网络流dinic

bool bfs()

{

int head=0,tail=1;

for(int i=0;i<=T;i++)h[i]=-1;

q[0]=0;h[0]=0;

while(head!=tail)

{

int now=q[head];head++;

for(int i=last[now];i;i=e[i].next)

if(e[i].v&&h[e[i].to]==-1)

{

h[e[i].to]=h[now]+1;

q[tail++]=e[i].to;

}

}

return h[T]!=-1;

}

int dfs(int x,int f)

{

if(x==T)return f;

int w,used=0;

for(int i=cur[x];i;i=e[i].next)

if(h[e[i].to]==h[x]+1)

{

w=f-used;

w=dfs(e[i].to,min(w,e[i].v));

e[i].v-=w;e[i^1].v+=w;

if(e[i].v)cur[x]=i;

used+=w;if(used==f)return f;

}

if(!used)h[x]=-1;

return used;

}

void dinic()

{

while(bfs())

{

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

cur[i]=last[i];

ans+=dfs(0,inf);

}

}

zkw费用流

bool spfa()

{

    memset(mark,0,sizeof(mark));

    for(int i=0;i<=T;i++)d[i]=-1;

    int head=0,tail=1;

    q[0]=T;mark[T]=1;d[T]=0;

    while(head!=tail)

    {

int now=q[head];head++;if(head==605)head=0;

for(int i=last[now];i;i=e[i].next)

if(e[i^1].v&&d[now]+e[i^1].c>d[e[i].to])

{

d[e[i].to]=d[now]+e[i^1].c;

if(!mark[e[i].to])

{

mark[e[i].to]=1;

q[tail++]=e[i].to;

if(tail==605)tail=0;

}

}

mark[now]=0;

    }

    return d[0]!=-1;

}

int dfs(int x,int f)

{

    mark[x]=1;

    if(x==T)return f;

    int w,used=0;

    for(int i=last[x];i;i=e[i].next)

if(d[e[i].to]==d[x]-e[i].c&&e[i].v&&!mark[e[i].to])

{

w=f-used;

w=dfs(e[i].to,min(w,e[i].v));

ans+=w\*e[i].c;

e[i].v-=w;e[i^1].v+=w;

used+=w;if(used==f)return f;

}

    return used;

}

void zkw()

{

    while(spfa())

    {

mark[T]=1;

while(mark[T])

{

memset(mark,0,sizeof(mark));

dfs(0,inf);

}

    }

}