头文件+dij优化

#include<cstdio>

#include<iostream>

#include<queue>

#include<cmath>

#include<cstring>

#include<algorithm>

#define ll long long

#define pb(x) push\_back(x)

#define fir first

#define sec second

using namespace std;

const int INF=0x3f3f3f3f;

//freopen("data.in","r",stdin);

//freopen("data.out","w",stdout);

//ios\_base::sync\_with\_stdio(false);cin.tie(NULL);cout.tie(NULL);

/\* Dijkstar 算法+堆优化 使用优先队列优化，复杂度 O (E log E) \*/

const int INF=0x3f3f3f3f;

const int maxn=1110;

int d[maxn];

int n;

typedef pair<int,int>P;

struct node{

int to;int cost;

};

vector<node>g[maxn];

void dij(){

memset(d,0x3f,sizeof(d));

d[1]=0;

priority\_queue< P,vector<P>,greater<P> >que;

que.push(P(0,1));

while(!que.empty()){

P a=que.top();que.pop();

int u=a.second;

//if(d[u]<a.first) continue; optimize in cf 938d

for(int j=0;j<g[u].size();j++){

node e=g[u][j];

if(d[e.to]>e.cost+d[u]) {

d[e.to]=e.cost+d[u];que.push(P(d[e.to],e.to));

}

}

}}