标准图

```
int n, m;
struct edge {
edge() { u = 0; v = 0; }
edge(int u, int v, llint w) { this->u = u; this->v = v; this->w = w; }
    int u, v;
    llint w;
    friend bool operator<(const edge a, const edge b) { return  a.w < b.w; }
};
struct node {
    vector<edge> e;
    int val;
    int in_deg = 0;
    node() { e = *(new vector<edge>); val = 0; }
    inline friend bool operator < (node a, node b) { return a.in_deg < b.in_deg; }
};
//编号不连续时用map
map<int, node> nod;
node nod[MAXN];

void addedg(int u, int v, llint w)
{
    nod[v].in_deg++;
    nod[u].e.emplace_back(edge(u, v, w));
}
void addedg_u(int a, int b, int w)
{
    edge etemp = edge(a, b, w);
    nod[a].in_deg++;
    nod[b].in_deg++;
    nod[a].e.emplace_back(etemp);
    nod[b].e.emplace_back(edge(b, a, w));
}
```