Pastebin Link: <http://paste.ubuntu.com/25932495/>

#define NODES 100105  
*/\**  
 *\* Assumes 0 based indexing of tree*  
 *\* Use clearAll for multiple test cases*  
 *\* Call proc() to calculate the tree's diametre,*  
 *\* nodes at the ends of the diameter,*   
 *\* and nodes at the centre of the diameter*   
*\*/*  
**struct** Tree{  
 vector <**int**> adj[NODES], centre; *//Adjacency List and nodes at centre of diameter respectively*  
 vector <**int**> adjw[NODES]; *//Adjacence List of edge weights*  
 **int** dis[3][NODES];  
 **int** far[NODES]; *//Distance of farthest node from each node*  
 **int** end1, end2, n, dia; *//n=number of vertices of tree*  
 *//end1 and end2 are the nodes at the diameter;*  
 *//dia is the diametre of the tree*  
 **void** clearAll(){  
 centre.clear();  
 **for**(**int** i=0; i<NODES; i++){  
 adj[i].clear();  
 adjw[i].clear();  
 }  
 }  
 **void** addEdge(**int** u, **int** v, **int** w){ *//Adds edge between nodes u, v with weight w*  
 adj[u].push\_back(v);  
 adjw[u].push\_back(w);  
 adj[v].push\_back(u);  
 adjw[v].push\_back(w);  
 }  
 **void** dfs(**int** nd, **int** ind){  
 **int** v;  
 **for**(**int** i=0; i<adj[nd].size(); i++){  
 v=adj[nd][i];  
 **if**(dis[ind][v]==-1){  
 dis[ind][v]=adjw[nd][i]+dis[ind][nd];  
 dfs(v, ind);  
 }  
 }  
 }  
 **int** findmax(**int** ind){  
 **int** mx=0;  
 **for**(**int** i=1; i<n; i++){  
 **if**(dis[ind][mx]<dis[ind][i]) mx=i;  
 }  
 **return** mx;  
 }  
 **void** findDia(){  
 memset(dis, -1, **sizeof**(dis));  
 dis[0][0]=0;  
 dfs(0, 0);  
 end1=findmax(0);  
 dis[1][end1]=0;  
 dfs(end1, 1);  
 end2=findmax(1);  
 dis[2][end2]=0;  
 dfs(end2, 2);  
 dia=dis[1][end2];  
 }  
 **void** findFarthestNodes(){  
 **for**(**int** i=0; i<n; i++){  
 far[i]=max(dis[1][i], dis[2][i]);  
 }  
 }  
 **int** findMinFar(){  
 **int** mn=far[0];  
 **for**(**int** i=1; i<n; i++){  
 mn=min(mn, far[i]);  
 }  
 **return** mn;  
 }  
 **void** findCentres(){  
 **int** mn=findMinFar();  
 **for**(**int** i=0; i<n; i++){  
 **if**(far[i]==mn){  
 centre.push\_back(i);  
 }  
 }  
 }  
 **void** proc(){  
 findDia();  
 findFarthestNodes();  
 findCentres();  
 }  
};