结论

1、树上最远距离

树上到每个点距离最远的距离一定为树上一条直径的两个端点之一。

分别从两个端点开始 dfs 即可 $O(n)$ 求取每个点的树上最远点。

证明见 链接

或者可以考虑树形 dp第一次 dfs 维护每个节点子树方向上的最远距离和次远距离。

第二次 dfs 维护每个节点祖先方向上的最远距离，答案即为子树方向上的最远距离和祖先方向上的最远距离的较大者。

```
int dp[MAXN][3],hson[MAXN],dis[MAXN];
void dfs1(int u,int fa){
    dp[u][0]=dp[u][1]=dp[u][2];
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        if(v==fa)continue;
        dfs1(v,u);
        if(dp[v][0]+edge[i].w>dp[u][0]){
            hson[u]=v;
            dp[u][1]=dp[u][0];
            dp[u][0]=dp[v][0]+edge[i].w;
        }
        else if(dp[v][0]+edge[i].w>dp[u][1])
        dp[u][1]=dp[v][0]+edge[i].w;
    }
}
void dfs2(int u,int fa){
    dis[u]=max(dp[u][0],dp[u][2]);
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        if(v==fa)continue;
        if(v==hson[u])
        dp[v][2]=edge[i].w+max(dp[u][1],dp[u][2]);
        else
        dp[v][2]=edge[i].w+max(dp[u][0],dp[u][2]);
        dfs2(v,u);
    }
}
```