9-19-19

① Lemma 4.4. If all edge weights are distinct, there is a unique MST.

Proof: Assume for contradiction, $\exists$ T & T'

and both are MSTs and $T \neq T'$

e = min cost edge $\{$ edge is in one of T or T' but not in both $\}$.

without loss of generality, let $e \in T$

Add e to T' ; cycle in $e \cup \{T'\}$

examine all edges of this cycle: among the edges of this cycle, find an edge of smallest cost among those in T' but not in T. Let it be e'

$\text{Cost}(e') > \text{Cost}(e)$

delete e' from $e \cup \{T'\}$, we get a spanning tree T''

$T'' = T' \cup \{e\} - \{e'\}$

$\text{Cost}(T'') < \text{Cost } T'$, T' is not an MST

↯

No cycles formed if MST is unique

② Algorithm for MST (SynchGHS)

Initially, each node (process) is a component by itself

Each component has a leader and a component id.

Repeat until no MWOE for this exists.

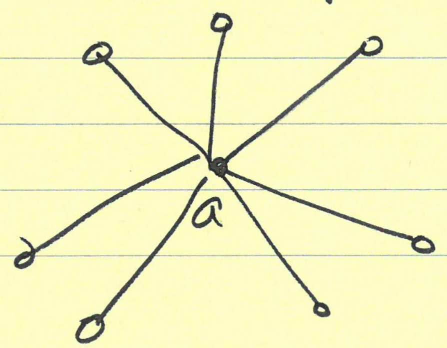
- Find min weight outgoing edge. (MWOE)
- Combine with component with other ~~edg~~ end of MWOE.
- get new component id, new leader id

Finding MWOE of a component.

Leader broadcasts an initiate (component id, leader id, ...) on the tree spanning this component

1 Each node gets this message.
2 - - - finds local candidate for MWOE
3 Use convergecast to find MWOE for the entire component

Step 2:



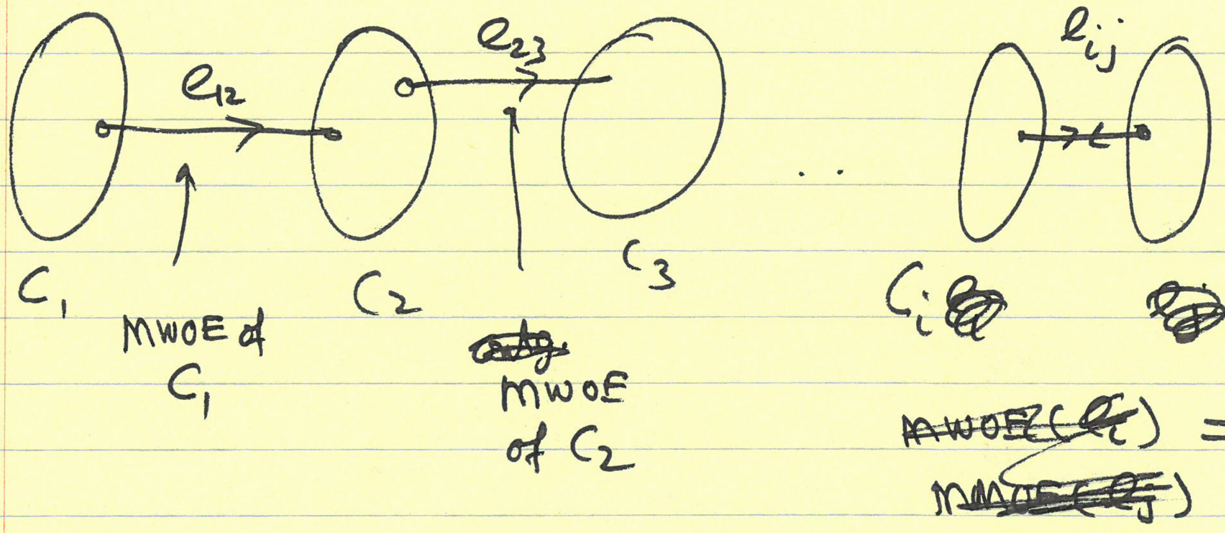
For node a:
Find edges incident on a that may be outgoing.
~~edge~~

③ ~~Sort~~

Sort these "unknown" edges in increasing cost.

Send test message, (one by one), wait for a reply

- accept (outgoing edge)
- reject (not an outgoing edge)



~~MWOE(C_i)~~ = ~~MWOE(C_j)~~

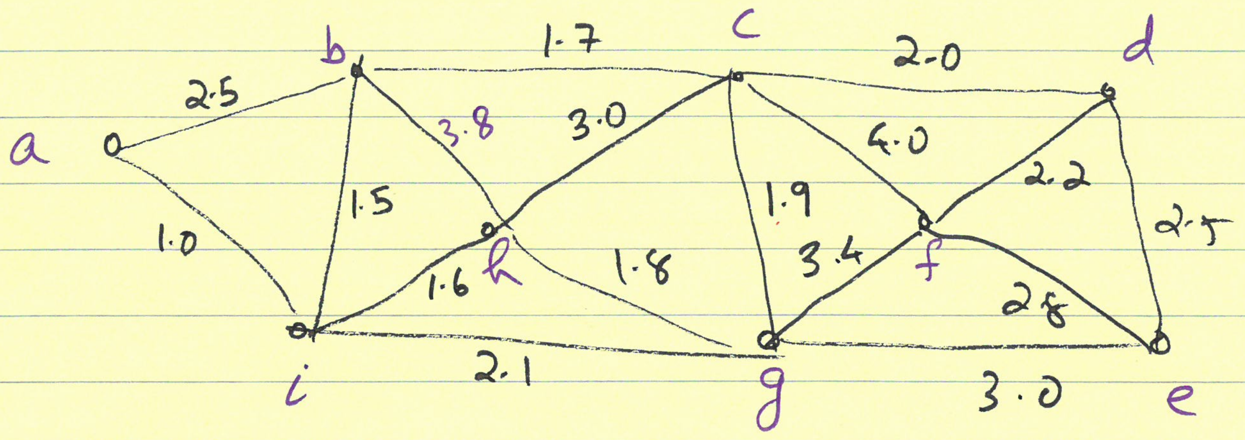
e_{ij} = MWOE(C_i) = MWOE(C_j)

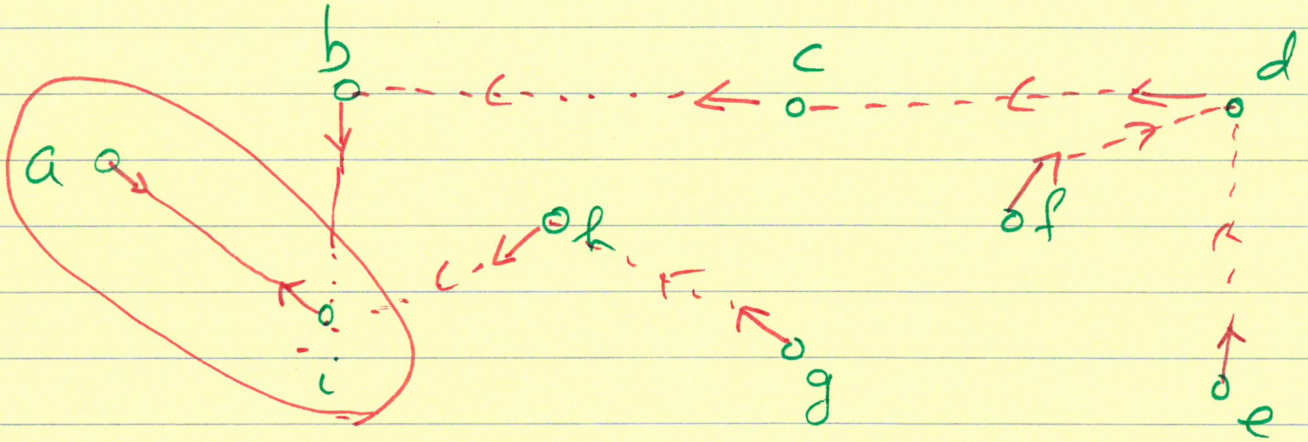
e_{ij} is a CORE edge

C_i & C_j merge to form a new (Combined) Component with id = weight(e_{ij})

Leader of $C_i \cup C_j \cup \{e_{ij}\}$ is one of the two end vertices of e_{ij} (the one with smaller id)

Example 1

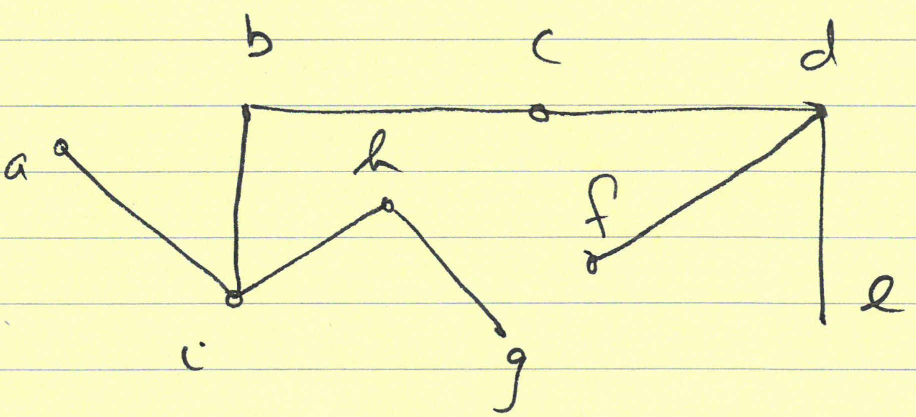




New Component C
id = (1.0, a, i)
leader of C = a

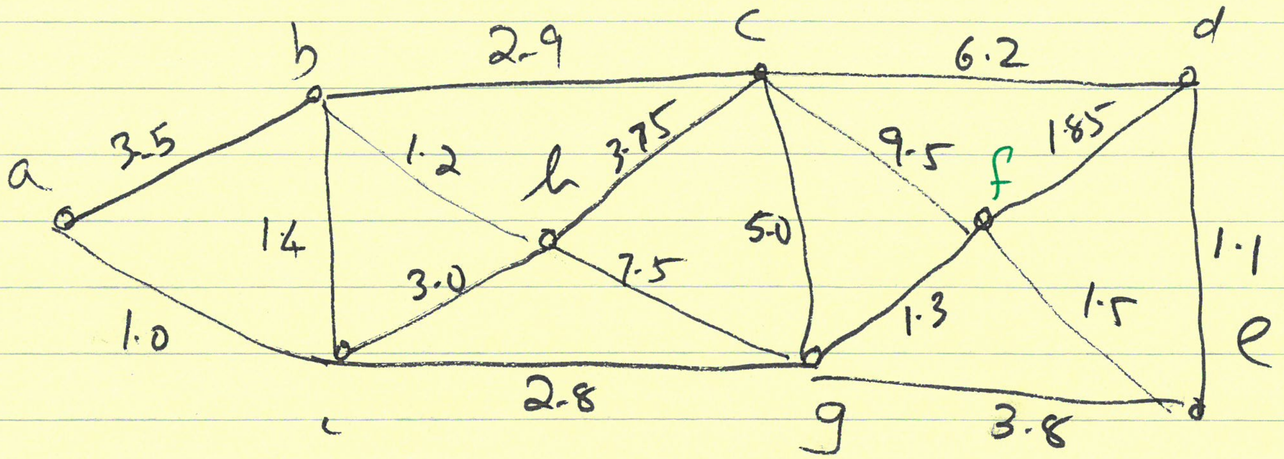
one stage.

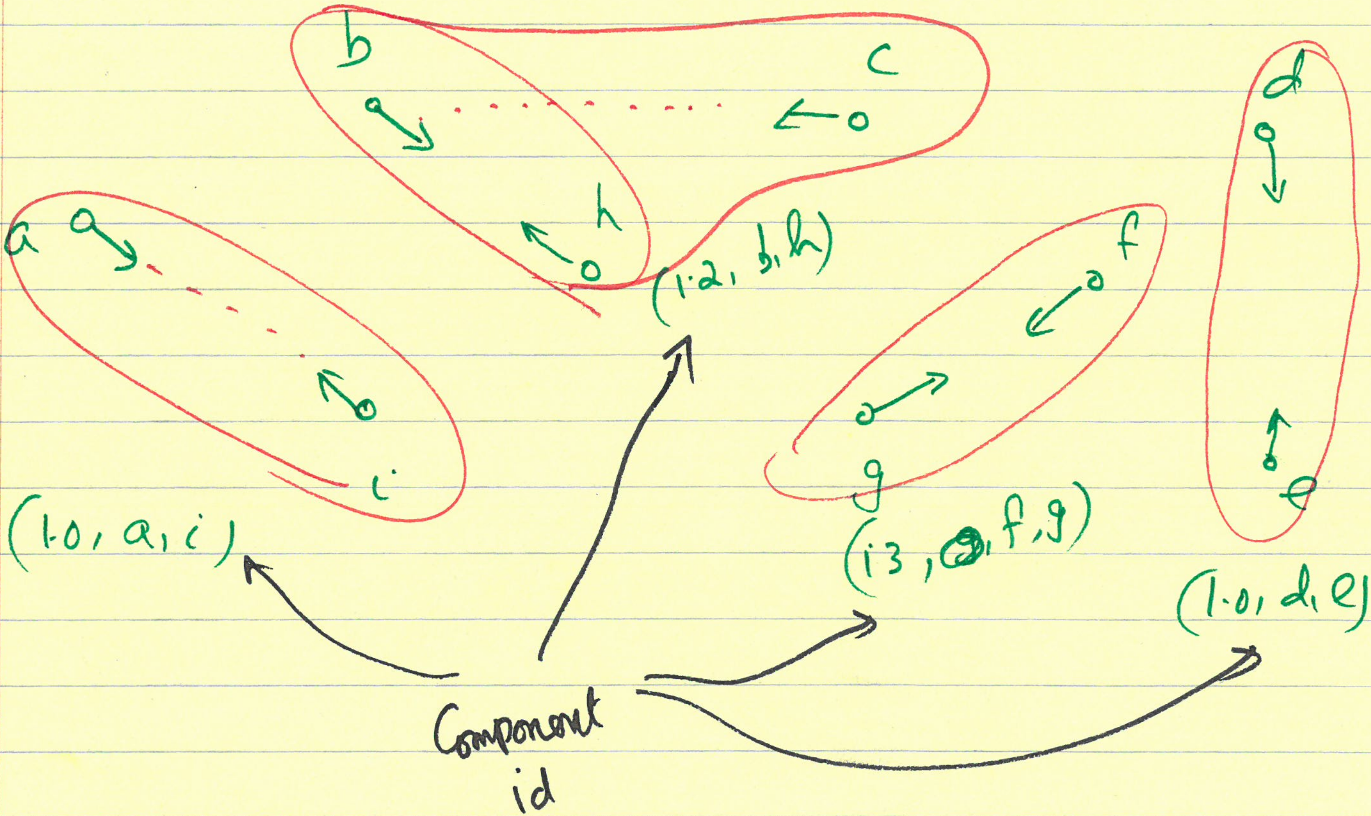
~~Tree~~: Forest



Component id = (1.0, a, i)

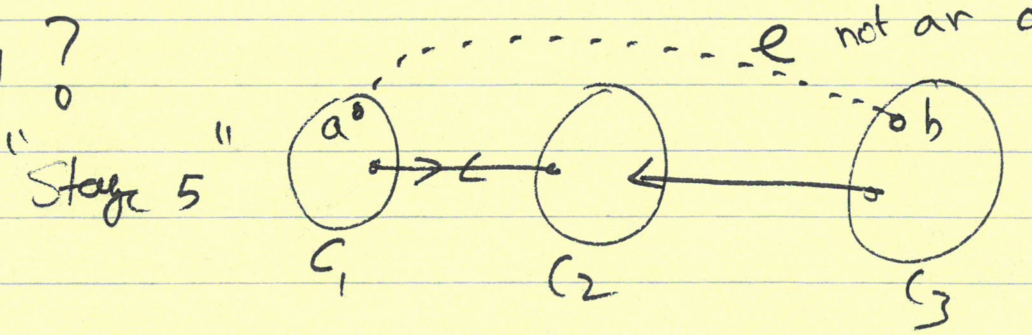
5





Merging and Combining for a "stage" has to be completed ~~before~~ for all nodes before "next stage" can begin. for any node.

Why?



e not an outgoing edge, but a may send a test message to b.

⑥ Time $O(\log_2 n)$ stages. $O(n)$ rounds/stage

$: O(n \log n)$

message:

test / ~~[illegible]~~ accept $O(n \log n)$

test / reject $O(|E|)$

Initiate $O(n \log n)$ ⟵ $\{ O(n)$ per stage

Convergecast reply ↓ $\{$ for all components

Connect $O(n \log n)$

message: $O(|E| + n \log n)$