**//MST, O(E log E)**

struct edge

{

int u,v,w;

bool operator < (const edge &p) const

{

return w>p.w;

}

};

edge eg;

int find\_par(int x)

{

if(x==par[x]) return x;

return par[x]=find\_par(par[x]);

}

void make\_union(int u,int v)

{

if(rank[u]<rank[v]) par[u]=find\_par(v);

else

{

par[v]=find\_par(u);

if(rank[u]==rank[v]) rank[u]++;

}

}

int mst(priority\_queue<edge>pq)

{

cnt=cost=0;

for(int i=1; i<=node; i++) par[i]=i, rank[i]=0;

while(!pq.empty())

{

if(cnt==node-1) break;

u=pq.top().u, v=pq.top().v, w=pq.top().w;

pu=find\_par(u), pv=find\_par(v);

if(pu!=pv)

{

cost+=w, cnt++;

make\_union(pu,pv);

}

pq.pop();

}

return cost;

}