

inline void dfs(int rt,int fa){

siz[rt]=1;

for (int i=head[rt];i;i=e[i].nxt){

if (e[i].to==fa) continue;

dfs(e[i].to,rt);int v;

siz[rt]+=siz[e[i].to];

if (siz[e[i].to]>siz[maxsiz[rt]]) maxsiz[rt]=e[i].to;

if (siz[e[i].to]<=n/2) v=siz[e[i].to];

else v=dp[e[i].to][0];

if (dp[rt][0]<v) dp[rt][1]=dp[rt][0],dp[rt][0]=v,pos[rt]=e[i].to;

else if (dp[rt][1]<v) dp[rt][1]=v;

}

}

inline void redfs(int rt,int fa){

ans[rt]=1;

if (siz[maxsiz[rt]]>n/2) ans[rt]=(siz[maxsiz[rt]]-dp[maxsiz[rt]][0]<=n/2);

else if (n-siz[rt]>n/2) ans[rt]=(n-siz[rt]-dpp[rt]<=n/2);

for (int i=head[rt];i;i=e[i].nxt){

if (e[i].to==fa) continue;int v;

if (n-siz[rt]>n/2) v=dpp[rt];

else v=n-siz[rt];

dpp[e[i].to]=max(dpp[e[i].to],v);

if (pos[rt]==e[i].to) dpp[e[i].to]=max(dpp[e[i].to],dp[rt][1]);

else dpp[e[i].to]=max(dpp[e[i].to],dp[rt][0]);

redfs(e[i].to,rt);

}

}

signed main(void){

n=read();

for (int i=1;i<=n-1;i++){

int u=read(),v=read();

add\_edge(u,v),add\_edge(v,u);

}

dfs(1,0),redfs(1,0);

for (int i=1;i<=n;i++) printf("%d ",ans[i]);

return 0;

}