

d[0] = 0;

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

{

cin >> a[i];

}

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

{

**for** (int j = a[i]; j <=tar; j++)

{

d[j] = **min**(d[j], d[j- a[i]] + 1);

}

}

**for** (int i = tar; i >= 0; i--)

{

**if** (d[i] < 10000)cout << d[i];

**break**;

}

}

**二、雷达安装**

#include <iostream>

#include <algorithm>

using namespace std;

int n;

double d;

struct island

{

int x, y;

double left, right;

bool operator <(const island& is)

{

return left < is.left;

}

}a[1005];

int main()

{

int case\_num = 0;

while (cin >> n >> d && n && d)

{

case\_num++;

int c = 1;

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

{

cin >> a[i].x>>a[i].y;

if (d < a[i].y||d<0||a[i].y<0)

{

c = -1;

}

else

{

a[i].left = a[i].x - sqrt(d \* d - a[i].y \* a[i].y);

a[i].right= a[i].x + sqrt(d \* d - a[i].y \* a[i].y);

}

}

sort(a, a + n);

double temp = a[0].right;

for (int i = 1; i < n&&c!=-1; i++)

{

if (a[i].right < temp)temp = a[i].right;

else if (a[i].left > temp)

{

c++;

temp = a[i].right;

}

}

std::cout <<"Case "<<case\_num<<": "<<c<<endl;

}

}

广搜

一、抓住那头牛

**#include** <iostream>

**#include** <queue>

**#include** <algorithm>

**#define** max\_n 100001

**using** **namespace** std;

int visit[max\_n];

int n, k;

int **main**()

{

cin >> n >> k;

queue<int>que;

**fill**(visit, visit +max\_n, -1);

que.**push**(n);

visit[n] = 0;

**while** (!que.**empty**())

{

int cur= que.**front**();

que.**pop**();

**if** (cur == k)

{

cout << visit[cur];

**return** 0;

}

**else**

{

int loc = cur;

int stt = visit[cur] + 1;

**if** (visit[loc - 1] == -1 && loc-1 >= 0)

{

que.**push**(loc - 1);

visit[loc - 1] = stt;

}

**if** (visit[loc + 1] == -1 && loc+1 <= max\_n)

{

que.**push**(loc + 1);

visit[loc + 1] = stt;

}

**if** (loc \* 2 <= max\_n && visit[loc \* 2] == -1)

{

que.**push**(loc \* 2);

visit[loc \* 2] = stt;

}

}

}

}

二、象棋中马的问题

#include <iostream>

#include <queue>

#include <cstring>

using namespace std;

int bar[100][100];

int dir[8][2] = { -1, 2, 1, 2, 2, 1, 2, -1, 1, -2, -1, -2, -2, -1, -2, 1 };

int bar\_dir[8][2] = { 0, 1, 0, 1, 1, 0, 1, 0, 0, -1, 0, -1, -1, 0, -1, 0 };

struct node

{

int x, y, step;

node(int x\_,int y\_,int step\_):x{x\_},y{y\_},step{step\_}{}

};

int p, q;

int sx, sy, ex, ey;

int barnum;

int visit[100][100];

int main()

{

int n;

cin >> n;

while (n--)

{

cin >> p >> q>>sx>>sy>>ex>>ey;

cin >> barnum;

int flag = 0;

int minstep;

memset(bar, 0, sizeof(bar));

memset(visit, 0, sizeof(visit));

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

{

int z, w;

cin >> z >> w;

bar[z][w] = 1;

}

queue<node>que;

node start = { sx,sy,0 };

visit[sx][sy] = 1;

que.push(start);

while (!que.empty())

{

node cur = que.front();

que.pop();

if (cur.x == ex && cur.y == ey)

{

flag = 1;

minstep = cur.step;

break;

}

for (int i = 0; i < 8; i++)

{

int xx = cur.x + dir[i][0];

int yy = cur.y + dir[i][1];

int bar\_x = cur.x + bar\_dir[i][0];

int bar\_y = cur.y + bar\_dir[i][1];

if (visit[xx][yy] == 0 && bar[bar\_x][bar\_y] == 0&&bar[xx][yy]==0

&& xx >= 0 && xx <= p && yy >= 0 && yy <= q)

{

que.push(node(xx, yy, cur.step + 1));

visit[xx][yy] = 1;

}

}

}

if (flag == 1)cout << minstep<<endl;

else cout << "can not reach!"<<endl;

}

}

三、拯救公主

#include <iostream>

#include <queue>

using namespace std;

int mov[4][2] = { {1,0},{-1,0},{0,1},{0,-1} };

struct position

{

int xx, yy, time;

};

int main()

{

int a, b;

cin >> a >> b;

char x[100][100];

int visited[100][100] = { 0 };

int sta\_x, sta\_y;

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

{

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

{

char ch;

cin >> ch;

x[i][j] = ch;

if (ch == '\*')

{

sta\_x = i;

sta\_y = j;

}

}

}

int flag = 0;

int time = 0;

position pp = { sta\_x, sta\_y ,0 };

queue<position>que;

visited[sta\_x][sta\_y] = 1;

que.push(pp);

while (!que.empty())

{

pp = que.front();

int cur\_x = pp.xx;

int cur\_y = pp.yy;

int ss = pp.time;

que.pop();

if (x[cur\_x][cur\_y] == '+')

{

flag = 1;

time = pp.time;

break;

}

for (int i = 0; i < 4; i++)

{

int new\_x = cur\_x + mov[i][0];

int new\_y = cur\_y + mov[i][1];

if (new\_x >= 0 && new\_x < a && new\_y >= 0 && new\_y < b)

{

if (x[new\_x][new\_y] != '1' && visited[new\_x][new\_y] != 1)

{

visited[new\_x][new\_y] = 1;

position tt = { new\_x,new\_y,ss + 1 };

que.push(tt);

}

}

}

}

cout << time;

}

四、踩方格

**#include** <iostream>

**#include** <cstring>

**using** **namespace** std;

int step;

int s[30][50];

int **f**(int i,int j, int rs)

{

**if** (rs == 0)**return** 1;

s[i][j] = 1;

int num = 0;

**if** (s[i][j - 1] == 0)num += **f**(i, j - 1, rs - 1);

**if** (s[i][j + 1] == 0)num += **f**(i, j + 1, rs - 1);

**if** (s[i + 1][j] == 0)num += **f**(i + 1, j, rs - 1);

s[i][j] = 0;

**return** num;

}

int **main**()

{

cin >> step;

**memset**(s, 0, **sizeof**(s));

cout << **f**(0, 25, step);

}

**#include** <iostream>

**using** **namespace** std;

int **main**()

{

int n;

cin >> n;

int a[3][25] = { 0 };

a[0][1] = 3;

a[1][1] = 2;

a[2][1] = 2;

**for** (int i = 2; i <= n; i++)

{

a[0][i] = a[0][i - 1] + a[1][i - 1] + a[2][i - 1];

a[1][i] = a[0][i - 1] + a[1][i - 1];

a[2][i] = a[0][i - 1] + a[2][i - 1];

}

cout << a[0][n];

}

**动态规划**

一、爬楼梯

**#include** <iostream>

**using** **namespace** std;

int **f**(int n)

{

**if** (n == 1)**return** 1;

**if** (n == 2)**return** 2;

**else** **return** **f**(n - 1) + **f**(n - 2);

}

int **main**()

{

cout<<**f**(20);

}

二、最长上升子序列

**#include** <iostream>

**#include** <algorithm>

**using** **namespace** std;

int **main**()

{

int n, a[1000],ans[1000];

cin >> n;

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

{

cin >> a[i];

ans[i] = 1;

}

int res=1;

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

{

**for** (int j = 0; j < i; j++)

{

**if** (a[i] > a[j])

{

ans[i] = **max**(ans[i], ans[j] + 1);

}

}

res = **max**(res, ans[i]);

}

cout << res;

}

三、数字三角形

**#include** <iostream>

**#include** <algorithm>

**#include** <cstring>

**using** **namespace** std;

int a[100][100];

int n;

int maxsum[100][100];

int **max\_sum**(int i, int j)

{

**if** (maxsum[i][j] != -1)**return** maxsum[i][j];

**if** (i == n - 1)**return** a[i][j];

**else**

{

maxsum[i][j] = **max**(**max\_sum**(i + 1, j), **max\_sum**(i + 1, j + 1)) + a[i][j];

**return** maxsum[i][j];

}

}

int **main**()

{

cin >> n;

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

{

**for** (int j = 0; j <= i; j++)

{

cin >> a[i][j];

}

}

**memset**(maxsum, -1, **sizeof**(maxsum));

cout << **max\_sum**(0, 0);

}

四、收服宠物小精灵

**#include** <iostream>

**#include** <algorithm>

**using** **namespace** std;

int N, M, K;

int cost[105][2] = { 0 };

int num[1005][505] = { 0 };

int **main**()

{

cin>> N>>M>>K;

**for** (int i = 1; i <= K; i++)

{

cin >> cost[i][0] >> cost[i][1];

}

**for** (int i = 1; i <= K; i++) {

**for** (int j = N; j >= cost[i][0]; j--) {

**for** (int t = M; t >= cost[i][1]; t--) {

num[j][t] = **max**(num[j][t], num[j - cost[i][0]][t - cost[i][1]] + 1);

}

}

}

int min\_num = 0;

**for** (int i = 0; i <= M; i++) {

**if** (num[N][i] ==num[N][M]) {

min\_num = i;

**break**;

}

}

cout << num[N][M] <<" "<< M - min\_num;

}