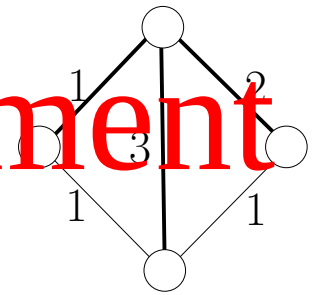
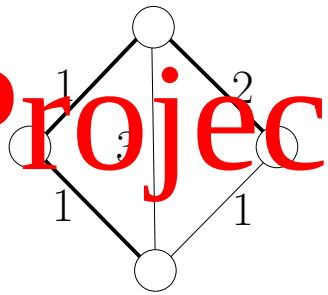
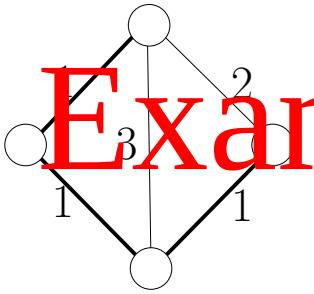
Distributed MST

14 Aug 2020

Spanning trees (ST)







For (1,2,3,...) :

- (1) : min-height ST
  here also BFS ST (cf. sync Echo)
- (2) : shortest paths ST (cf. sync/async Bellm
- (3) : minimum ST (cf. sync/async GHS)
  here also DFS ST (cf. sync/async Cidon)
- (1,2,3,...) : arbitrary ST (cf. async Echo)

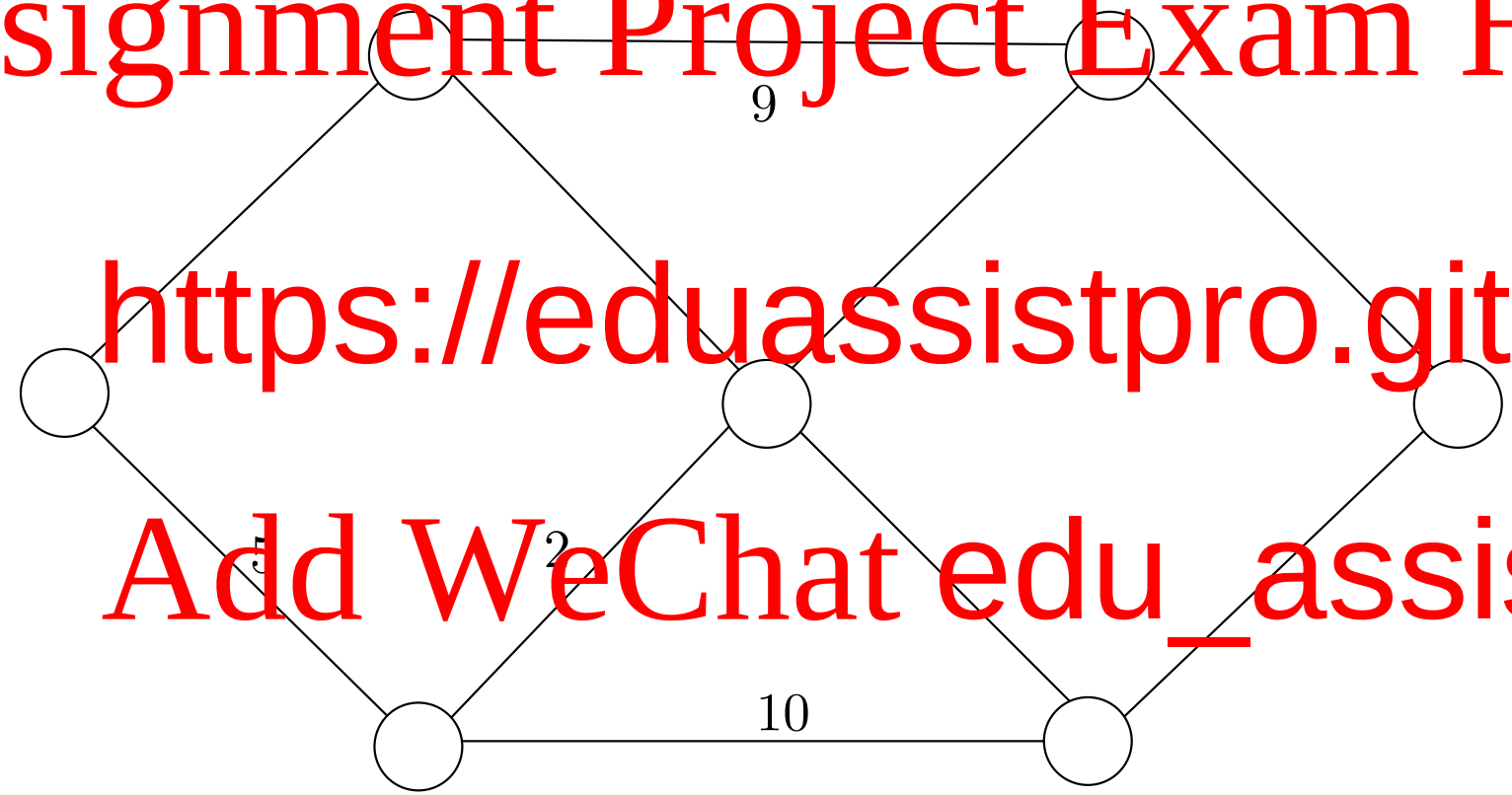
Minimum spanning tree (MST) algorithms

- If edges have different weights, then there is a unique MST
- Prim (1957), Jarník (1930), Dijkstra (1959) – $O(M \log N)$ or $O(M + N \log N)$
- og N...)
- ka, log N)
- faster algorithms – almost linear: Chazelle randomization: integer weights; ...
- Distributed MST (sync,async): GHS - Gallager, Humblet and Spira (1983): $O(N \log N)$; improvements linear $O(N)$; or even sub-linear

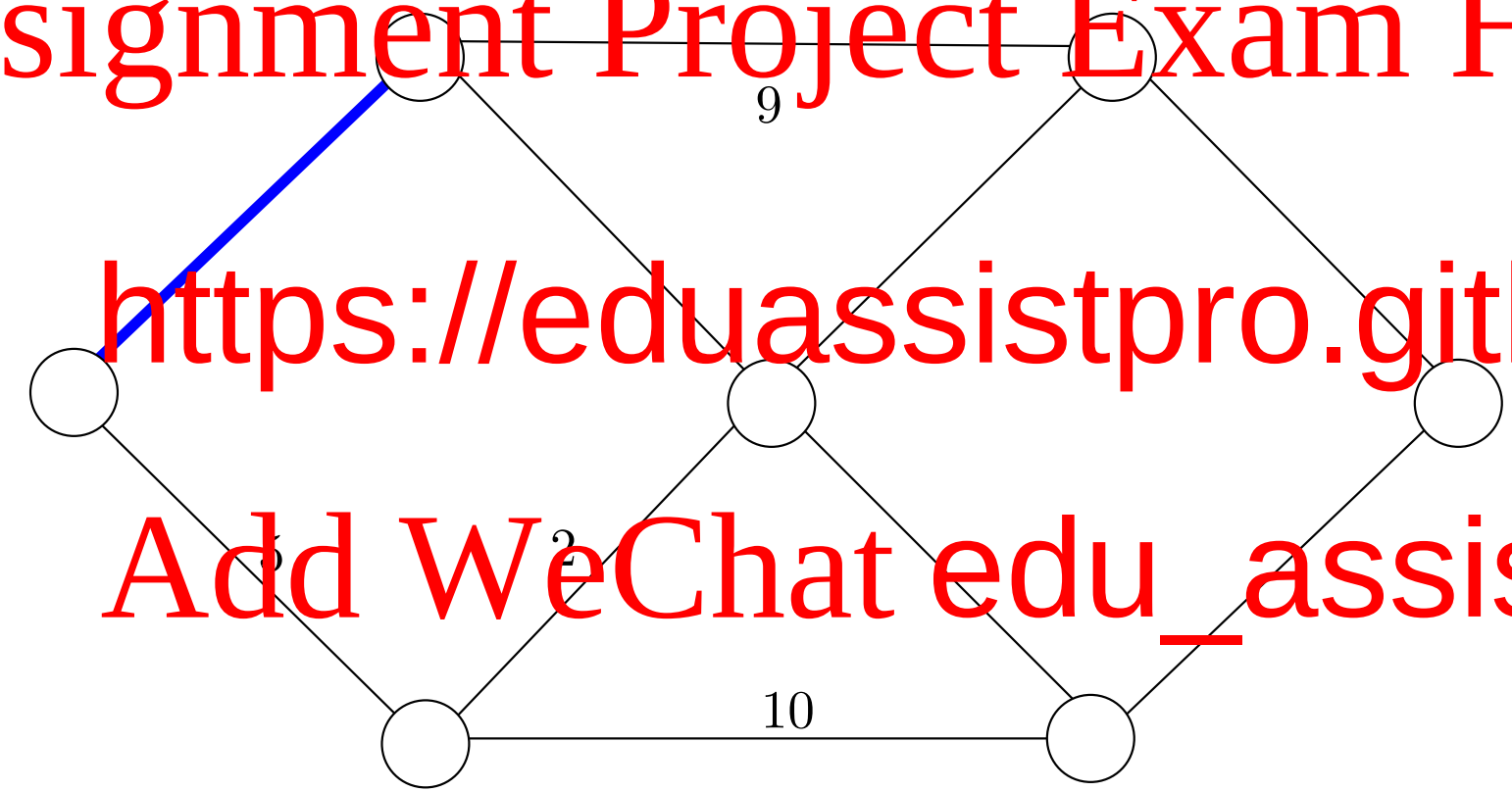
[1] "Because Sollin was the only computer scientist in this list living in an English speaking country, this algorithm is frequently called Sollin's algorithm" (Wiki)

Side-bar: Minimum spanning networks in biology

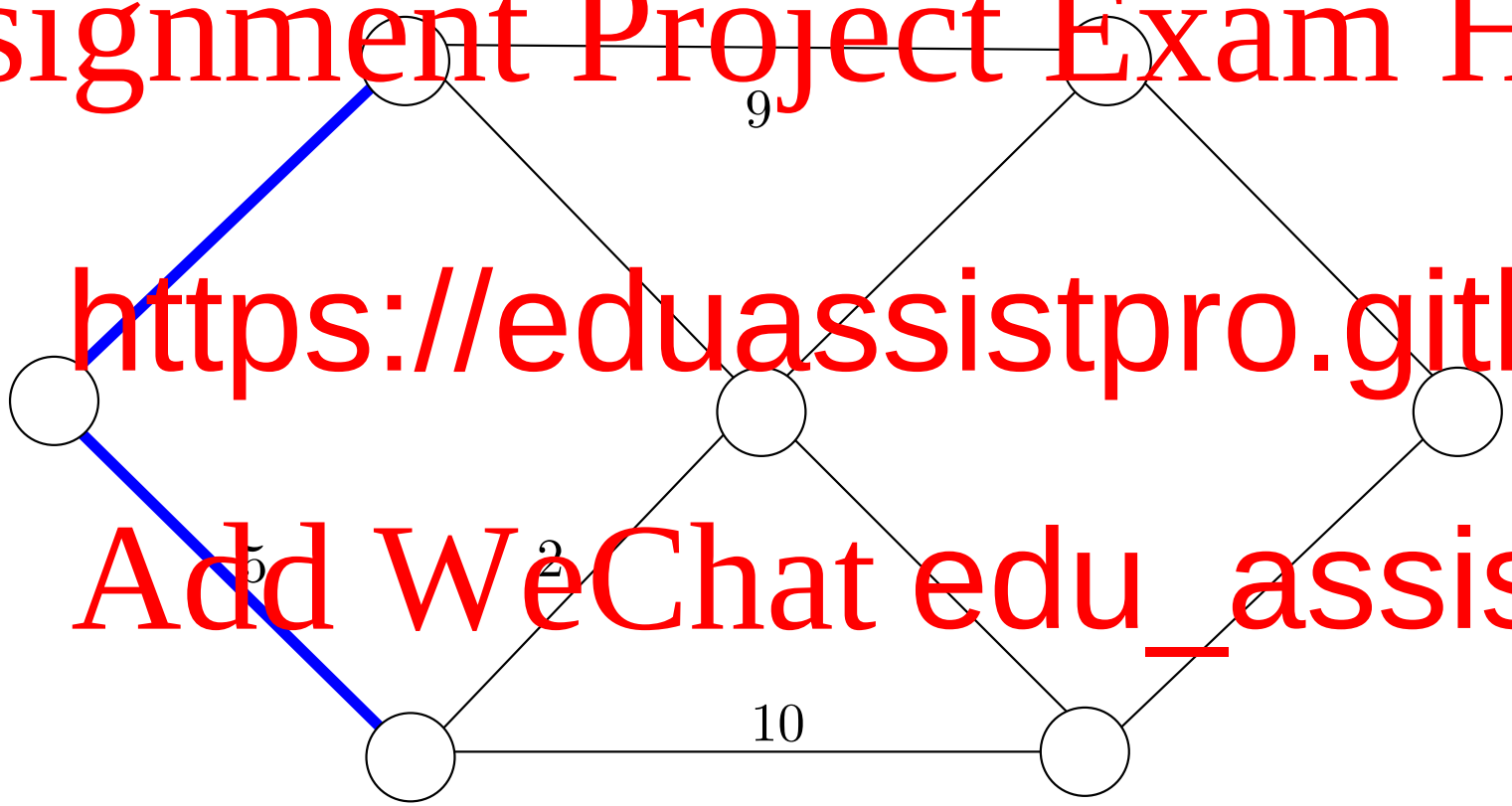
Prim



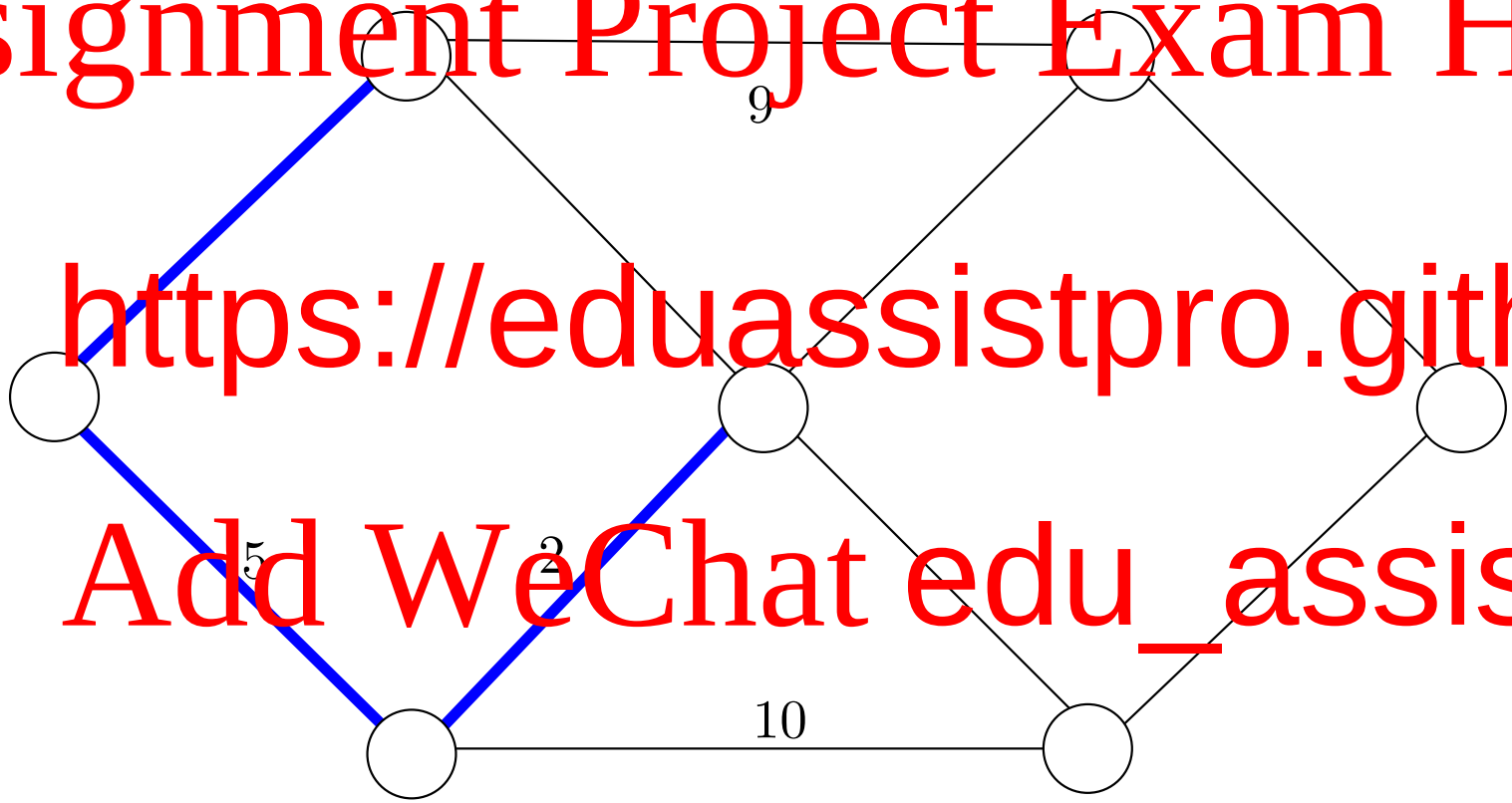
Prim



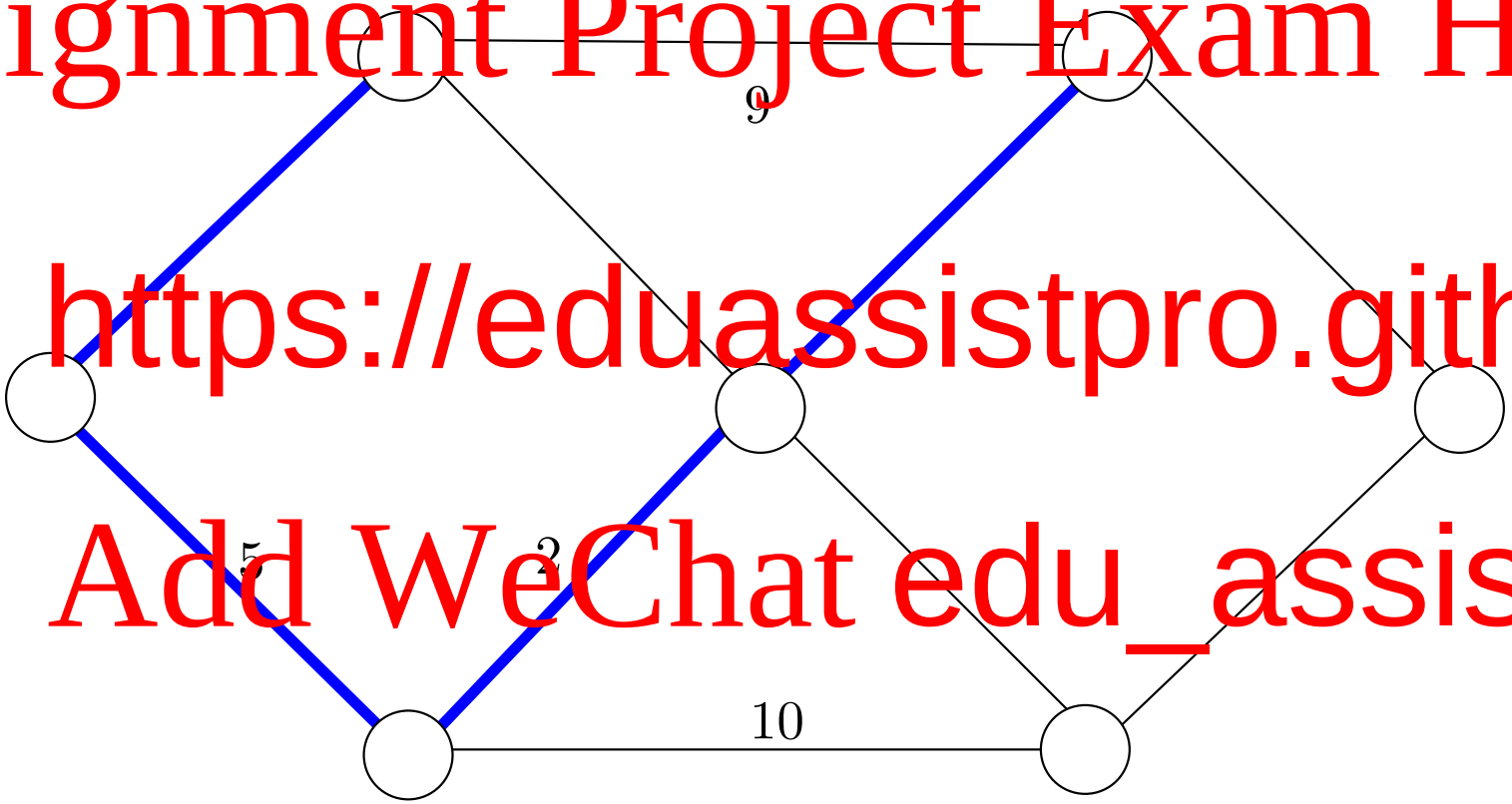
Prim



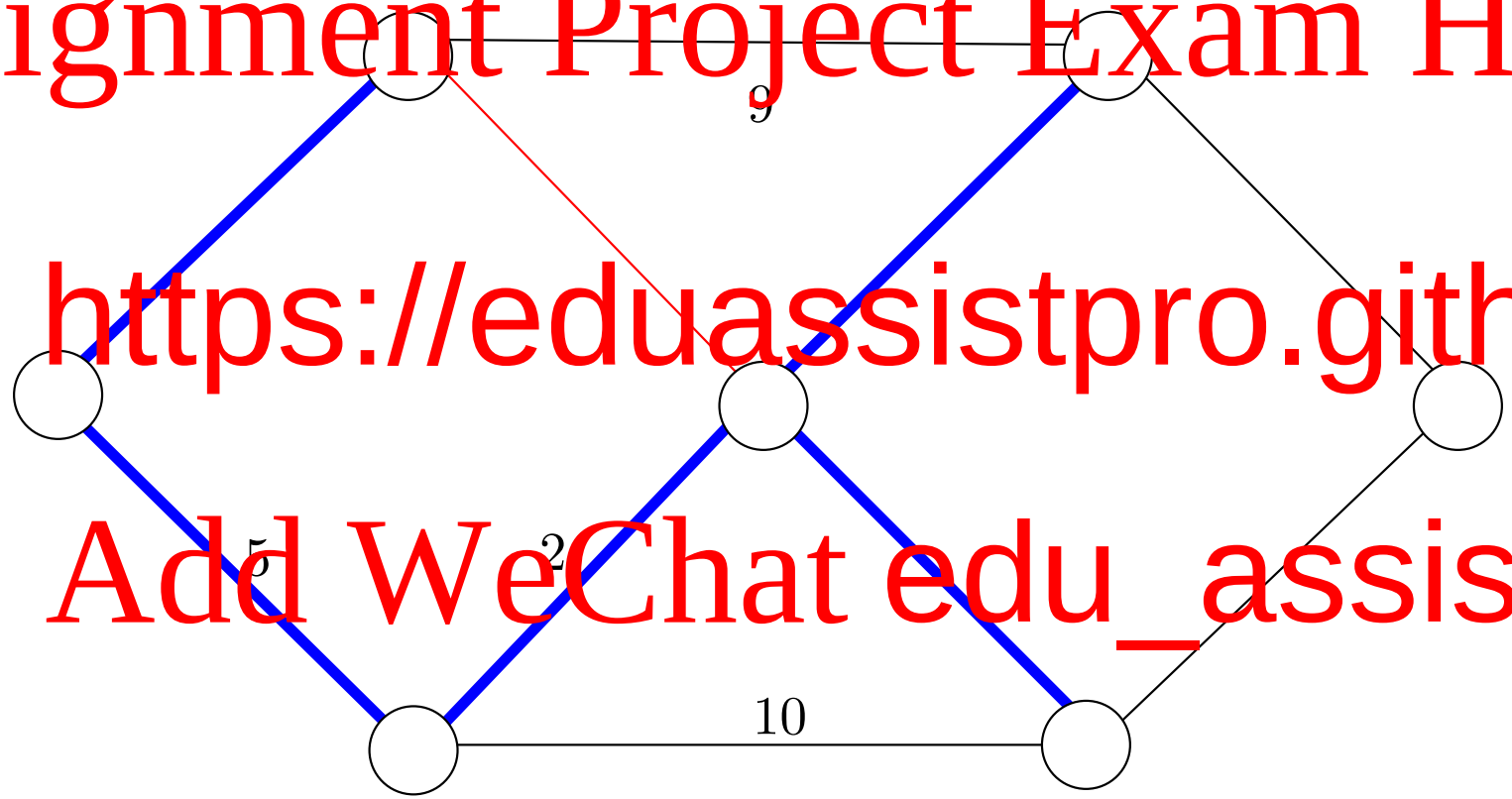
Prim



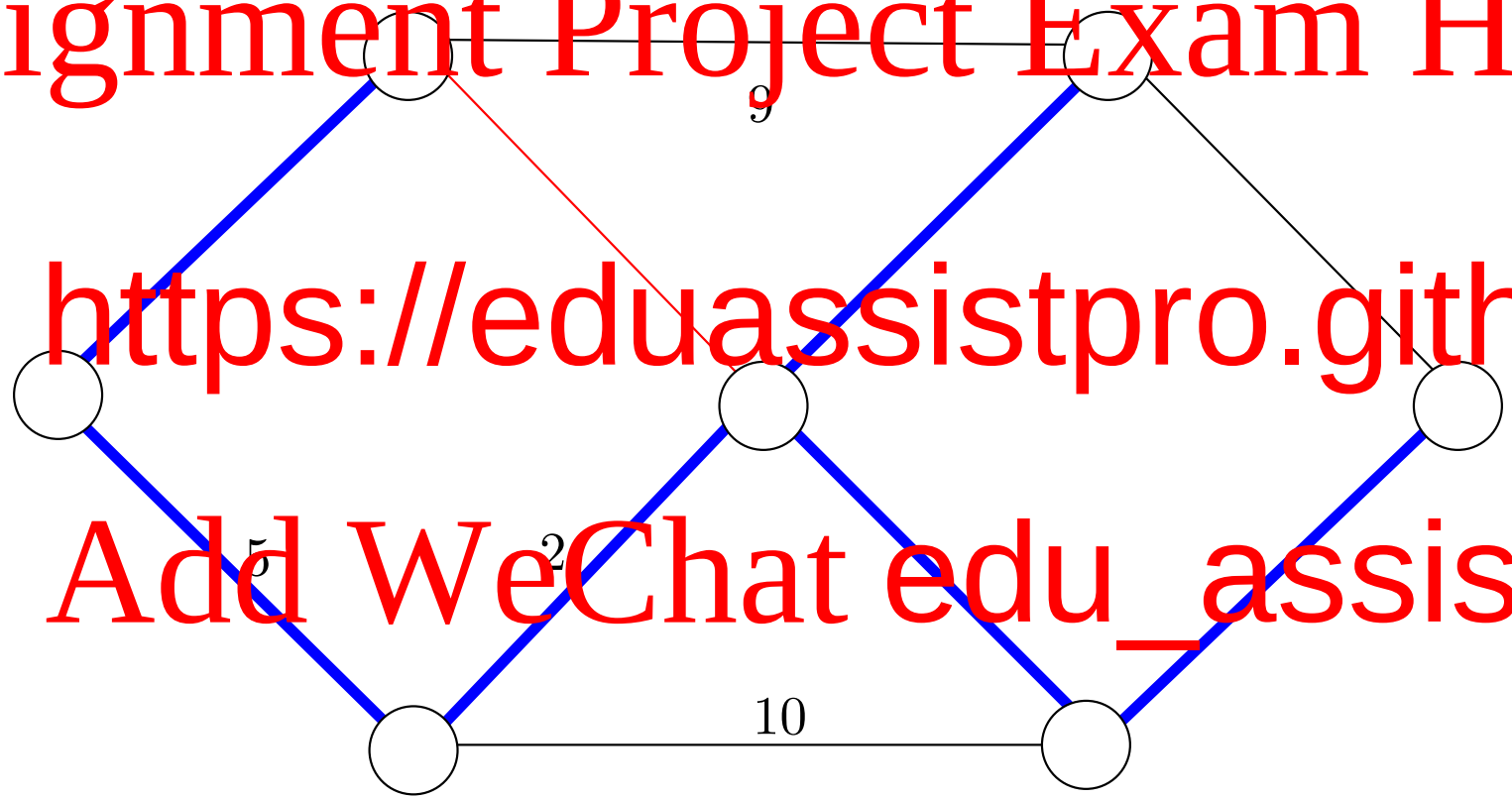
Prim



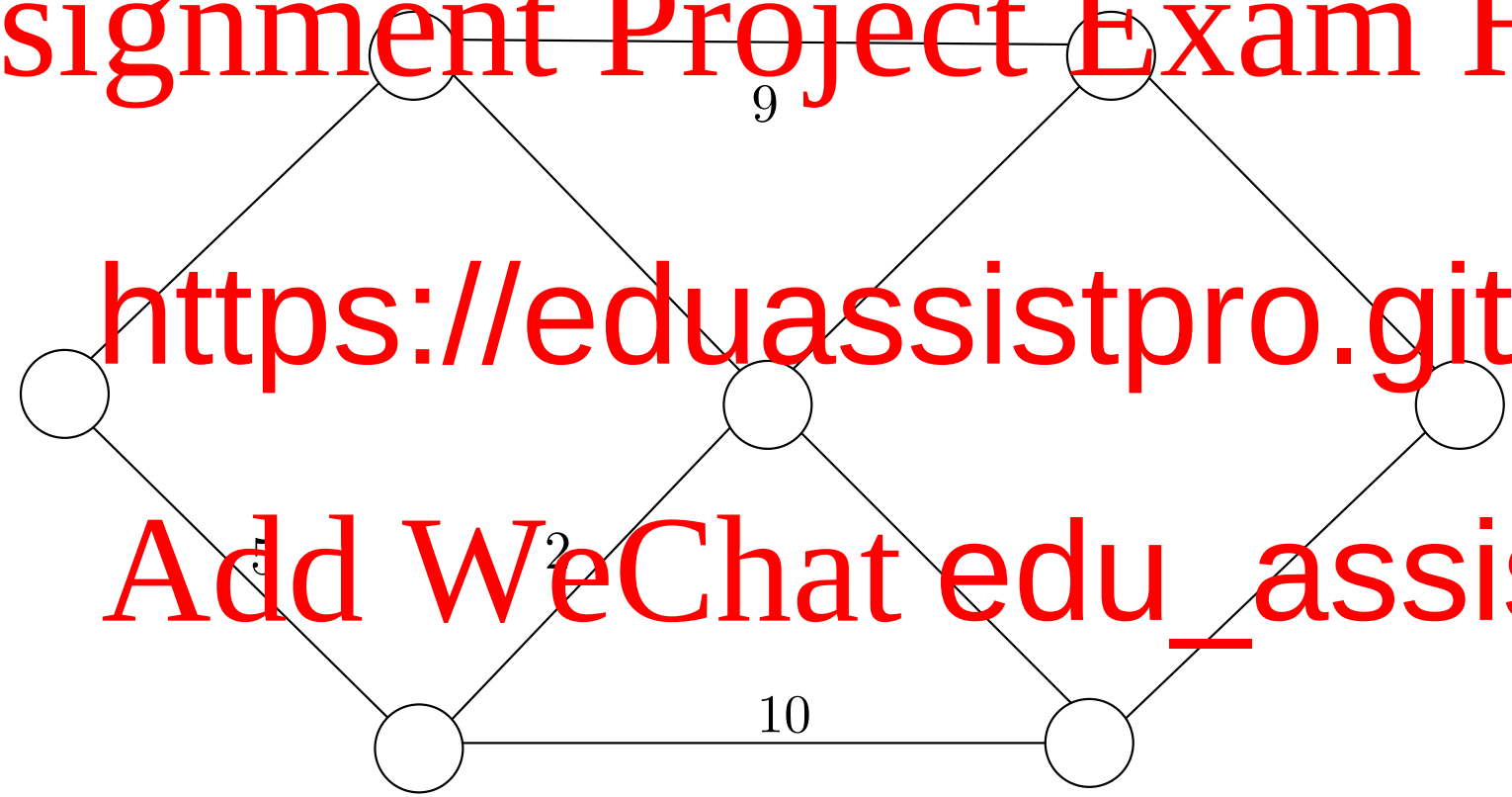
Prim



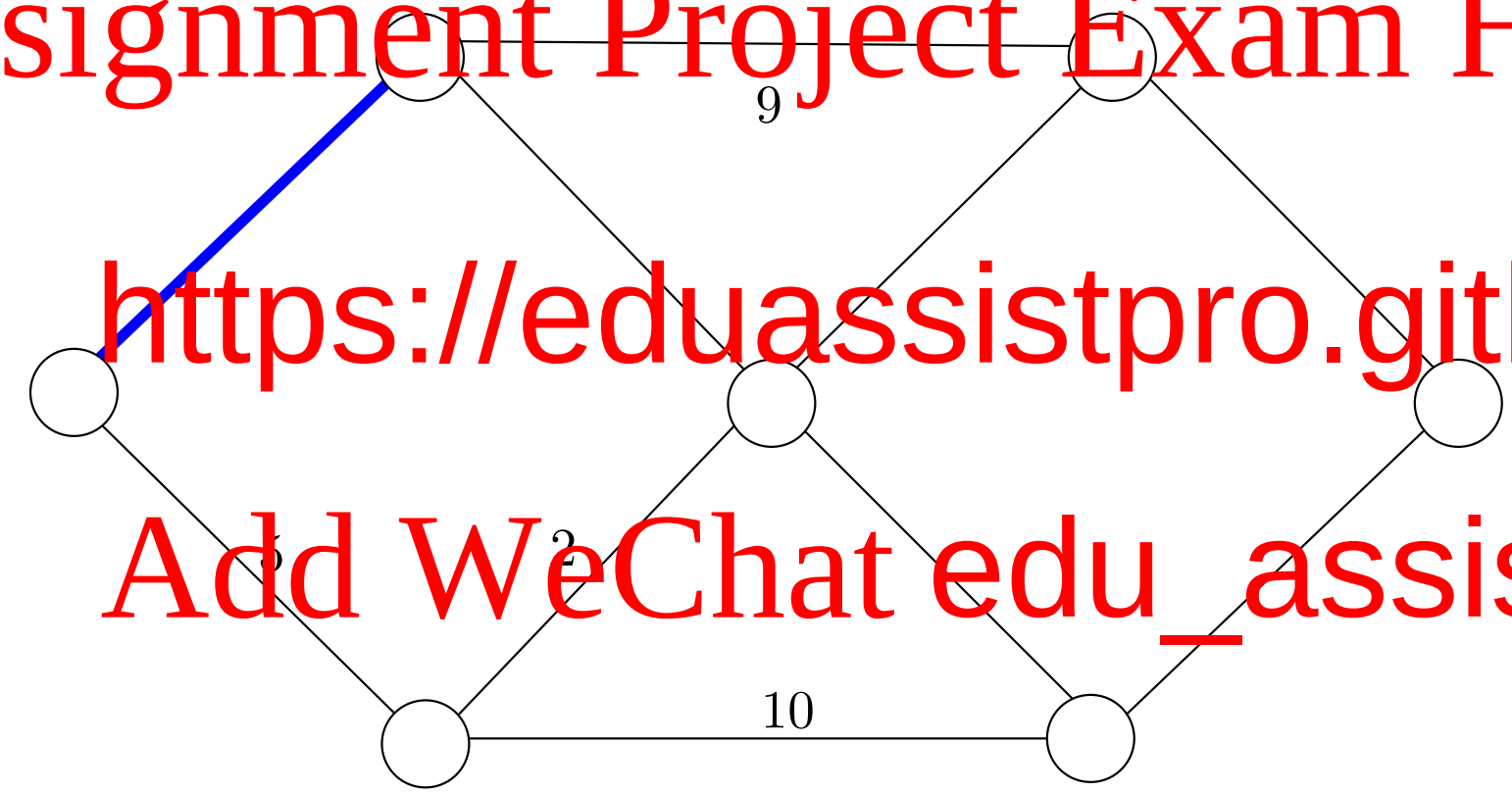
Prim



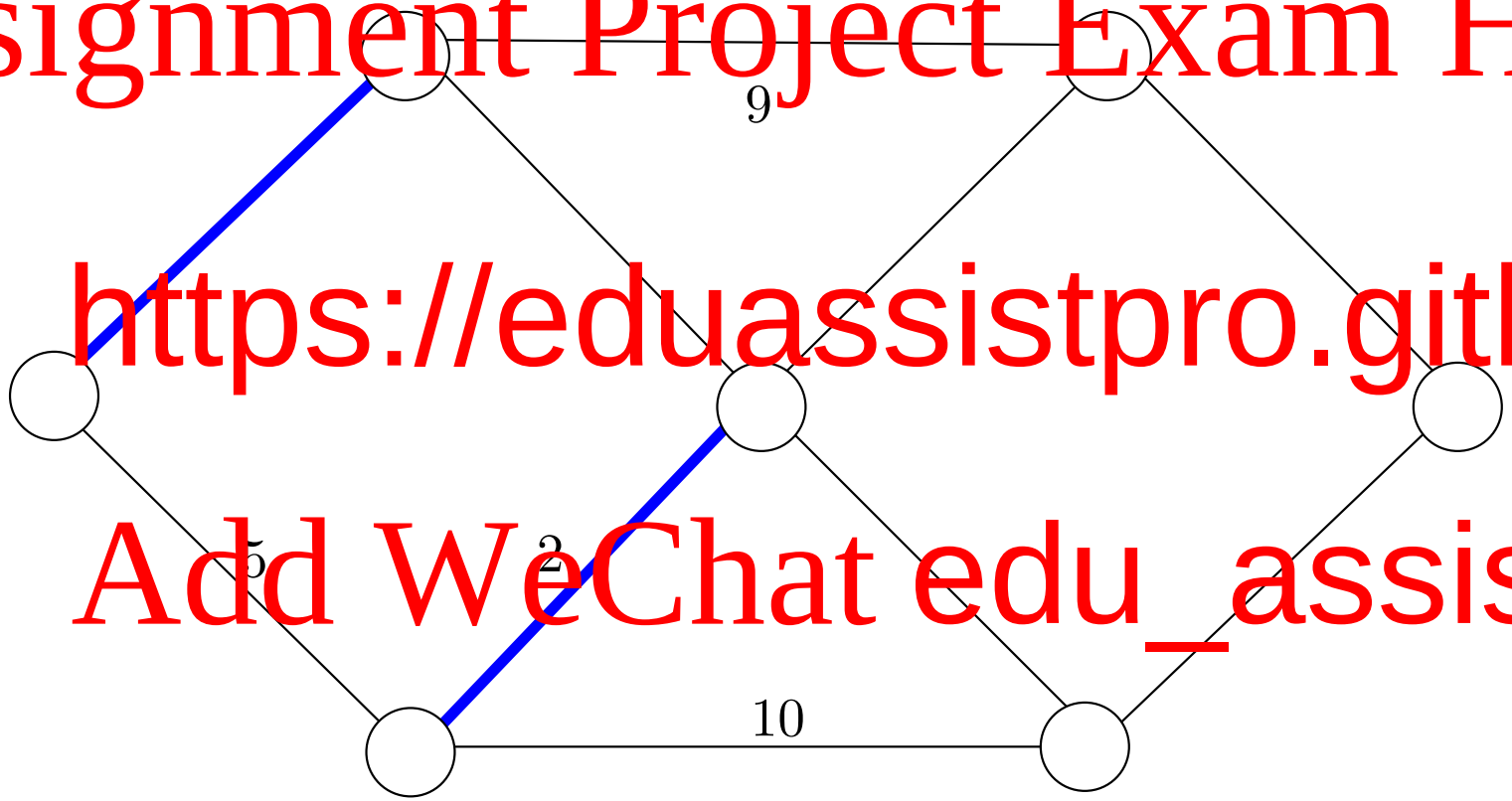
Kruskal



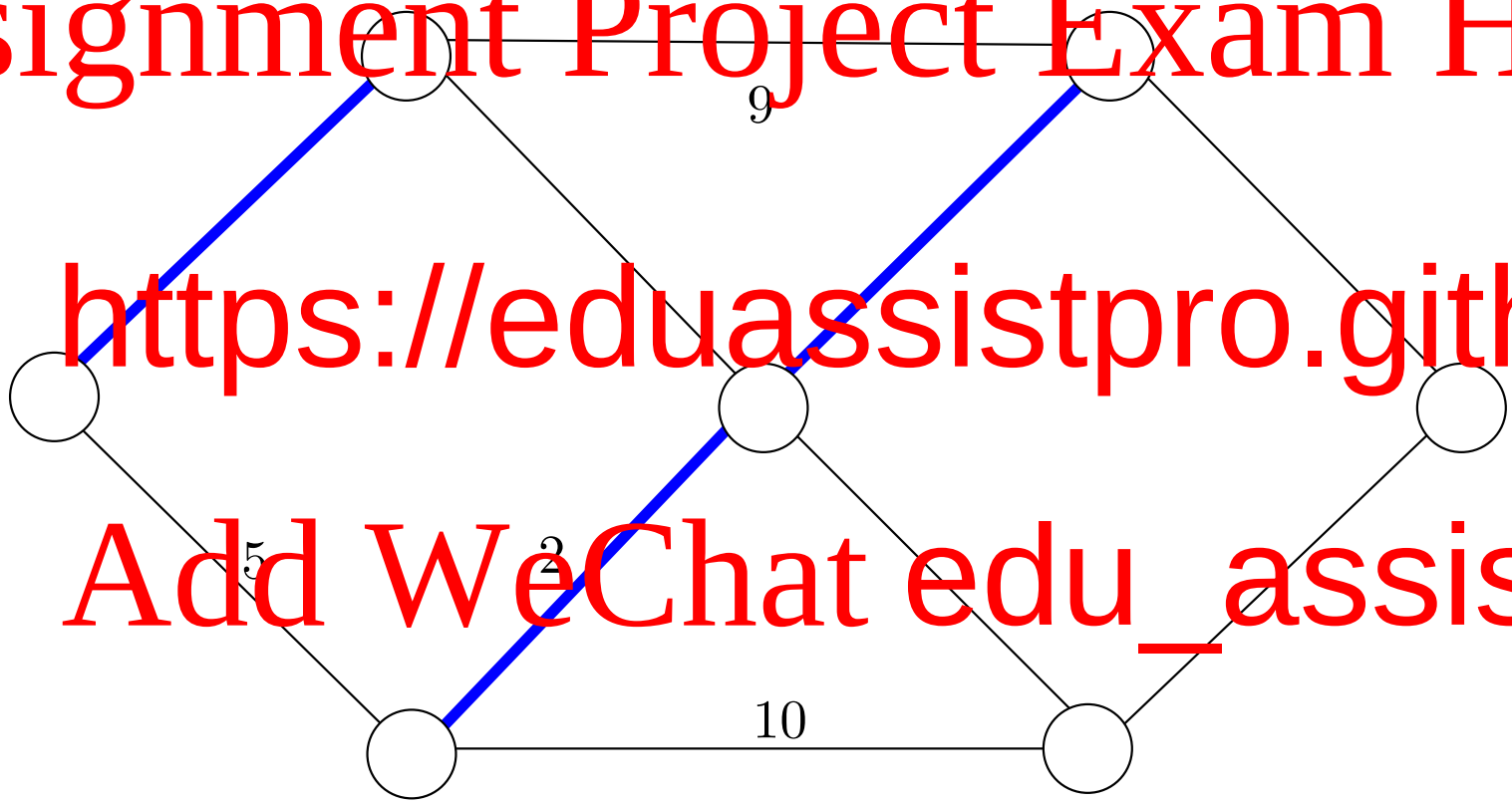
Kruskal



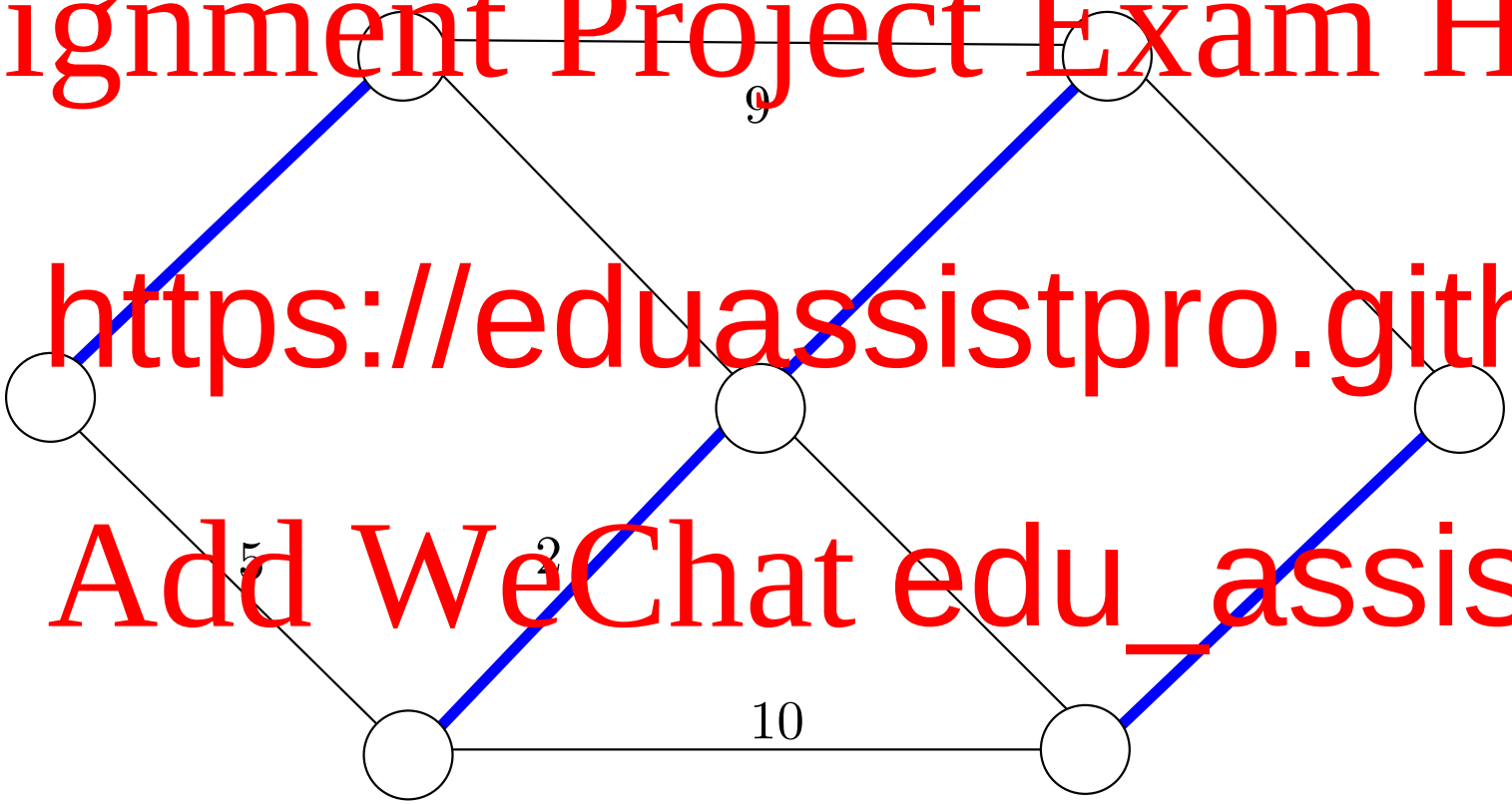
Kruskal



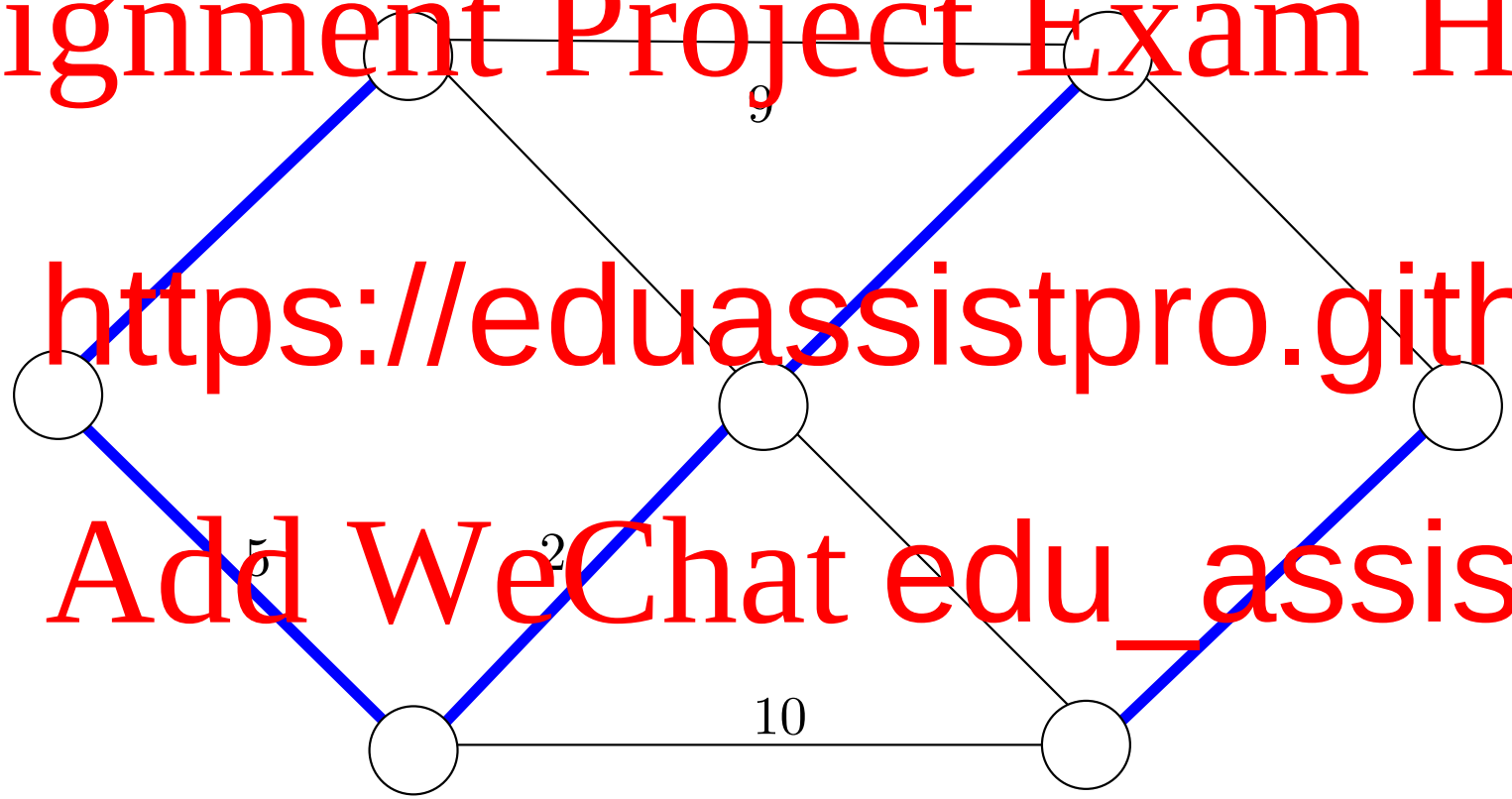
Kruskal



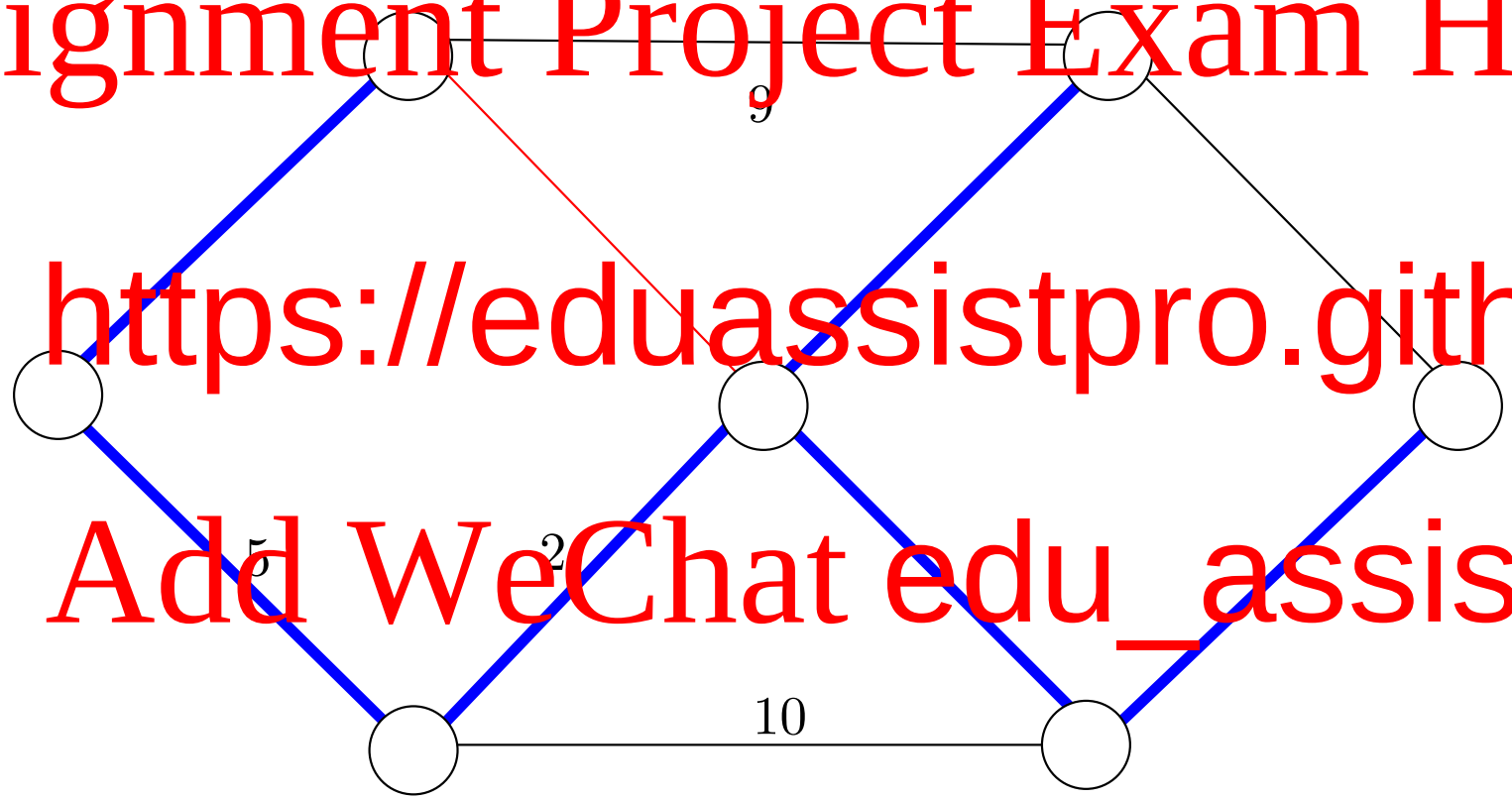
Kruskal



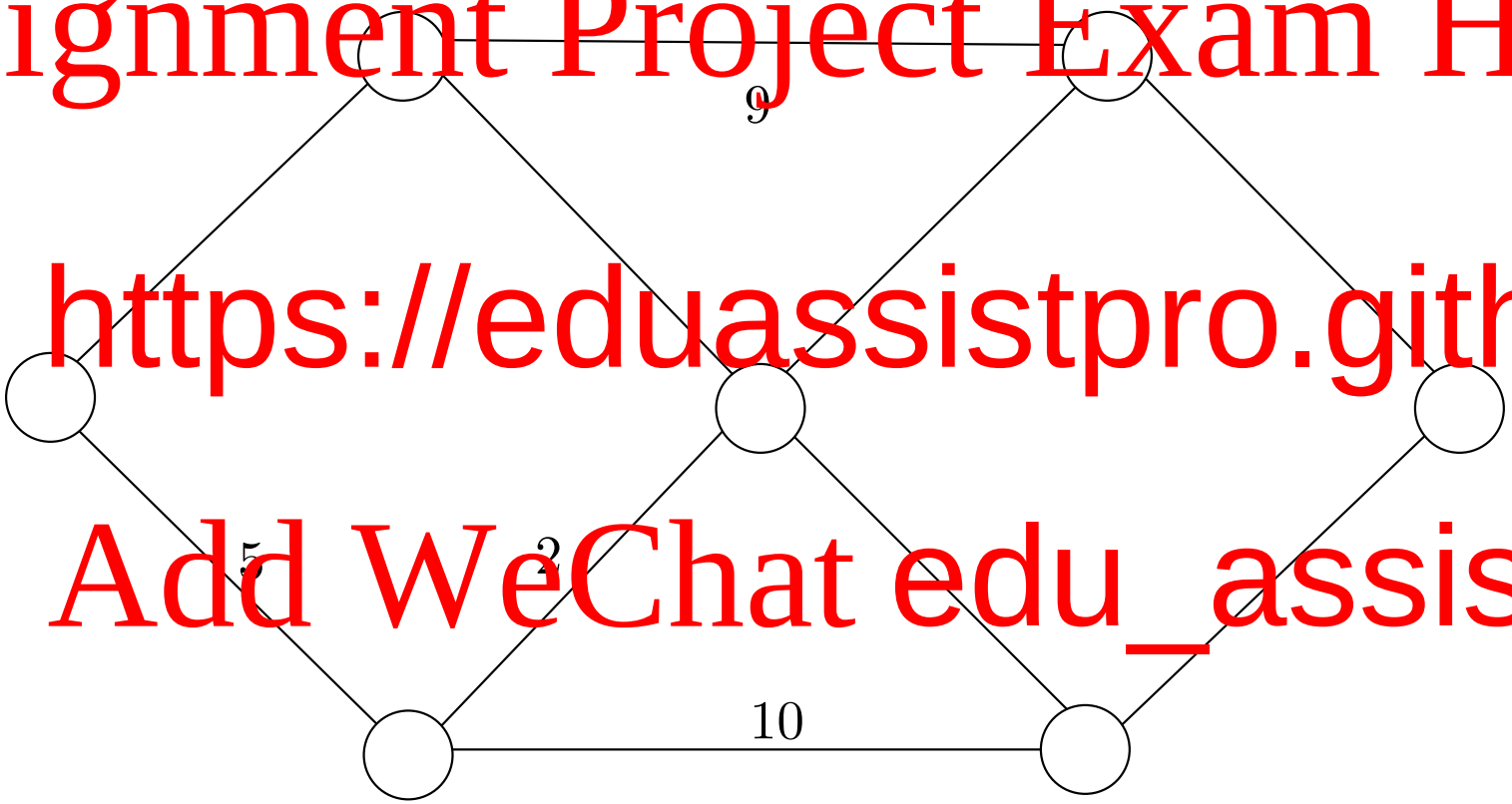
Kruskal



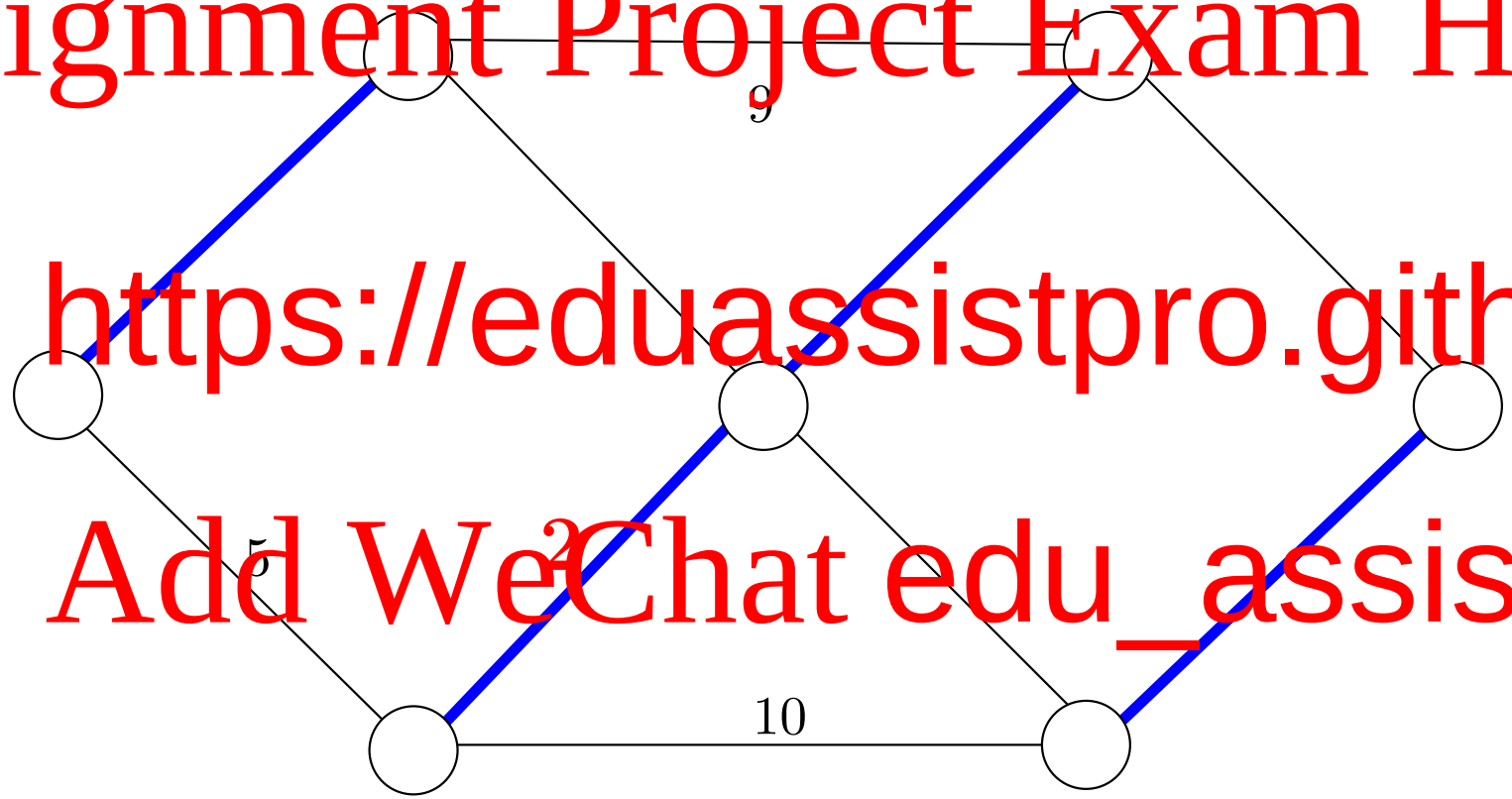
Kruskal



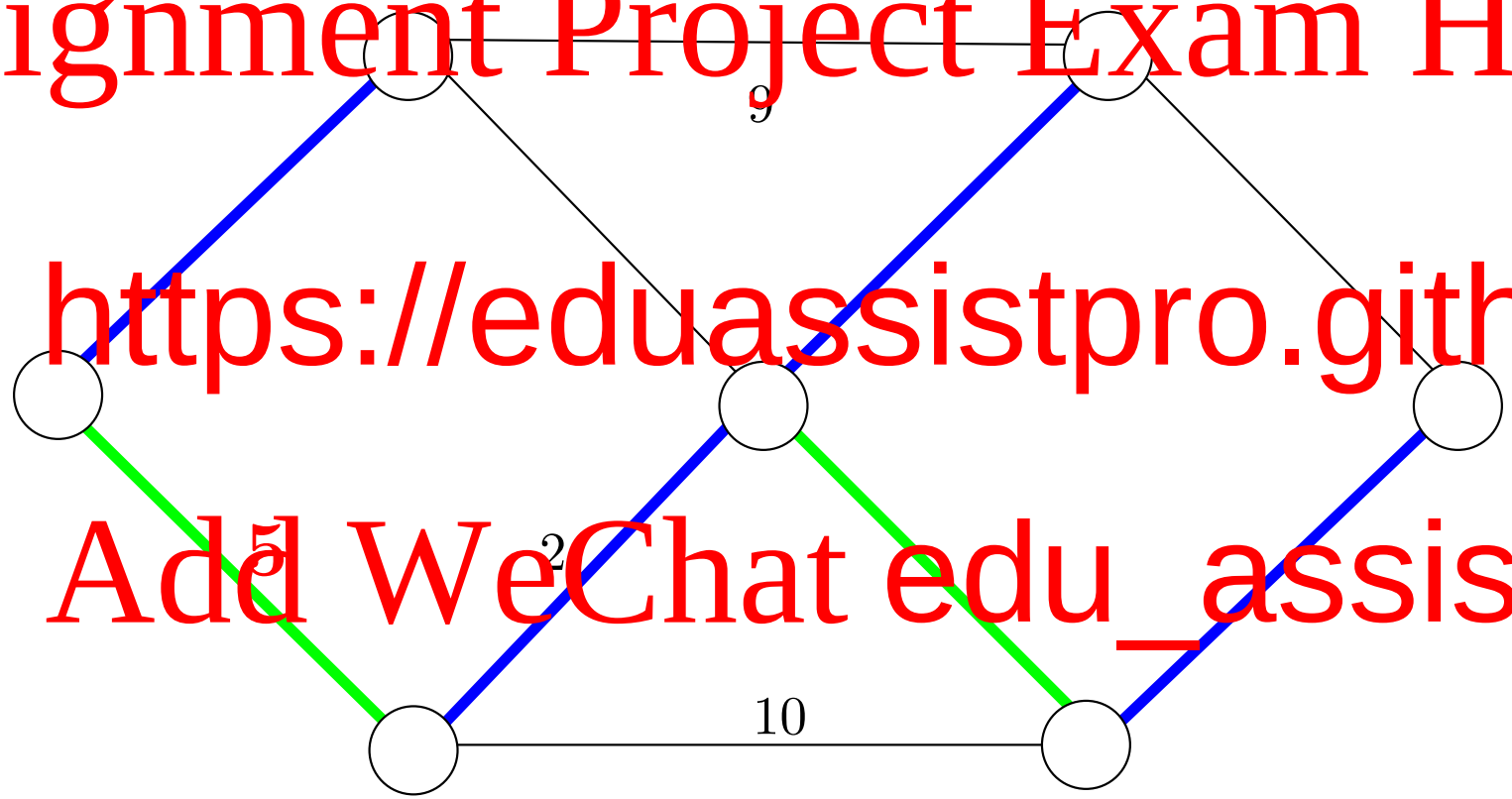
Borůvka – red: Common MWOE (min-weight out edge)



Borůvka – red: Common MWOE (min-weight out edge)



Borůvka – red: Common MWOE (min-weight out edge)

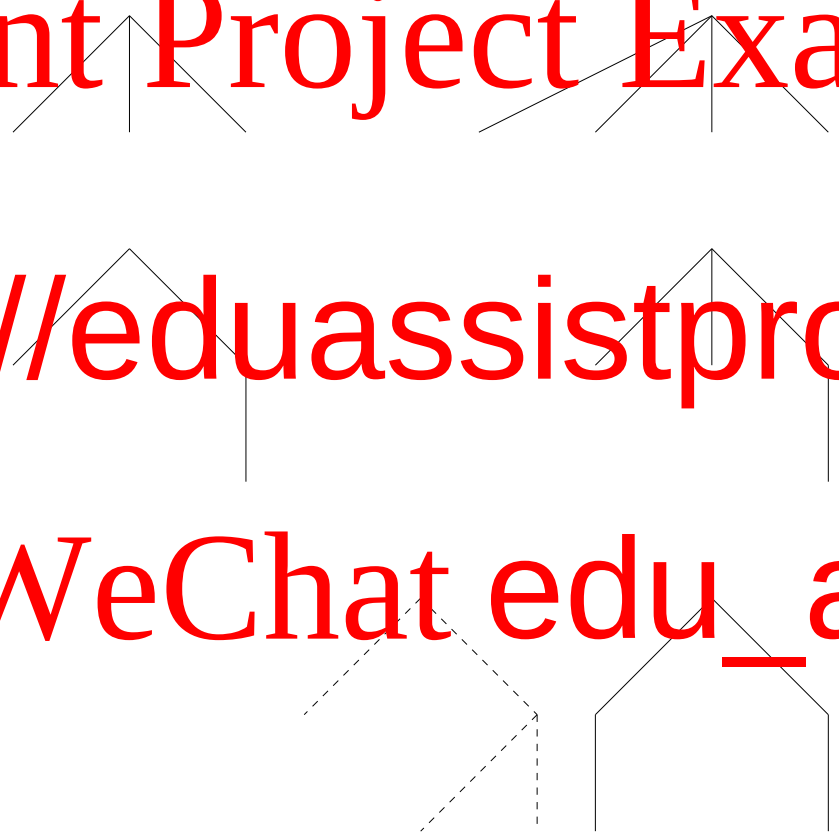


Prim and Kruskal as particular cases of Boruvka

- Kruskal is a restricted Boruvka, where we only use the overall lowest-cost MWOE (not necessarily common) - thus, at any given step, we only merge 2 trees (one single 2-way merge)
- Boruvka deals with all MWOEs at the same "level" (in the distributed case) - thus, we can perform multi-way merges "simultaneously" (a
- Quotation marks indicate that many things may happen at the same "step" or "level" ...
- The distributed MST versions exploit this ability of Boruvka

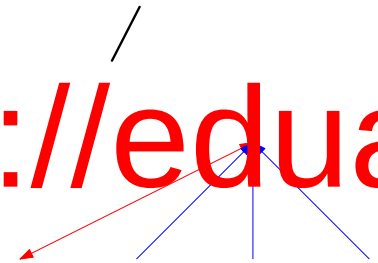
All unrooted trees (edge shapes) with 4 & 5 nodes

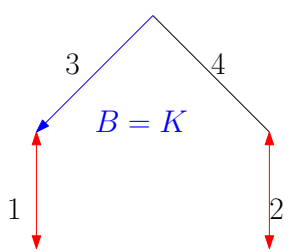
- Looking for MWEs (minimal working examples) where Kruskal is essentially different from Boruvka: 4 or 5 nodes?

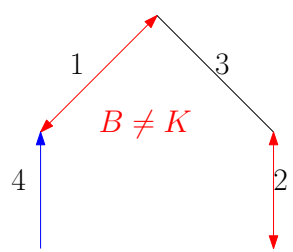


- Here the selected roots have minimum eccentricities
- WLG (without loss of generality), are these all that need consideration?

Five nodes – Kruskal essentially different from Boruvka?

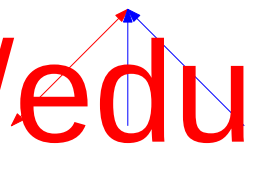
- B step 1: collects all red (Common MWOE) and blue (one way MWOE) edges
- K collects in order: red edges (costs 1 ,2), then costs 3, 4
- 

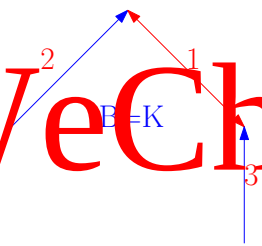


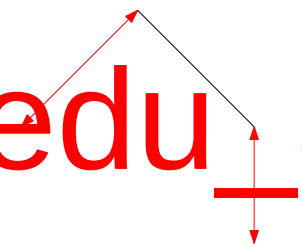


Four nodes

- On 4 nodes, Kruskal is NOT essentially different from Boruvka







Further discussions

Prove, for Borůvka's algorithm:

- In any multi-way merge there is always one Common MWOE
-

Prove that these hold for:

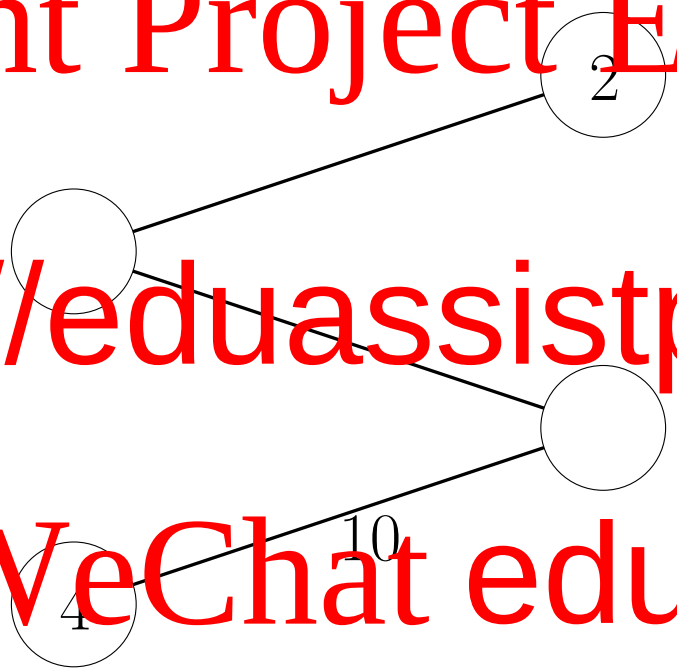
- Level 0 trees, i.e. initial nodes (size 1 trees)
- Level k trees, for any $k \geq 0$

$\exists!$

Distributed MST (Sync)

- Based on Borůvka; see Lynch §4.4
- Nodes have unique IDs
-
- https://eduassistpro.github.io/
  - could be obtained by a preliminary phase
- Edges have unique weights or same weight e ordered
  - e..g, by lexicographical comparisons, where each edge $\{i, j\}$ is represented by an ordered triple $(w, v, v'), v < v'$, where $w =$ edge weight, $v, v' =$ ID's of i, j

Sync MST – Comparing weights with equal weights



$$\{1,2\} = (10,1,2) < \{1,3\} = (10,1,3) < \{4,3\} = (10,3,4)$$

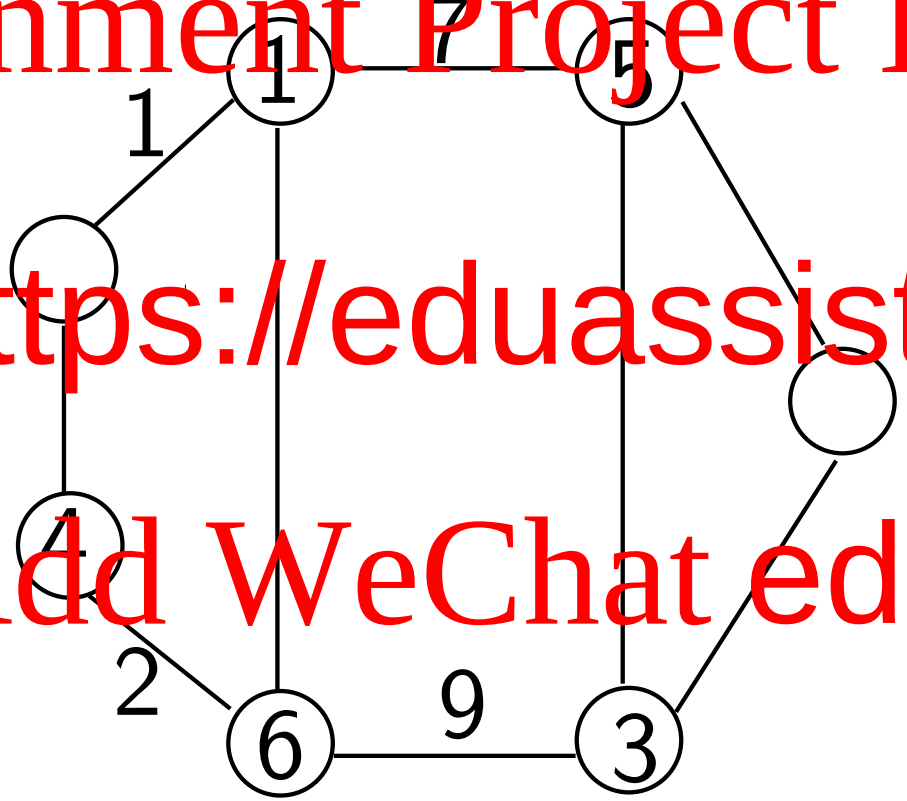
Distributed MST (Sync)

- Time complexity: $O(N \log N)$
- Message complexity: $O((N + M) \log N)$
- Levels: $O(\log N)$; each level defines a spanning forest
-
-
- level k components into new compone
- Thus, level $k \geq 0$ has component subt
- Each component has a distinguished lead
  identifies the component
- For connected graphs, the algorithm ends with a MST
  (unique, if edges have different weights)

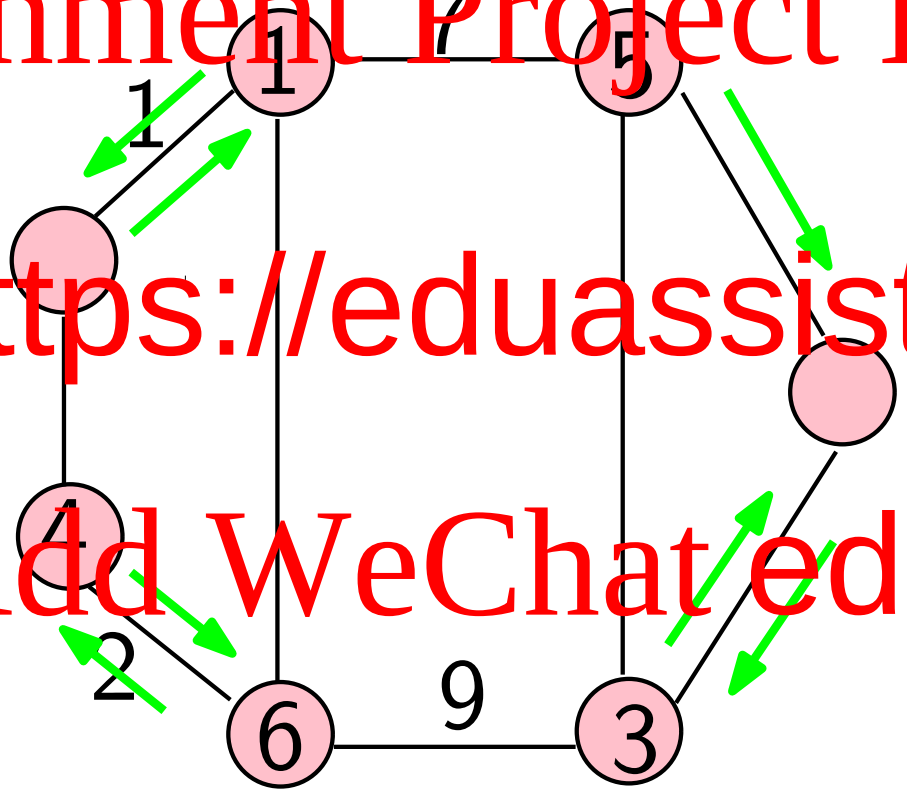
Distributed MST (Sync)

- Details may vary, with slightly different performance...
- The following diagrams use a set of suggestive, but not exactly necessary, messages
  - *report*
  - *connect* (implies a *change-root*)
- To ensure correct component identificat 1 takes a predefined number of steps, $O(N)$ – nodes may need to stay idle until this count is completed
  - depending on the actual algorithm details, this may happen in different ways

Sync MST – Level 0 Components

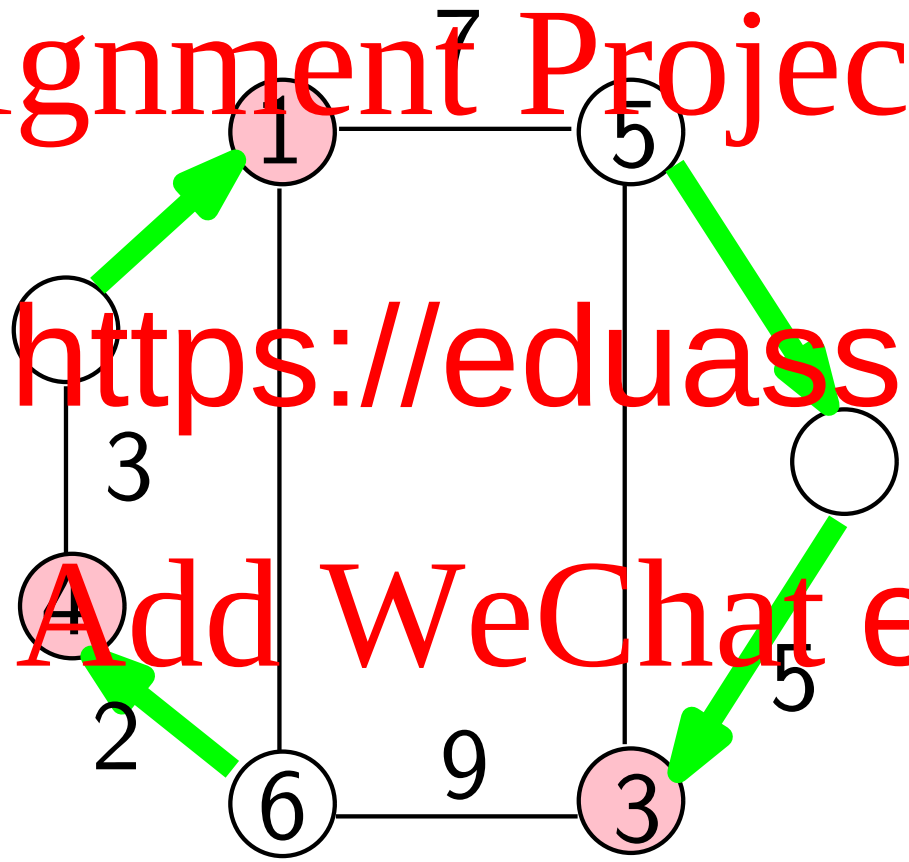


Sync MST – Level 0 Components

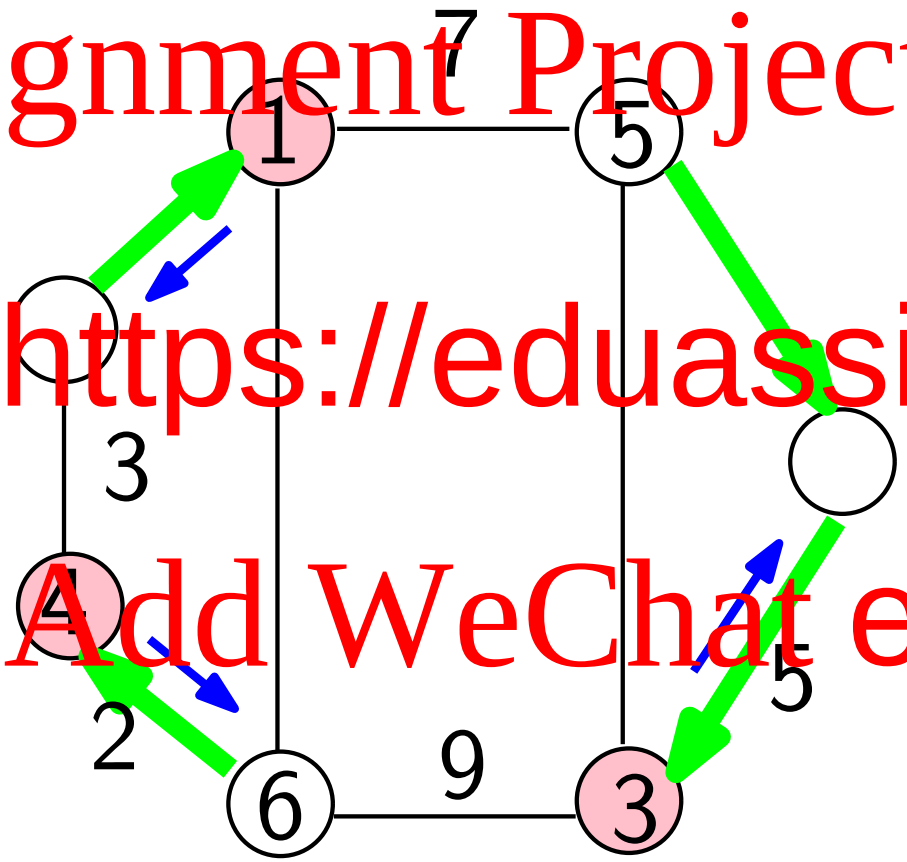


connect!

Sync MST – Level 1 Components

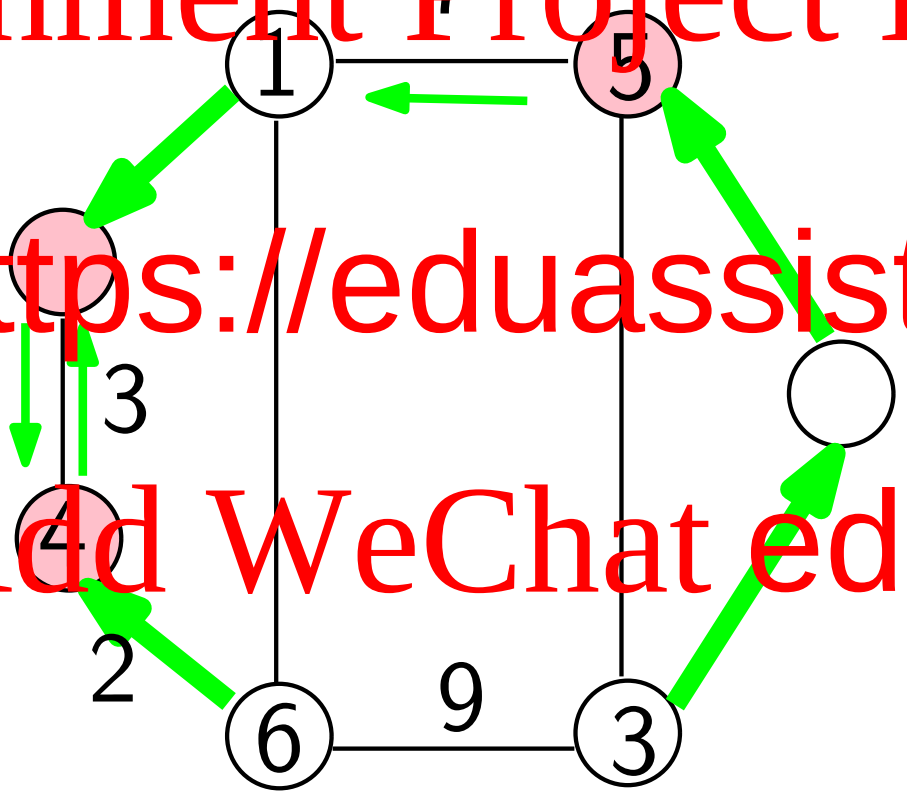


Sync MST – Level 1 Components



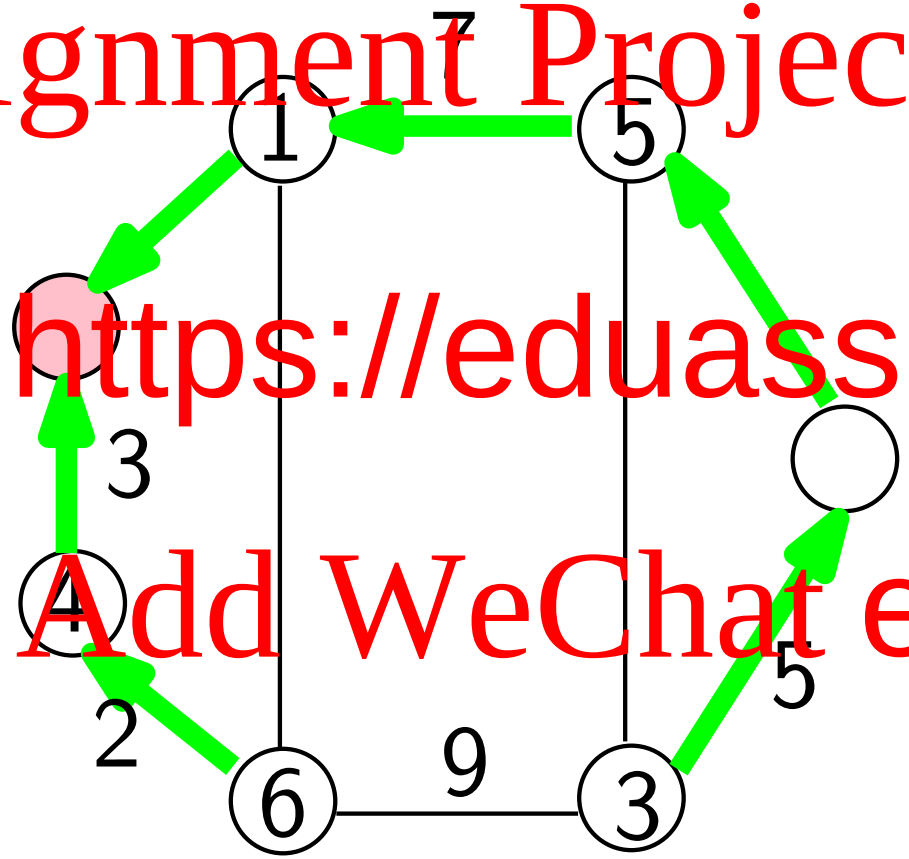
initiate

Sync MST – Level 1 Components

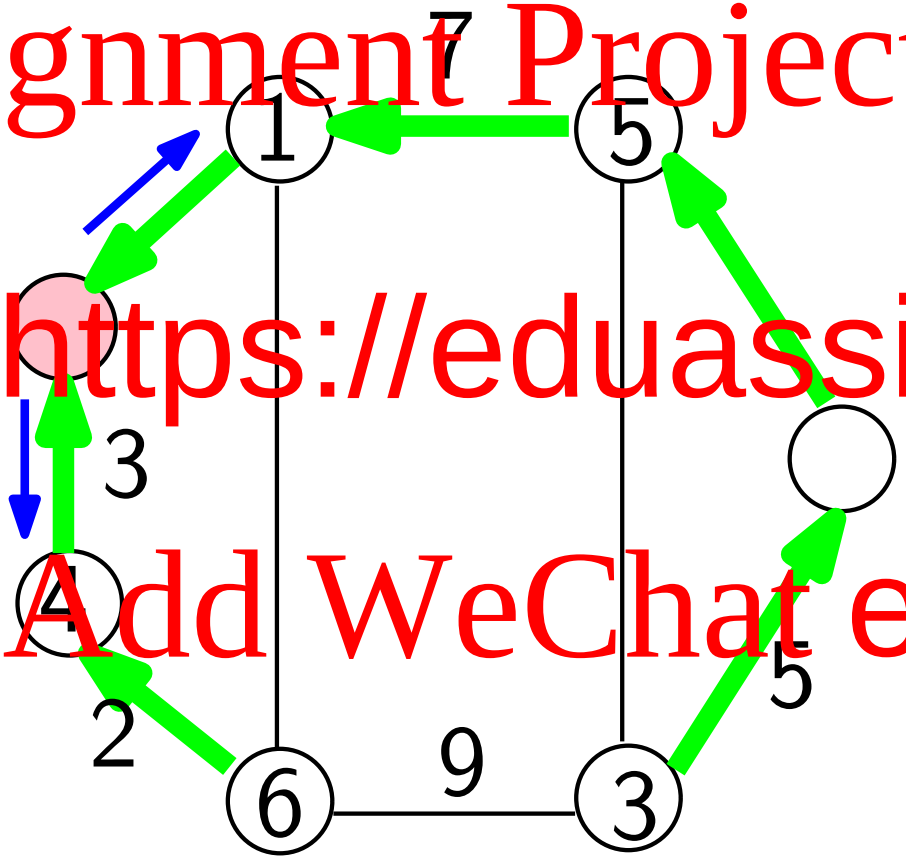


connect!

Sync MST – Level 2 Components

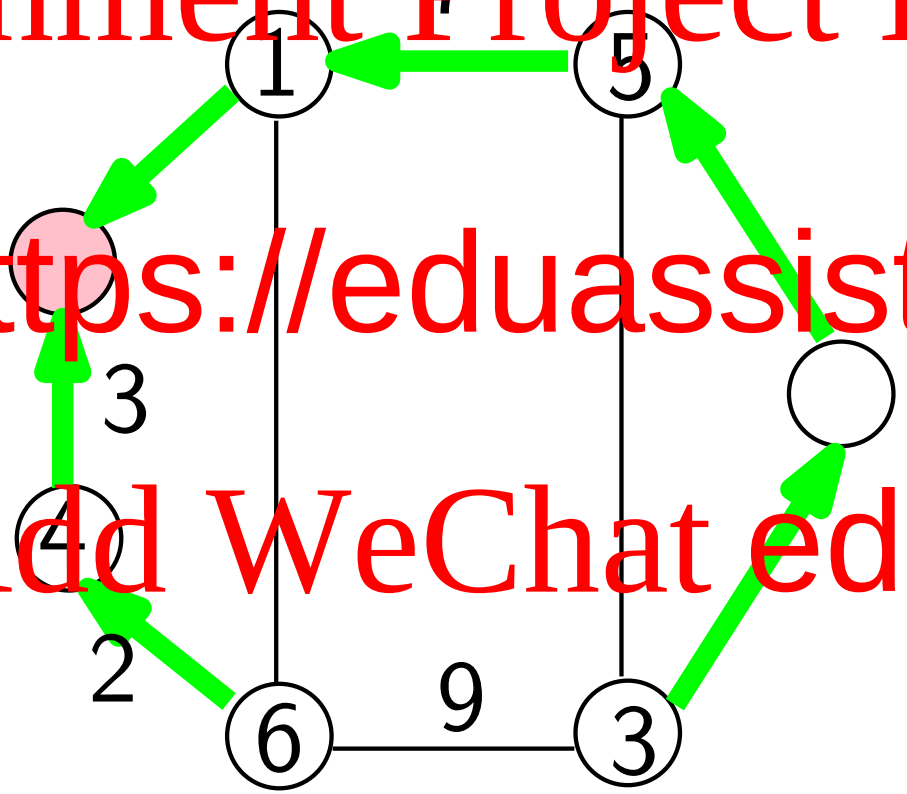


Sync MST – Level 2 Components



initiate

Sync MST – Level 2 Components



Sync MST – Memento

- Distributed Sync MST is based on Borůvka
-
-

- At each given step: Kruskal merges exactly t
merges of trees simultaneously (2-way or n

- The steps of sequential Borůvka correspond to more complex levels (phases) in distributed Sync MST

Sync MST – Memento

- The MST is unique, if weights are pairwise distinct (but this can always be arranged, e.g. by lexicographic comparisons)
- There is one single common MWOE, aka the core, in all tree
- the core, i.e. of the single common MWOE (t that the root stays reasonably close to the lea
- A level k union tree has size $\geq 2^k$ exponentially and there are at most $\log N$ phases
- To ensure required synchronicity, each level requires $O(N)$ steps (this is ok, as there are only few levels)

Sync MST – Memento

- The accept and reject messages are not really needed; i.e. the test messages may hold enough info for this decision
- The report messages are evaluated on-the fly, at each node,
- the best MWOE of that subtree
- The connect messages are routed on the tre from the leader to the node holding the overal
- The connect messages reshape the tree, by transporting the leadership, i.e. resetting parent and child pointers along their path

Async MST – GHS and variants

Specific difficulties of the async version - not present in the sync version:

- Two neighbours may be part of the same component tree, but they have not learned this yet (logical error)
- 2^k;
- More generally, component trees may be a (logical and complexity issues)

Read more in Lynch's textbook:

- §4.4 : sync GHS
- §15.5 : async GHS, plus summary revision of sync GHS
- §15.3 : async STtoLeader (on unrooted STs)