$\max\{ \text{level}([v,B]) + 1, \text{level}([w,B]) \}$.

We compute the value $Val([v,B],[w,A])$ at the moment when we

- consider the edge $([v,B],[w,A])$

  if $\text{level}([v,B]) + 1 > \text{level}([w,B])$

- compute $\text{level}([w,B])$

  if $\text{level}([v,B]) + 1 < \text{level}([w,B])$.

Question:

At the moment when $Val([v,B],[w,A])$ is computed, what ~~is~~ can be the first phase such that $([v,B],[w,A])$ fullfills the edge properties for this phase?

At that phase there holds

$$\text{Dom}([w,A]) = \text{Dom}([w,B])$$

$\Longrightarrow$

$([v,B],[w,A])$ cannot fulfill the edge properties before

Phase $Val([v,B],[w,A]) - \text{level}(\text{Dom}(nca([w,A],[w,B])))+$

Furthermore, at that phase

$$\mathrm{DOM}([v,B]) \neq \mathrm{DOM}([w,A]) = \mathrm{DOM}([w,B])$$

Hence,

$$\mathrm{level}(\mathrm{DOM}(\mathrm{nca}([v,B],[w,A]))) > \mathrm{level}(\mathrm{DOM}(\mathrm{nca}([w,A],[w,B])))$$

implies that the edge $([v,B],[w,A])$ is dead.

$\Rightarrow$

$([v,B],[w,A])$ living $\Rightarrow$

$$\mathrm{level}(\mathrm{DOM}(\mathrm{nca}([v,B],[w,A]))) < \mathrm{level}(\mathrm{nca}([w,A],[w,B])).$$

$\leadsto$

At the moment when $\mathrm{Val}([v,B],[w,A])$ is computed, we also compute

- $\mathrm{nca}([w,A],[w,B])$ and then $\mathrm{level}(\mathrm{DOM}(\mathrm{nca}([w,A],[w,B])))$,
- $\mathrm{nca}([v,B],[w,A])$ and then $\mathrm{level}(\mathrm{nca}([v,B],[w,A]))$.

If $\mathrm{level}(\mathrm{DOM}(\mathrm{nca}([w,A],[w,B]))) > \mathrm{level}(\mathrm{nca}([v,B],[w,A]))$

then the edge $([v,B],[w,A])$ is stored in an appropriate data structure.

Question:

What is an appropriate data structure?

- Store the edges into a two-dimensional array $\mathfrak{A}$ such that

  - Row $i$ contains the edges $([v,B],[w,A])$ with $Val([v,B],[w,A]) = i$.

  - Column $j$ contains the edges $([v,B],[w,A])$ such that at the moment of the insertion of the edge $level(DOM([w,A])) = j$.

$\Longrightarrow$

The component $\mathfrak{A}_{ij}$ contain the edges $([v,B],[w,A])$ with

$Val([v,B],[w,A]) = i$ and

$level(DOM([w,A])) = j$ at the moment of the insertion of the edge

In component $\mathfrak{A}_{ij}$, all edges $([v,B],[w,A])$ with $DOM([w,A]) = [u,B]$ are grouped to= gether in the set

$$\mathfrak{A}_{ij}([u,B]).$$

11.06.

For each such a set, the value $S(\mathfrak{A}_{ij}([u,B]))$ is always computed where

$S(\mathfrak{A}_{ij}([u,B])) :=$

$\min\{ level(nca([v,B],[w,\cancel{B}])) \mid ([v,B],[w,A]) \in \mathfrak{A}_{ij}([u,B])\}$

$k := i - S(OT_{i,j}([u,B])) - 1$ denotes the last level such that an edge in $OT_{i,j}([u,B])$ can fulfill the edge properties with respect to the construction of Level $k$.

Let us consider the moment when $Val([v,B],[w,A])$ is computed again. At that moment, the algorithm performs

- <u>If</u> $level(DOM(nca([w,A],[w,B])))$

  $> level(nca([v,B],[w,B]))$

  <u>then</u>

  (* Let $[u,B] = DOM(nca([w,A],[w,B]))$,

  $j = level([u,B])$ and

  $i = Val([v,B],[w,A])$ *)

  $OT_{i,j}([u,B]) := OT_{i,j}([u,B]) \cup \{([v,B],[w,A])\}$

  $S(OT_{i,j}([u,B])) :=$

  $\min\{S(OT_{i,j}([u,B])), level(nca([v,B],[w,B]))\}$.

During Phase $l+1$, for all $i$ and $j$ such that $i-j = l$, the edges in $OT_{i,j}$ are considered group by group.

We say that the edges in a set $OT_{i,j}([u,B])$ are of <u>type</u> $[u,B]$.

Note that for fixed $[u,B]$, the sets $\mathcal{O}_{ij}([u,B])$ are considered in strictly increasing order with respect to the first index $i$.

<u>Consideration of the set $\mathcal{O}_{ij}([u,B])$</u>

In dependence of the size of level$([u,A])$ we distinguish two cases.

<u>Case 1:</u> level$([u,A]) > \ell$.

This means that at the beginning of Phase $\ell+1$, level$([u,A])$ is not defined.

- $[u,A]$ is inserted into Level $\ell+1$.
- For each $([v,B],[w,A]) \in \mathcal{O}_{ij}([u,B])$ with

  level$(\text{nca}([v,B],[w,A])) <$ level$[u,B]$

  insert the indexed edge $([v,B],[w,A])_{[u,A]}$ into $E_M$.
- Open the initially empty Queue

  Queue$([u,B])$.
- ~~To indicate that no edge in Queue$([u,B])$ is inserted into any $\mathcal{O}_{i',j'}$, $j'<j$ open a Boolean variable $B([u,B])$ with initial value 0.~~ not needed

At the moment when an edge is taken from Queue$([u,B])$ and inserted into an $OT_{i';j'}$, the variable is set to the value 1.

- Possibly, the node $[u,A]$ is also inserted into Level $\ell+1$ during Part 1 of Phase $\ell+1$ because of some edges $([v',B],[u,A])$.

At the moment when all edges inserted into $\bar{E}_M$ because of the definition of level$([u,A])$ do not fulfill the edge properties, the first $OT_{i';j}([u,B])$ in

$$\text{Queue}([u,B])$$

is taken from the queue and inserted into

$$OT_{i';j'}$$

where $j' = \text{level}(\text{DOM}([u,B]))$.

For the decision of that moment let

$D_1([u,B]) :=$

$$\min\{\text{level}(\text{nca}([v',B],[u,B])) \mid ([v',B],[u,A]) \in \bar{E}_M\}$$

$D_2([u,B]) :=$

$$\min\{\text{level}(\text{nca}([v,B],[w,B])) \mid ([v,B],[w,A])_{[u,A]} \in \bar{E}_M\}$$

and

$D([u,B]) := \min \{ D_1([u,B]), D_2([u,B]) \}$.

$k := i - D([u,B])$ denotes the first level such that no edge which is inserted into $E_M$ because of the definition of level$([u,A])$ fulfills the edge properties with respect to Phase~~Level~~ $k$.

We need this knowledge at the beginning of Part 2 of Phase $k$. Hence we have an array $Q$ such that we can store this information into $Q[k]$.

Case 2: level$([u,A]) < \ell$.

This means that at the beginning of Phase $\ell+1$, level$([u,A])$ is already defined.

**if** $S(\mathcal{O}_{ij}([u,B])) < D([u,B])$

**then**

$\mathcal{O}_{ij}([u,B])$ is added to the end of Queue$([u,B])$

**else**

$\mathcal{O}_{ij}([u,B])$ is deleted.

**fi**.

This finishes the description of "Consideration of the set $\mathcal{O}_{ij}([u,B])$."

As mentioned above, at the beginning of Part 2 of Phase $\ell+1$, we have to check

$$Q[\ell+1].$$

For all $[u,B]$ such that $Q[\ell+1]$ contains the information the first $\mathcal{O}_{i',j}([u,B])$ in Queue$([u,B])$ has to be taken from the queue and to be inserted into

$$\mathcal{O}_{i',j'} \quad \text{where } j' = \text{level}(\text{DOM}([u,B])),$$

we do it.

It remains the analysis of the total used for the update of the data structure $\mathcal{O}$.

Goal: To prove that this time is bounded by $O(m)$.

For all edges $([v,B],[w,A])$ the following is fulfilled:

i) $\text{Val}([v,B],[w,A])$ is computed at most once.

ii) $([v,B],[w,A])$ is stored into a set $\mathcal{O}_{i,j}([u,B])$ at most once.

$\Longrightarrow$

The total time for doing this is $O(m)$.

Let us analyze the time used for

"Consideration of the set $\mathfrak{O}_{ij}([u,B])$".

<u>Case 1</u>:

- Each edge in $\mathfrak{O}_{ij}([u,B])$ <u>is</u> checked and either inserted into $\overline{E}_M$ or not inserted into $\overline{E}_M$. After doing this, the edge is never considered again

$\Longrightarrow$

$$\text{time} \quad O(|\mathfrak{O}_{ij}([u,B])|).$$

- Opening Queue$([u,B])$ can be done in constant time.

- Computation of $D_1([u,B])$

  time $O(|\{([v;B],[u,A]) \mid ([v;B],[u,A]) \in \overline{E}_M\}|)$

  Computation of $D_2([u,B])$

  time $O(|\mathfrak{O}_{ij}([u,B])|)$

Total time $O(m)$

<u>Case 2</u>:

The set $\mathfrak{O}_{ij}([u,B])$ is

- either completely deleted
- or added to Queue$([u,B])$.

total time of the set deletion $O(m)$

It remains the analysis of the time spent on the update of the queues.

13.06.

Each edge $([v,B],[w,A])$ which enters the data structure at the moment when $Val([v,B],[w,A])$ is computed, is inserted in an unique set

$$\mathcal{O}_{ij}([u,B])$$

where

$$[u,B] = DOM(nca([w,A],[w,B]))$$
at the moment when $Val([v,B],[w,A])$ is computed

$$j = level([u,B]) \quad \text{and}$$
$$i = Val([v,B],[w,A]).$$

At that moment, the type $[u,B]$ of the edge $([v,B],[w,A])$ is defined. which never changes.

As long as the edges in $\mathcal{O}_{ij}([u,B])$ do not disappear from the data structure, we only consider the whole set in one step and not each single edge in $\mathcal{O}_{ij}([u,B])$.

Defining $level([u,A])$, we have to store all types of the edges which are inserted as an indexed edge into $\overline{E}_M$.

At the beginning of Phase $l+1$, $Q[l+1]$ is checked. If we get the information that

$$l+1 = i - D([u,B])$$

for a node $[u,B]$, we have not only to insert the first set on Queue $([u,B])$ into the data structure OT.

We have also to insert the first set on Queue $([z,B])$ for all $[z,B]$ such that $[z,B]$ is the type of an indexed edge $(\sim,\sim)_{[u,A]}$ in $\overline{E}_M$.

But this implies that at least one edge of type $[z,B]$ disappears.

$\Rightarrow$ total time for set insertions queues $\rightsquigarrow$ data structure OT

$O(m)$.

Such a set which is inserted into OT can leave OT and enter a queue again.

Set deletions can be counted against set insertions

$\Rightarrow$

total time for set deletions OT $\rightsquigarrow$ queues $O(m)$.