```
范例: P3916、P5318
                                                                                                                     int n,m,g[100005][100005];//邻接矩阵 空间过大会RE、MLE
                                                                                                                     void Read(){
                                                                                                                          id Read(){
  int a,b;
  cin>>n>>m;
  for(int i=0;i<m;++i){
    scanf("%d%d",&a,&b);
    g[a][b]=1;</pre>
                                                                                            邻接矩阵
                                                                                                                     占用空间较大,一般不用
                                                                                                                    struct Edge{
int u,v;
                                                                                                                      ,,
vector<Edge> s;//边集数组
                                                                                                                     void Read(){
                                                                                                                          int a,b;
int a,b;
cin>>n> m;
for(int i=0;i<m;++i){
    scanf("%d%d",&ta,&tb);
    s.push_back((Edge){a,b});</pre>
                                                                                                                                                                     bool cmp(Edge x,Edge y)[//排序规则,先边的起始点升序,起始点相同 按结束点
if(x.u==y.u) return x.v<y.v;
else return x.u<y.u;
                                                                                            边集数组 🤆
                                                                                                                     排序方便,特殊地方使用 😊
                                                                                                                                                                     sort(s.begin(),s.end(),cmp);
                                                                                                                  vector<int> e[100005];//邻接表
for(int i=0;i<m;++i){
                                                                                                                        cin>>u>>v;
e[u].push_back(v);
                                                              图的存储
                                                                                                                       struct Edge(
int to,nxt;
)e[100005];//注意无向图的边数*2
int n,m.idx,h[100005],A[100005];
void Addedge(int u,int v)(//链式前向星
e[idx].to=v;
e[idx].to=v;
e[idx]-nxt=h[u];
h[u]=idx++;
                                                                                                                         }
void Read(){
    int a,b;
    cin>>n>m;
    memset(h,-1,sizeof(h));
    for(int i=0;i-m;++i){
        scanf("%d %d*,&a,&b);
        Addedge(a,b);//添加边无问图添加两条边
    }
图的存储和遍历
                                                                                            链式前向星 🙃
                                                                                                                        常用
                                                                                                                       P3916-链式前向星(反向)
                                                                                                             void dfs(int x){//深度优先搜索
                                                                                                                  vis[x]=true;
cout<<x<<" ";
                                                                                                                  邻接矩阵 🖯
                                                                                                               void dfs(int u,int v){
if(!A[v]){
                                                                                                                          // A[v]=u;
for(int p=h[v];p!=-1;p=e[p].nxt){ //或~p
int tt=e[p].to;
dfs(u,tt);
                                                              DFS
                                                                                   链式前向星 😑
                                                                                                        void dfs(int x){//深度优先搜索
vis[x]=true;
cout < < x < < ";
for(int i=o]: < e[x].size();i++){
int t=e[x][i];
if(lvis[t]) dfs(t);
}
                                                                                    邻接表 😊
                                                                                                            void bfs(int x){//广度优先搜索
queue<int> q;//使用STL
q.push(x);
                                                                                                                  cout < < x < < " "
                                                                                                                 cout<<x<<";
viskyl=true;
wint(t=q,front();
q.pop();
for(int i=1;<=n;i++){
if(visi[] && g(t[i)](
q.push(i);
cout<=i<<";
vis[i]=true;
}
                                                                                   邻接矩阵 🙃
                                                                                  链式前向星
                                                                                                        void bfs(int x){//广度优先搜索
queue<int> q;//使用STL
q.push(x);
cout<<x<<* *;
                                                                                                             BFS
                                                                                                                         int tt=e[t][i];
                                                                                                                         if(!vis[tt]){
    q.push(tt);
    cout < < tt < " ";
    vis[tt] = true;
                                                                                   邻接表 ⊝ }
```