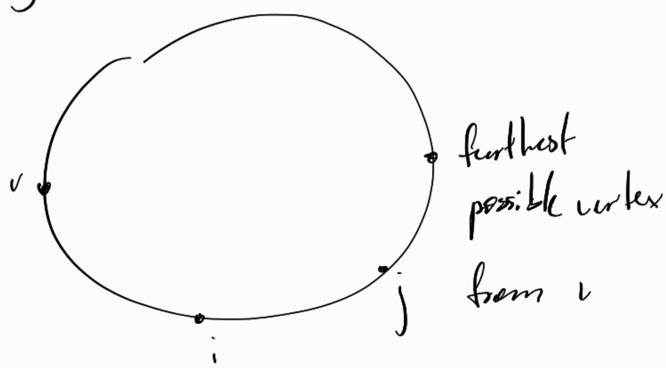
1.

· first, we shall prove that T is in fact a path

· for a contradiction, suppose it is not a path

· if $uv \in E(T)$, then $dist_T(u,v) = dist_{C_n}(u,v)$ (o/w just make it so and get a better sol)

· let v be a vertex of $\deg \geq 3$



· by PHP, there exist 2 neighbors i, j of v s.t. they lie on the same semicircle given by v and its antipodal vertex

· let j be further from v than i

· we claim that moving j to be a child of i decreases the total weight of T

· the result is still a tree

· has the distortion of the tree increased?

· any path that used the edge vj can now use $v \leftrightarrow i \leftrightarrow j$

· on a cycle, this path is as long as $v \leftrightarrow j$

· however, we decreased the total weight of a tree, why?

- an edge of a tree must have length $\geq dist_{C_n}(u,v)$
- thus we removed at least $dist_{C_n}(v,j)$ from the weight but added less than that

$\rightarrow \lightning$

- thus T is a path
- its endpoints are at dist $n-1$ in T but at dist 1 in C_n

2.

- using dynamic programming, we will compute the following 2 fcts

  $f, g : V \times [k] \times \binom{n}{2} \rightarrow \mathbb{R}^+$

- $f(r, k, d)$: let T_r be the subtree of T rooted in r. This is the cost in $G[T_r]$ s.t. all centers are in T_r, there is at most k of them and the dist from r to its nearest center is $\leq d$.
- note that the last param is some inter-vertex dist, there's $\leq \binom{n}{2}$ of them
- $g(r, l, d)$: same as f but d is the closest distance from r to any center, including those in $G \setminus T_r$ (i.e., outside T_r)
- for a vertex u, let $N_i(u)$ be the i^{th} closest vertex of G to u
  - in case of ties, we prioritize vertices in T_u
  - $N_1(v) = v$ for $\forall v \in V$

- $d_i(u)$ is $\text{dist}(u, N_i(u))$
- let u be a leaf
- for $\forall i \in [n]$ we have
  - $f(u, 1, d_i(u)) = g(u, 1, d_i(u)) = 0$ as 1 indicates that we've opened a center in u.
  - $g(u, 0, d_i(u)) = d_i(u)$ by def.
- let u be an internal vertex, ℓ its left child and r its right child
- how do we compute $f(u, k, 0)$ for any k? 0 indicates that u is a center
- let ℓ_i be an index s.t. $N_{\ell_i}(u) = \ell$, same for r_i
- $f(u, k, 0) = \min_{k_\ell + k_r = k-1} f(\ell, k_\ell, \text{dist}(u, \ell)) + f(r, k_r, \text{dist}(u, r))$

  since we opened u as a center, the dist from ℓ to its nearest center is $\leq \text{dist}(u, \ell)$
- same for $g(u, k, 0)$

- $f(u, k, d_i(u))$ for $i \in [?, n]$?
- who is $N_i(u)$ and where is it?
- if it is outside T_u, there's nothing to do

  $\to f(u, k, d_i(u)) = f(u, k, d_{i-1}(u))$
- let $v \in N_i(u)$ and suppose $v \in T_\ell$ (or T_r symmetrically)
- let ℓ_i be an index s.t. $N_{\ell_i}(\ell) = v$ and r_i similarly

- at this point we want to cover u by something at distance $\leq d_i(u)$
- we defined v to be this center, so we need to express this fact
- we will ask recursion to give a solution that is as good as using v as a center
- for the right subtree T_r, $v \notin T_r$, so we have to ask g

$$\rightarrow f(u, k, d_i(u)) = \min \Big\{ f(u, k, d_{i-1}(u)),$$

$$d_i(u) + \min_{k_l + k_r = k} \Big\{ f(\ell, k_1, d_{\ell_i}(\ell)) + g(r, k_2, d_{r_i}(r)) \Big\} \Big\}$$

↑ cost of covering u by v

remember, we can always pick v as the solutions, so these recursive sols are guaranteed to be $< \infty$

- it remains to compute g for internal vertices
- i.e. what is $g(u, k, d_i(u))$?
- as the center which covers u can be in $V \setminus T_u$, we recurse appropriately
- using the same notation as previously:

$$g(u, k, d_i(u)) = \min \Big\{ f(u, k, d_i(u)), \min_{k_\ell + k_r = k} \big\{ g(\ell, k_\ell, d_{\ell_i}(u)) + g(r, k_r, d_{r_i}(u)) \big\} \Big\}$$

- the solution is given by $\min_{i \in [k]} g(r, i, d_n(r))$ where r is the root of T
- there are $n^2 \cdot k \cdot n^2$ parameters, each can be computed in poly time

$\rightarrow$ poly alg

- to finish the problem, we embed the input $(V, dist)$ into a tree using the embedding on the lecture
- we compute opt sol on the tree
- how expensive is our sol S (in expectation)?
- let S^* be an opt sol
- $$\mathbb{E}[cost(S)] = \sum_{u \in V} \mathbb{E}[dist(u,S)] \leq \sum_{u \in V} O(\log n)\, dist(u,S^*) = O(\log n)\, OPT \quad \square$$