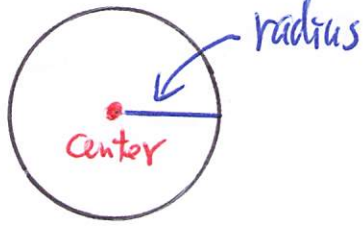
Tree Eccentricities, diameters, and radii

KunMaoChao

@2019
@2021
@2024



$G=(V,E,\omega)$

ω: positive edge weights

$U \subseteq V$

$$\underline{D_G(v,U) = \max_{u \in U}\{d_G(v,u)\}}$$

The eccentricity of v is the maximum of the distance to any vertex in the graph, i.e. $\max_{u \in V}\{d_G(v,u)\}$ or $D_G(v,V)$.

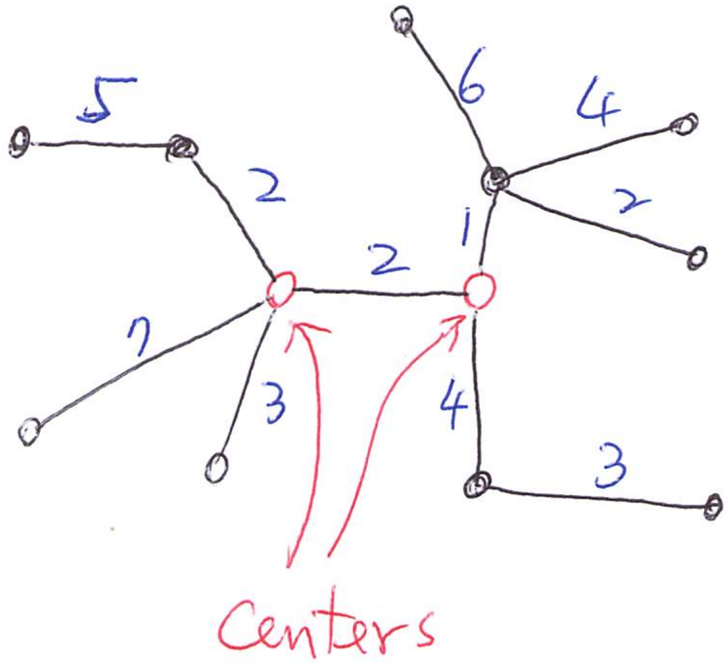
Diameter: $\max_{v \in V}\left\{\max_{u \in V}\{d_G(v,u)\}\right\} = \max_{v,u} d_G(v,u)$

also referred to as a path

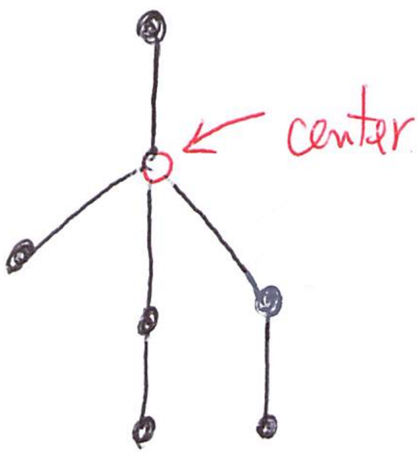
Radius: $\min_{v \in V}\left\{\max_{u \in V}\{d_G(v,u)\}\right\}$

① Center: the vertex with the minimum eccentricity

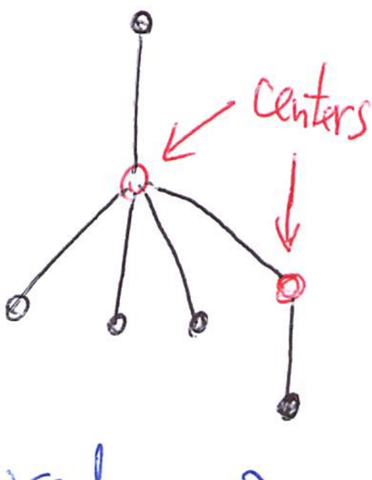
Kuen-Mao Chao @2019
@2021



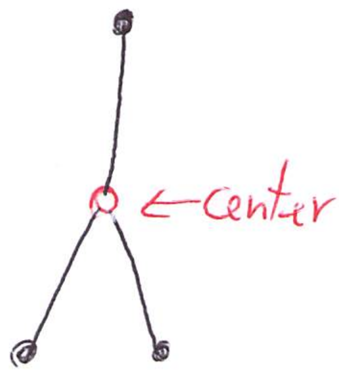
radius = 9
diameter = 16



radius = 2
diameter = 4



radius = 2
diameter = 3

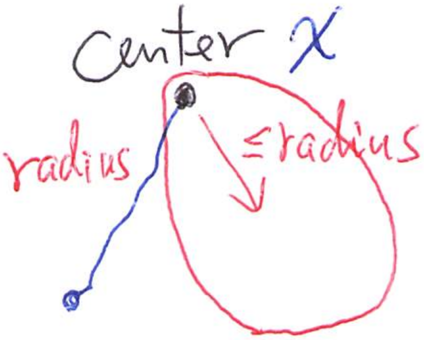


radius = 1
diameter = 2

Kun-Mao Chao @2019

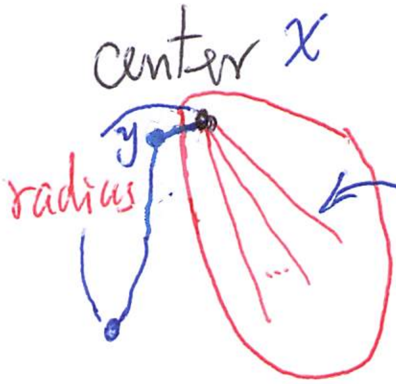
For an unweighted tree T,

$$2 \times \text{radius} - 1 \le \text{diameter} \le 2 \times \text{radius}.$$



$$d_T(u,v) \le d_T(u,x) + d_T(x,v)$$
$$\le \text{radius} + \text{radius}$$
$$= 2 \times \text{radius}$$

$\Rightarrow$ diameter $\le 2 \times$ radius
$\left(\max_{u,v} d_T(u,v)\right)$



At least one branch $\ge$ radius -1
(Otherwise, we have a smaller radius by selecting y as the center.)

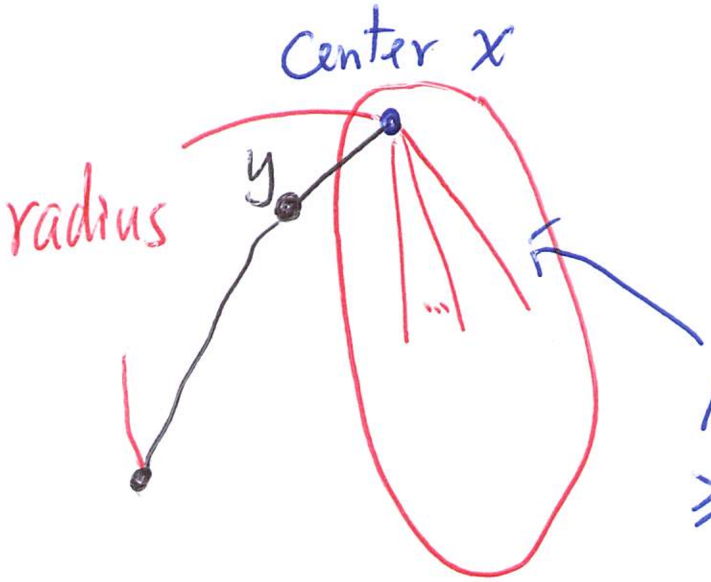
$$\Rightarrow \text{diameter} \ge \text{radius} + \text{radius} - 1$$
$$= 2 \times \text{radius} - 1$$

Kun-Mao Chao @2019
@2021

For positive weighted tree T,

$$2 \times \text{radius} - \max_{e \in E}\{w(e)\} \le \underline{\text{diameter} \le 2 \times \text{radius}}$$

$$d_T(u,v) \le d_T(u,\underset{\text{center}}{x}) + d_T(x,v) \le 2 \times \text{radius}$$



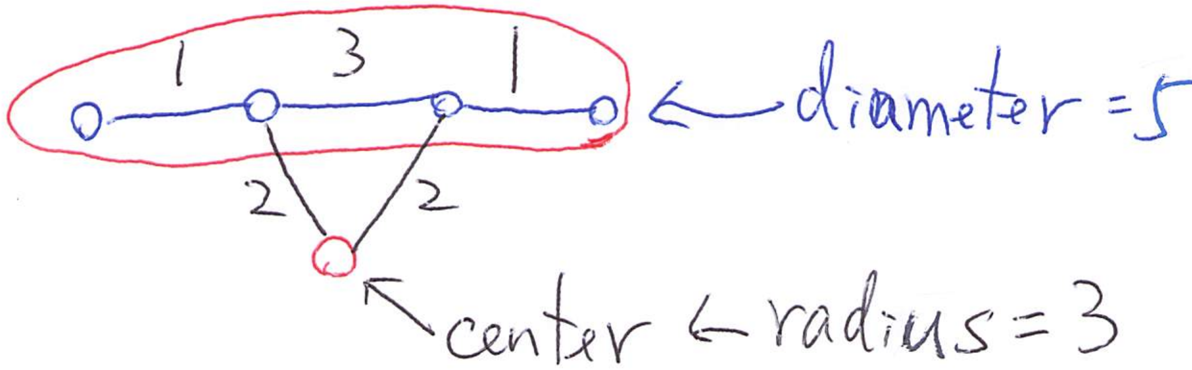
At least one branch $\ge \text{radius} - w((x,y)) \ge \text{radius} - \max_{e \in E}\{w(e)\}$

(Otherwise, we have a smaller radius by selecting $\underline{y}$ as the center.)

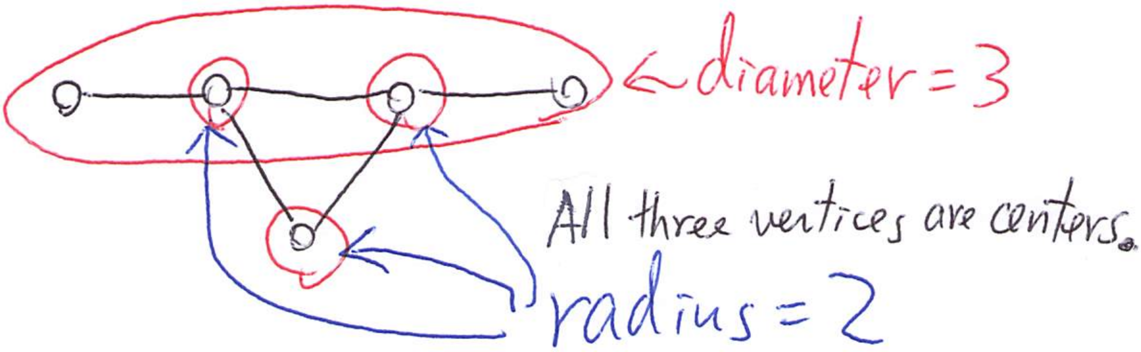
$$\Rightarrow \text{diameter} \ge \text{radius} + \text{radius} - \max_{e \in E}\{w(e)\}$$
$$= 2 \times \text{radius} - \max_{e \in E}\{w(e)\}$$

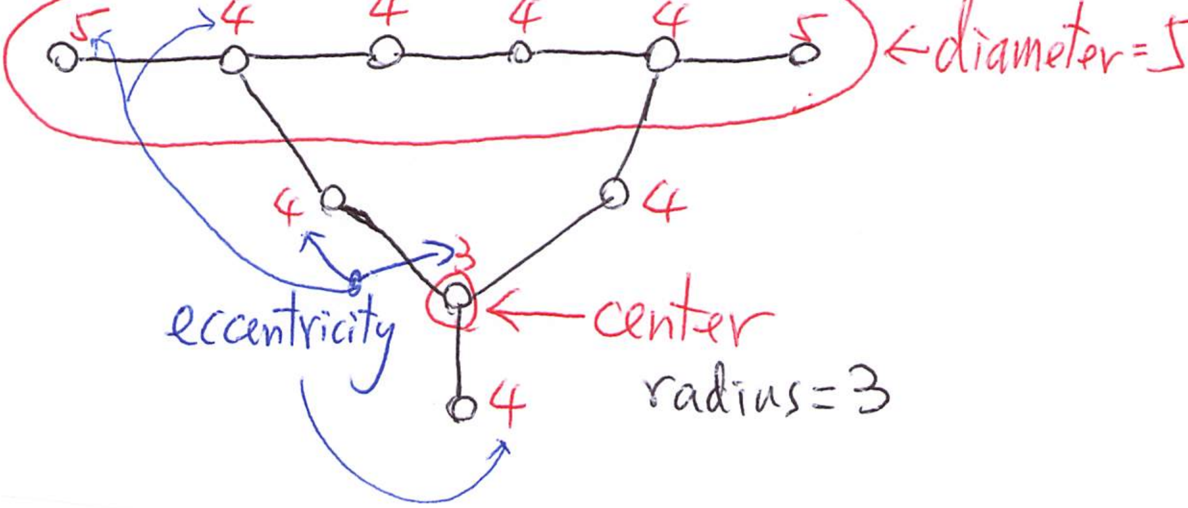
Examples (centers not in the diameter) Kun-Mao Chao @2021

weighted graph:



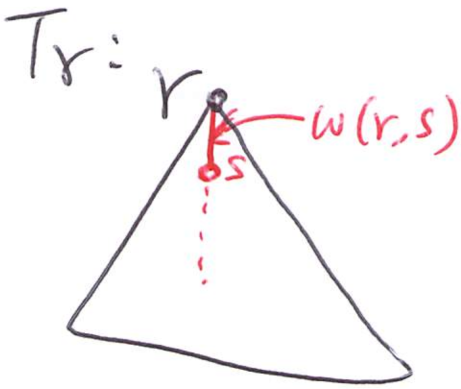
unweighted graph:





Kun-Mao Chao @2019

T_r : a subtree rooted at r

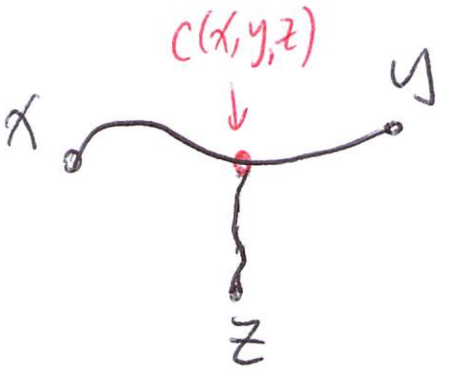


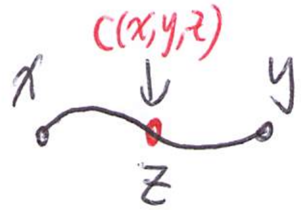
$$D_{T_r}(r, V(T_r)) = \max_{s \in child(r)} \{ D_{T_r}(s, V(T_s)) + w(r,s) \}$$

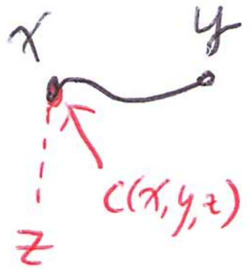
We can root the tree at any given vertex v and compute v's eccentricity in linear time by the above recurrence relation. (bottom-up Postorder traversal or recursive algorithm)

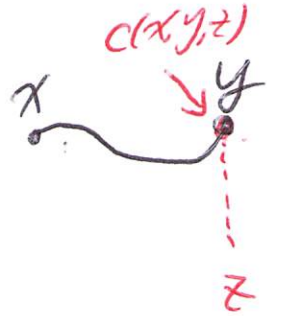
Kun-Mao Chao @2019

$$C(x,y,z) = SP_T(x,y) \cap SP_T(x,z) \cap SP_T(y,z)$$

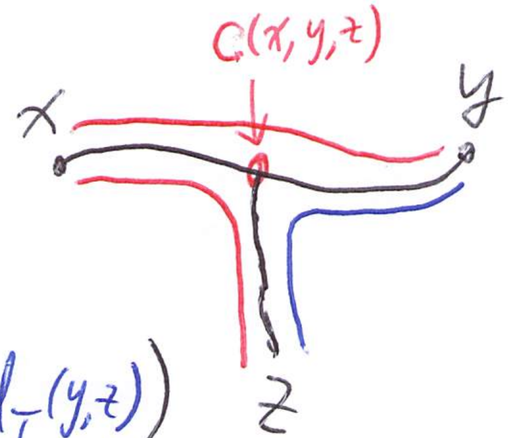






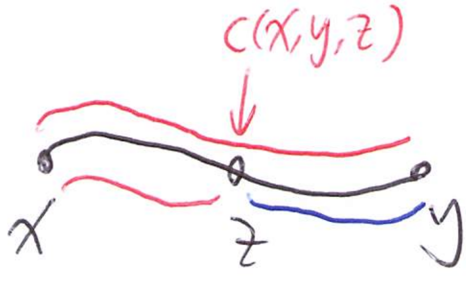


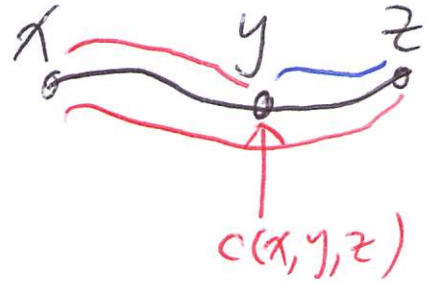
$$d_T(x, C(x,y,z))$$

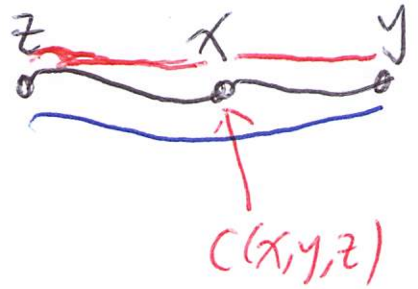


$$= \frac{1}{2}\left(d_T(x,y) + d_T(x,z) - d_T(y,z)\right)$$

$$= \frac{1}{2}\left(d_T(x,y) + d_T(x,z) + d_T(y,z)\right) - d_T(y,z)$$







Kun-Mao Chao @2019
@2021
@2024

$SP_T(v_1, v_2)$: diameter

r: a vertex on $SP_T(v_1, v_2)$

diameter

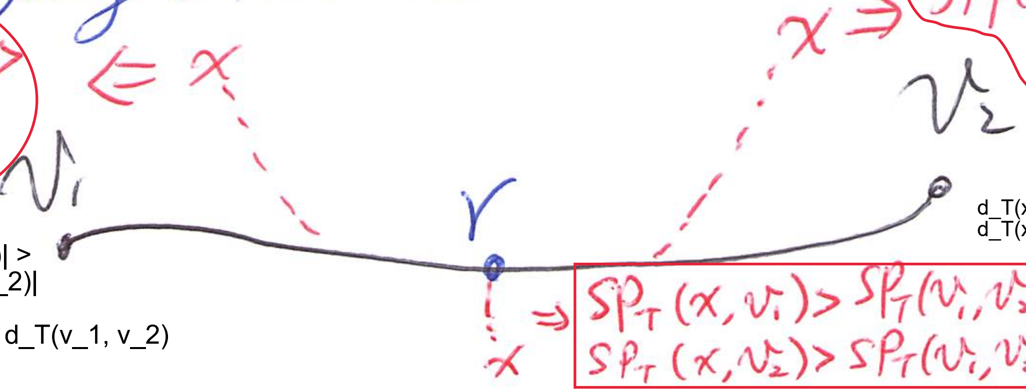
For any vertex x, $d_T(x, r) \le \max\{d_T(r, v_1), d_T(r, v_2)\}$.

pf. If $d_T(x, r) > \max\{d_T(r, v_1), d_T(r, v_2)\}$, then $SP_T(x, v_1)$ or $SP_T(x, v_2)$ is a path longer than the diameter.

|SP_T(x, v_1)| >
|SP_T(v_1, v_2)|
or
d_T(x, v_1) > d_T(v_1, v_2)

$SP_T(x, v_1) > SP_T(v_1, v_2)$

$SP_T(x, v_2) > SP_T(v_1, v_2)$



|SP_T(x, v_2)| >
|SP_T(v_1, v_2)|
or
d_T(x, v_2) > d_T(v_1, v_2)

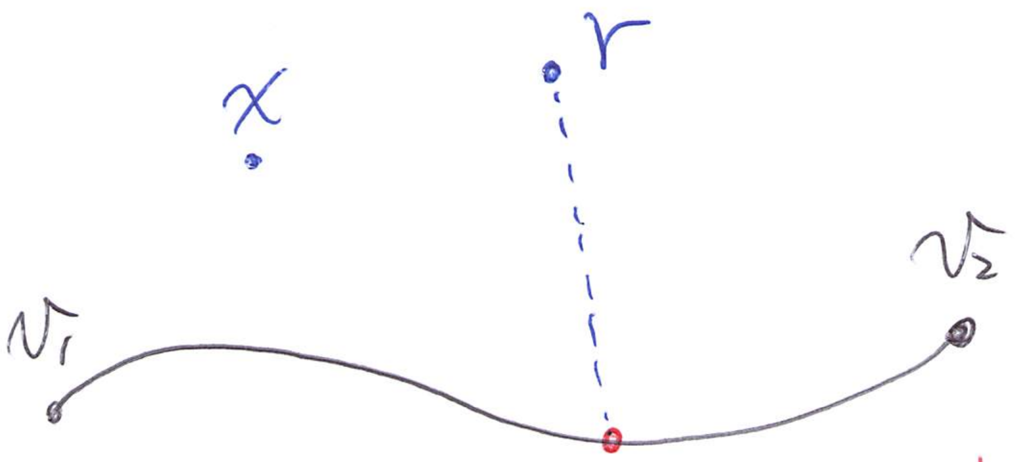
d_T(x, v_1) > d_T(v_1, v_2)
d_T(x, v_2) > d_T(v_1, v_2)

$\Rightarrow$ $\boxed{\begin{array}{l} SP_T(x, v_1) > SP_T(v_1, v_2) \\ SP_T(x, v_2) > SP_T(v_1, v_2) \end{array}}$

Kun-Mao Chao @2019

$SP_T(v_1, v_2)$: diameter

r : a vertex not on the diameter



$u = c(r, v_1, v_2)$

Wlog. let $d_T(u, v_1) \geq d_T(u, v_2)$.

$$d_T(x,r) \leq d_T(x, \underset{\text{on the diameter}}{u}) + d_T(u,r)$$
$$\leq d_T(v_1,u) + d_T(u,r)$$
$$= d_T(v_1, r)$$

For any vertex, either on the diameter or not, one of the endpoints of a diameter must be the farthest vertex.

8

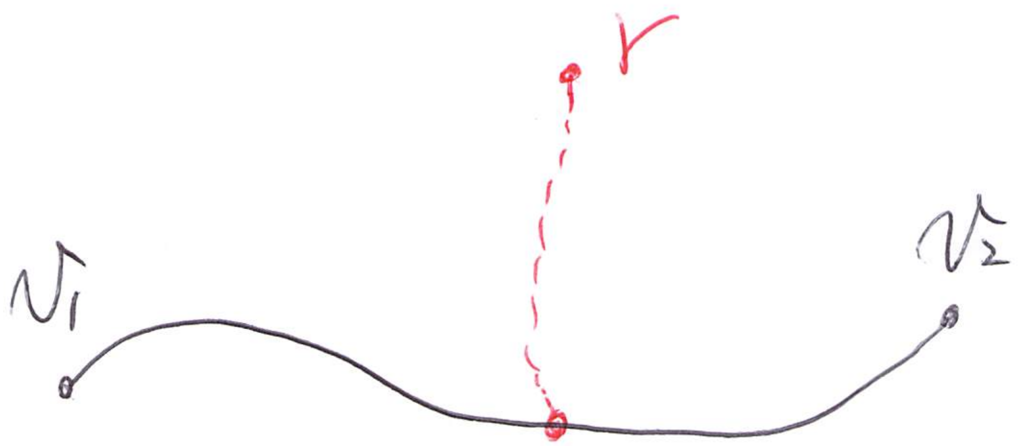
Kan-Mao Chen @2019

$SP_T(v_1, v_2)$: diameter

r : any vertex

v_3 : the vertex farthest to r

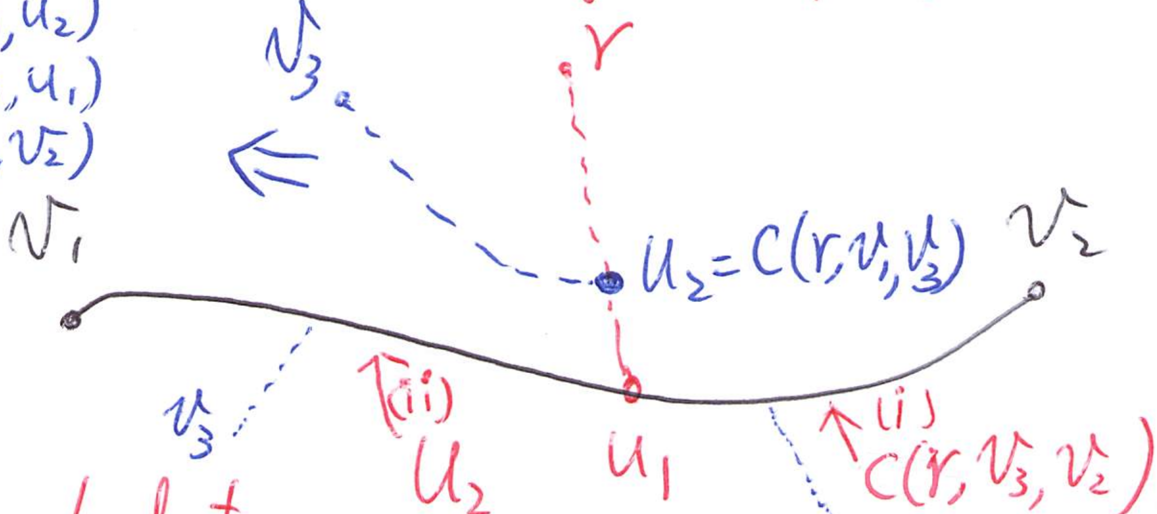
$\Rightarrow$ v_3 must be an endpoint of a diameter.



$u_1 = c(r, v_1, v_2)$

W.l.o.g., let $d_T(u_1, v_1) \geq d_T(u_1, v_2)$.

$d_T(v_1, v_3)$
$= d_T(v_1, u_2) + d_T(u_2, v_3)$
$= 2 d_T(v_1, u_2)$
$> 2 d_T(v_1, u_1)$
$\geq d_T(v_1, v_2)$



A contradiction.

$\Rightarrow$ u_2 is on $SP_T(v_1, u_1)$.

v_3

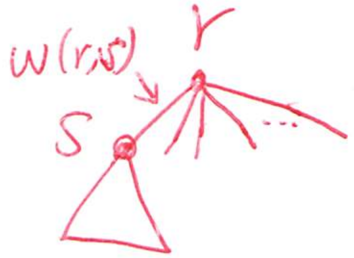
If $c(r, v_3, v_2) \neq u_1$, then (i) $SP_T(v_1, v_3)$ is a diameter.

Otherwise, (ii) $SP_T(v_3, v_2)$ is a diameter.

Kun-Mao Chao @2019

Algorithm: ECCENT(T_r)
Input: A tree $T_r = (V, E, w)$ rooted at r.
Output: The eccentricity of r in T_r.

1: **if** r is a leaf **then**
    return 0;
2: **for** each child s of r **do**
    compute ECCENT(T_s) recursively;
3: **return** $\max_{s \in child(r)}\{\text{ECCENT}(s) + w(r,s)\}$.

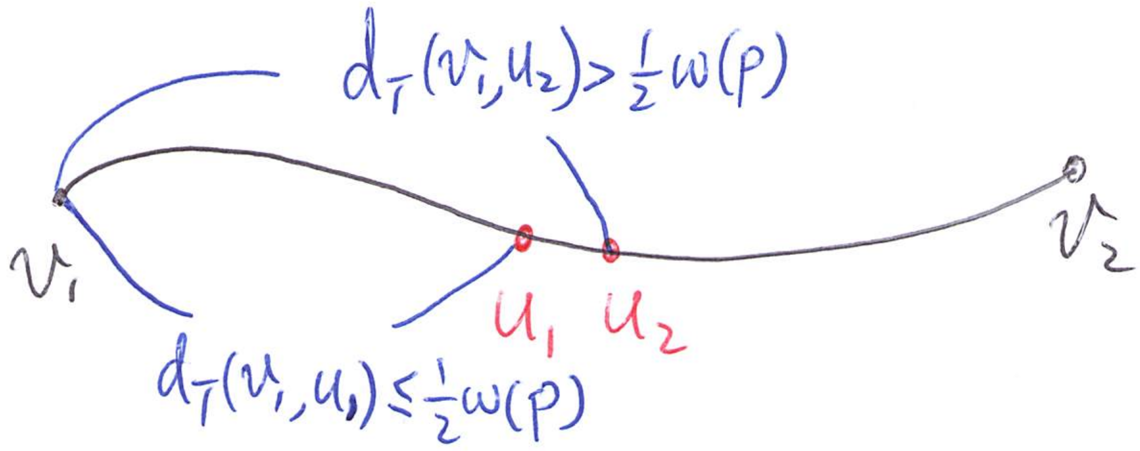


Algorithm: TREEDIAMETER
Input: A tree $T = (V, E, w)$.
Output: The diameter of T.

1: Root T at an arbitrary vertex r.
2: Use ECCENT to find the farthest vertex v to r.
3: Root T at v.
4: Use ECCENT to find the eccentricity of v.
5: Output the eccentricity of v as the diameter of T.

an endpoint of a diameter

$P = SP_T(v_1, v_2)$: diameter

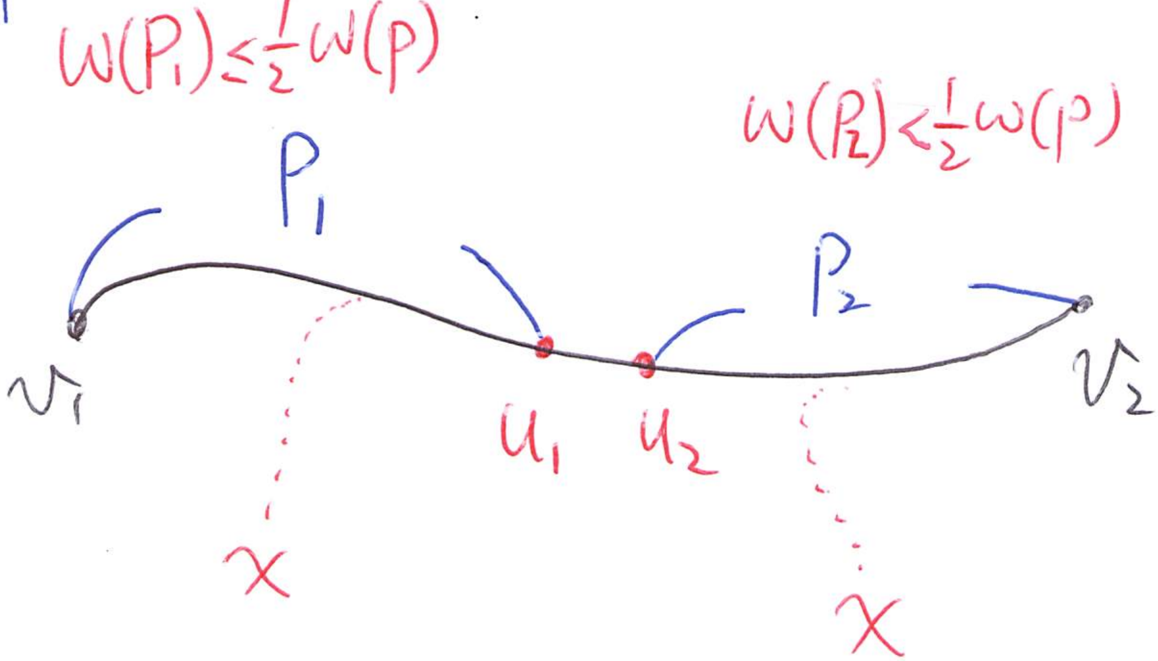


The eccentricity of u_1 is $d_T(u_1, v_2)$.

The eccentricity of u_2 is $d_T(u_2, v_1)$.

Kun-Mao Chao @2019

P:



For any vertex connected to P at a vertex in P_1, $d_T(x, v_2) > \underline{d_T(u_1, v_2)}$.

u_1's eccentricity

For any vertex connected to P at a vertex in P_2, $d_T(x, v_1) > \underline{d_T(u_2, v_1)}$.

u_2's eccentricity

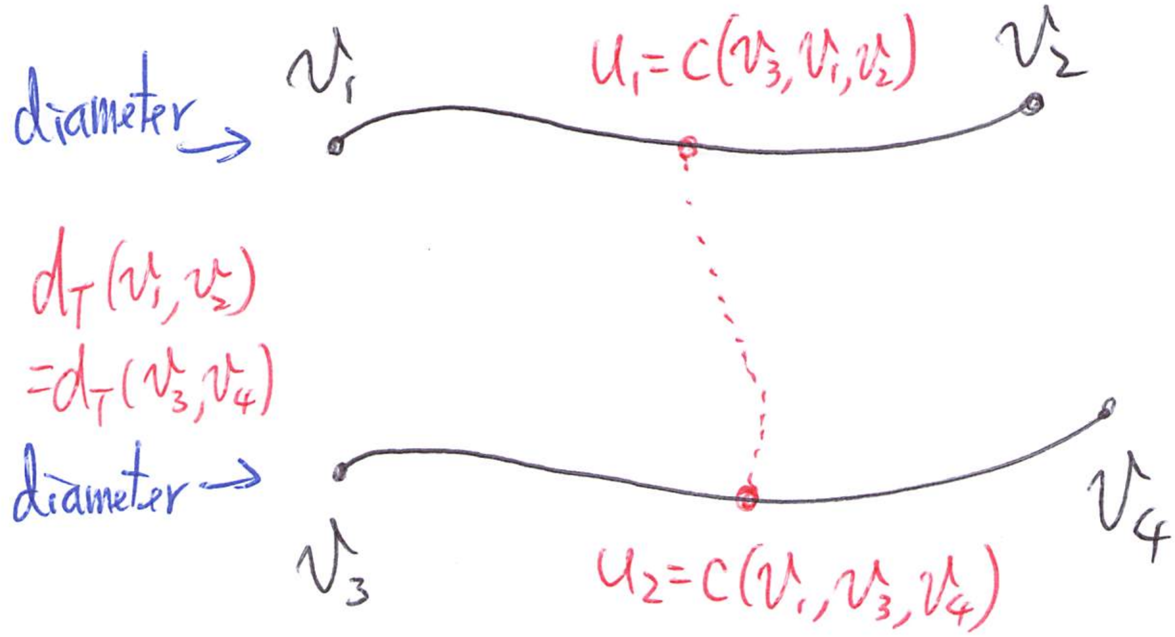
$\Rightarrow \min\{d_T(u_1, v_2), d_T(u_2, v_1)\}$ is the radius.

One of them must be a center.

We have proved the following theorem. Kun-Mao Chao @2019

☆ Thm. The diameter, radius, and center of a tree can be computed in linear time.

Two diameters of a tree cannot be disjoint.



$$d_T(v_1, v_4) + d_T(v_2, v_3)$$

$$= d_T(v_1, v_2) + d_T(v_3, v_4) + \overbrace{2d_T(u_1, u_2)}^{>0}$$

$$> 2\,d_T(v_1, v_2)$$

$$\Rightarrow \underline{d_T(v_1, v_4)} > d_T(v_1, v_2) \text{ or } \underline{d_T(v_2, v_3)} > d_T(v_1, v_2)$$

One of them is longer than the diameter. A contradiction!

Kun-Mao Chao @2019

All diameters of a tree share at least one common vertex.