Online Steiner tree

Input: $G=(V,E)$ undirected, weighted. A set $R \subseteq V$ of "required vertices".

Output: A tree of minimum weight that contains all the vertices in R.

Online setting: G & the weights are known in advance. R is not known. Each request is a required vertex. If the newly arrived vertex is not covered by the existing tree (the algorithm maintains a tree containing all the required vertices that have arrived), add new edge(s) to cover it. The objective is to minimize the total cost of the tree.

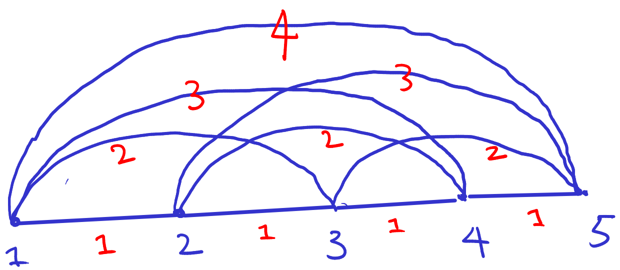
Observations: Enough to design an algorithm for the metric version of the problem.

Metric Steiner tree: Complete graph, weight function satisfies triangle inequality.

Greedy algorithm A:

- $V_T \leftarrow \phi$, $E_T \leftarrow \phi$
- for $i = 1$ to $|R|$
  - Let $a_1, \cdots, a_{i-1}$ be the required vertices that have arrived so far. Let a_i arrive in the current iteration.
  - Let $j^* = \arg\min \{ w(a_i, a_j) : j = 1, \cdots, i-1 \}$
  - $E_T \leftarrow E_T \cup \{ (a_i, a_{j^*}) \}$.
  - $V_T \leftarrow V_T \cup \{ a_i \}$.

Correctness: Immediate.

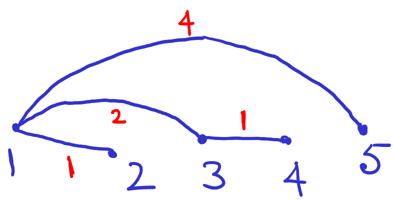


$R = \{1, 2, 3, 4, 5\}$

$OPT = 4$.

An optimal Steiner tree: 1—2-3-4-5

$\boxed{\text{output of the greedy on sequence } 1\ 5\ 3\ 2\ 4 :}$



Cost $= 8$.

<u>Exercise</u>: Extend this example to create inputs

for the online problem for which cost of the computed solution $= \Omega(\log n) \cdot OPT$.

no. of vertices

Theorem: A is $O(\log n)$-competitive.

proof: Let $s = |R| - 1$ and $e_1, \dots, e_s$ be the edges selected by A, ordered in non-increasing order of weights, i.e.,

$$w(e_1) \geq w(e_2) \geq \dots \geq w(e_s).$$

Claim: $w(e_i) \leq 2 \cdot \frac{OPT}{i}$.

Let us assume the claim and complete the proof first. Then we will prove the claim.

Cost of the computed solution

$$= \sum_{i=1}^{s} w(e_i)$$

$$\leq \sum_{i=1}^{s} \frac{2 \cdot OPT}{i} \quad \text{[From the claim]}$$

$$= 2 \cdot OPT \cdot \sum_{i=1}^{s} \frac{1}{i} = 2 \cdot OPT \cdot H_s$$

$$= O(\log n) \cdot OPT.$$