Pastebin Link: <https://paste.ubuntu.com/p/jGg47GPwBQ/>

**int** src, snk, nNode, nEdge;  
**int** fin[MAXN], pre[MAXN], dist[MAXN];  
**int** cap[MAXE], cost[MAXE], nxt[MAXE], to[MAXE], from[MAXE];

**inline** **void** init(**int** \_src, **int** \_snk, **int** nodes){  
 memset(fin, -1, **sizeof**(fin));  
 nNode=nodes, nEdge=0;  
 src=\_src, snk=\_snk;  
}  
  
**inline** **void** addEdge(**int** u, **int** v, **int** \_cap, **int** \_cost){  
 from[nEdge]=u, to[nEdge]=v, cap[nEdge]=\_cap, cost[nEdge]=\_cost;  
 nxt[nEdge]=fin[u], fin[u]=nEdge++;  
 from[nEdge]=v, to[nEdge]=u, cap[nEdge]=0, cost[nEdge]=-(\_cost);  
 nxt[nEdge]=fin[v], fin[v]=nEdge++;  
}  
  
**bool** bellman(){  
 **int** iter, u, v, i;  
 **bool** flag=**true**;  
 memset(dist, 0x7f, **sizeof**(dist));  
 memset(pre, -1, **sizeof**(pre));  
 dist[src]=0;  
 **for**(iter=1; iter<nNode && flag; iter++){  
 flag=**false**;  
 **for**(u=0; u<nNode; u++){  
 **for**(i=fin[u]; i>=0; i=nxt[i]){  
 v=to[i];  
 **if**(cap[i] && dist[v]>dist[u]+cost[i]){  
 dist[v]=dist[u]+cost[i];  
 pre[v]=i;  
 flag=**true**;  
 }  
 }  
 }  
 }  
 **return** (dist[snk]<INF);  
}  
  
**int** mcmf(**int** &fcost){  
 **int** netflow, i, bot, u;  
 netflow=fcost=0;  
 **while**(bellman()){  
 bot=INF;  
 **for**(u=pre[snk]; u>=0; u=pre[from[u]]) bot=min(bot, cap[u]);  
 **for**(u=pre[snk]; u>=0; u=pre[from[u]]){  
 cap[u]-=bot;  
 cap[u^1]+=bot;  
 fcost+=bot\*cost[u];  
 }  
 netflow+=bot;  
 }  
 **return** netflow;  
}