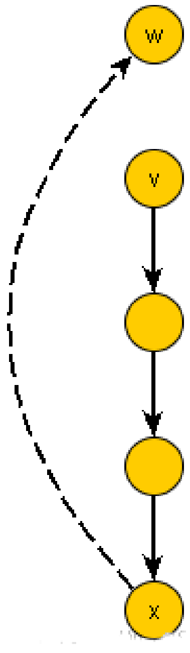
st-numbering

Prompt: Low(v)

LOW[v] =MIN({v} U {w | there exists a back edge $\{x, w\} \in B$

- such that x is a descendant of v,
- and w an ancestor of v in the depth
- first spanning forest (V, T)})

(1)



By Lemma 2, if vertex v is not the root, then v is an articulation point if and only if v has a son s such that $\mathrm{LOW}[s] \geq v$.

Reformulation for biconneced graphs?

st-orientation

Let $G = (V, E)$ be an undirected biconnected graph of n nodes and m edges.

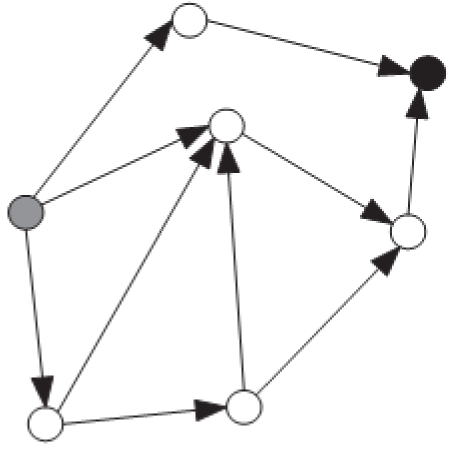
There are different algorithms for orienting the edges of G.

In fact, there are 2^m ways to achieve this.

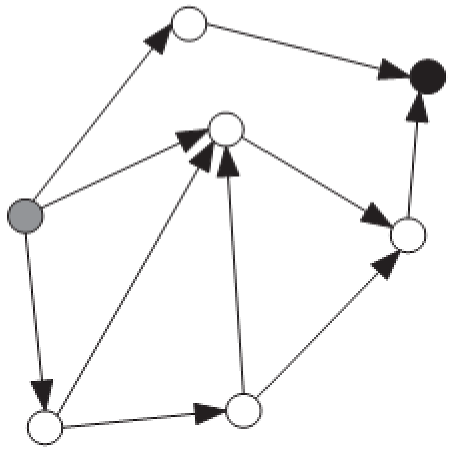
However, it is very useful in many applications to be able to produce st-oriented directed graphs which satisfy two distinct properties:

1. They have 1 single source s and 1 single sink t
2. They contain no cycles

Such an orientation of G's edges is called an *st*-orientation or a bipolar orientation



st-orientation

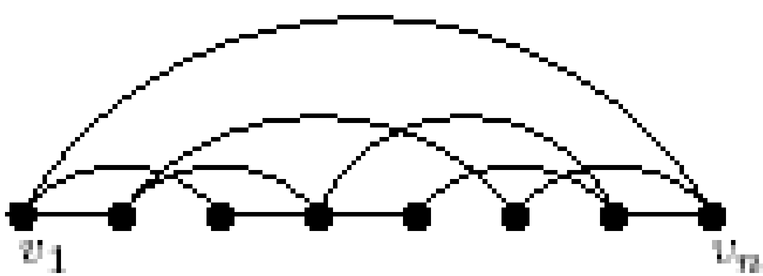


st-oriented graphs have many interesting properties.
First of all, we can run several polynomial time algorithms on them (for example longest path and topological sorting) and draw some useful conclusions.
But how can we compute an *st*-orientation?
Do all undirected graphs admit such an orientation?

In 1967, Lempel, Even and Cederbaum made a first approach to this problem, by presenting an algorithm for the computation of a numbering of the vertices of an undirected graph in order to check whether a graph is planar or not.

They proved that, given any edge $\{s, t\}$ of a biconnected graph G, the vertices of G can be numbered from 1 to n, so that

- vertex s receives number 1,
- vertex t receives number n
- and all other vertices are adjacent
- both to a lower-numbered and to a higher-numbered vertex.



Actually, an undirected graph $G = (V, E)$ can be *st*-numbered $\Leftrightarrow$ the graph $G' = (V, E \cup (s, t))$ is biconnected.

It is easy to prove that G has an st-orientation if and only if it has an st-numbering and we can compute either from the other in $O(m+n)$ time, as follows.

Given an st-orientation, we number the vertices of G in topological order.

This produces an st-numbering.

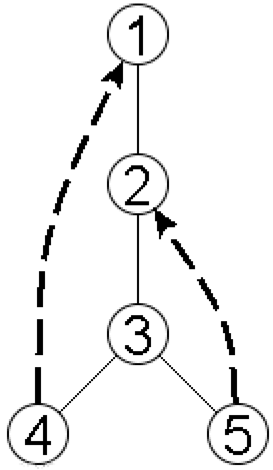
Given an st-numbering, we orient each edge from its lower-numbered to its higher-numbered endpoint.

This produces an st-orientation.

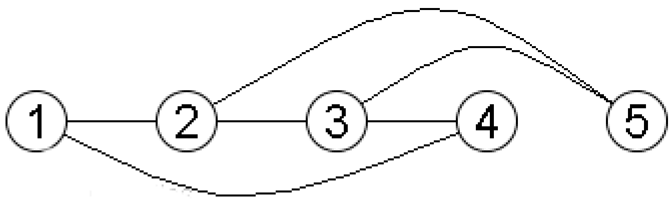
st-numbering: Example

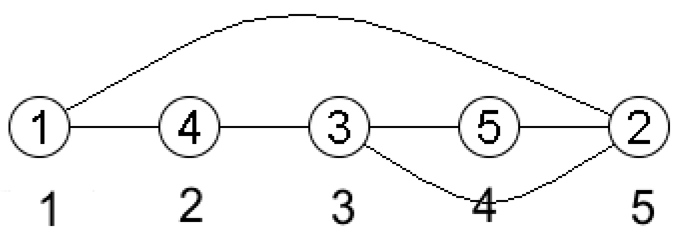
1) Is this graph biconnected?

2) Is its numbering an st-numbering?



st-numbering: Example





12-numbering

Simple algorithm for the computation of an *st*-numbering (Tarjan 1986)

The algorithm is based on a depth-first search traversal of the initial biconnected graph.

The algorithm consists of two passes.

The first pass is a DFS(s).

During DFS(s) the vertex s receives number 1.

During DFS(s) the vertex t receives number 2.

Also, for each vertex $v \in V$

v – dfs_number(v);

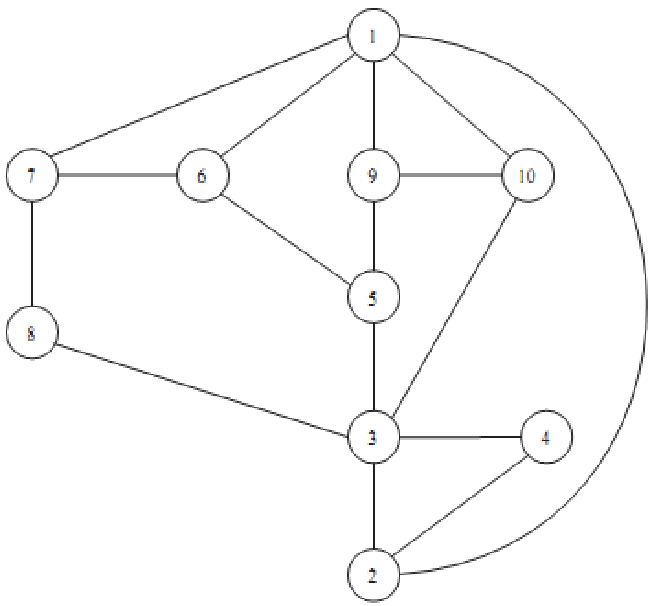
$low(v)$ –lowpoint (v);

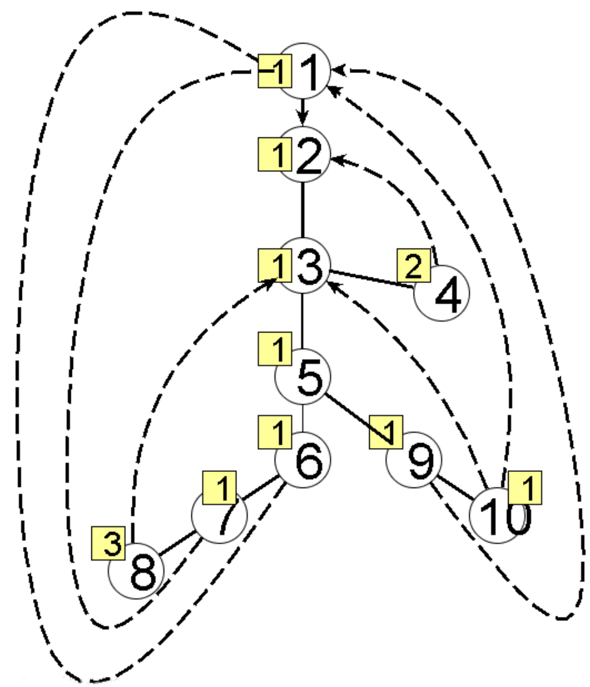
$p(v)$ – father of v

are computed.

Example

Suppose we want to compute a 2-1 numbering of the biconnected graph shown bellow.
First we execute a DFS, and we compute the DFS tree and the LOW values.





Simple algorithm for the computation of an *st*-numbering (second pass)

The second pass constructs a list *L* of the vertices, such that if vertices are numbered in the order they occur in *L*, an st-numbering results.

Actually, the second pass is a preorder traversal of the spanning tree.

Initialization: L = {s, t}; sign(s) = “-”;

The second pass of the algorithm consists of repeating the following step for each vertex $v \neq s, t$ in preorder:

1. **if** sign(low(v)) == “ + ” **then**
2. Insert *v* after *p*(*v*) in *L*
3. sign(*p*(*v*)) = “ - ” ;
4. **end if**
5. **if** sign(low(v)) == “-” **then**
6. Insert v before *p*(*v*) in L
7. sign(*p*(*v*)) = “ + ”;
8. **end if**

Example of st-numbering

L = [1^-, 2]

Vertex 3:

$Low(3) = 1$,

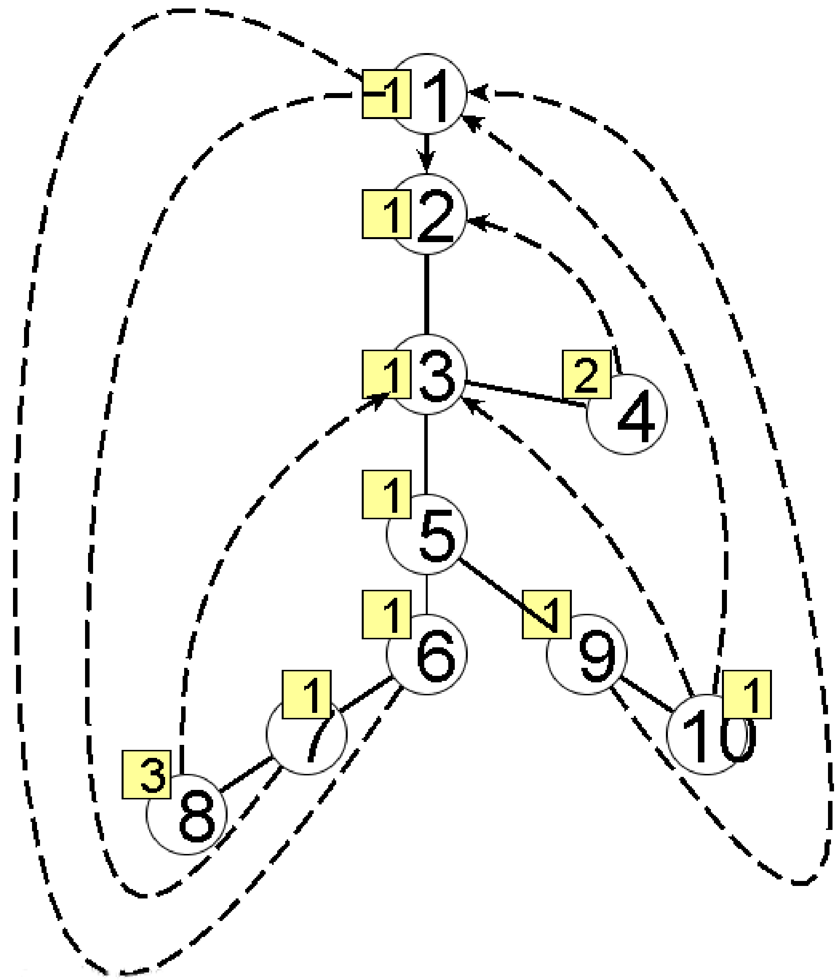
$sign(1)$ = "-"

$P(3) = 2$

Insert 3 before 2 in L,

sign(2):="+"

L = [1^-,3, 2^+],



Пример построения st-нумерации

L = [1^-, 3, 2^+]

Vertex 4:

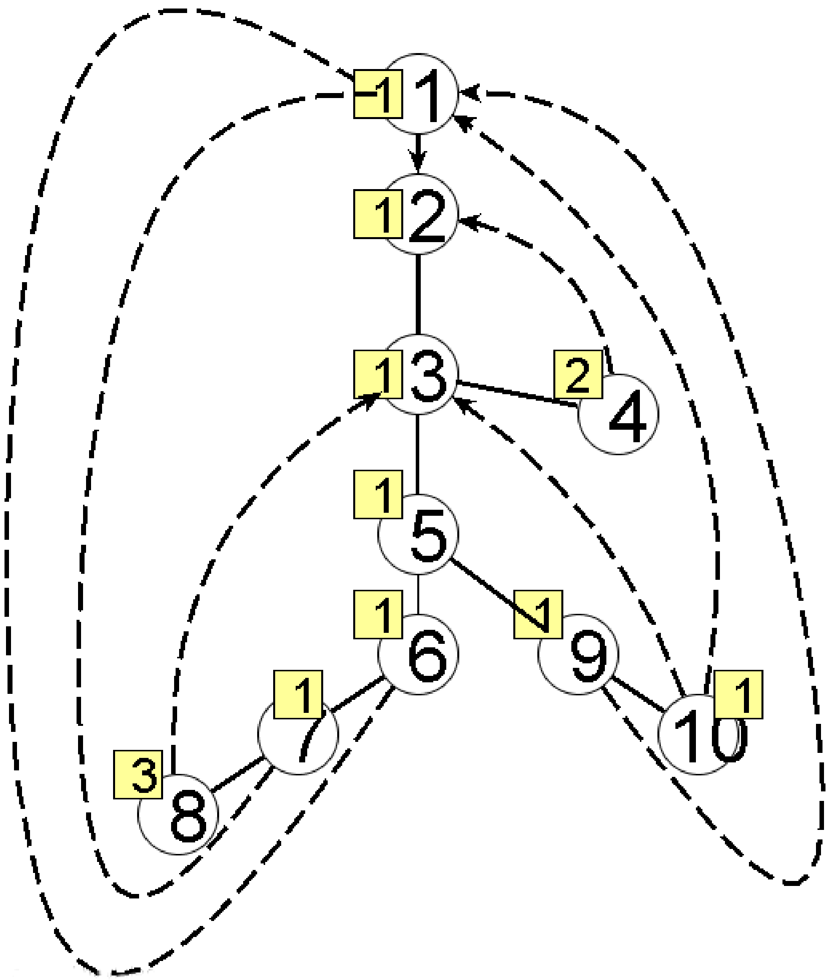
Low(4) = 2, sign(2) = "+"

P(4) = 3

Insert 4 after 3 in *L*,

sign(3):="-"

L = [1^-, 3^-, 4, 2^+]



Example of st-numbering

L = [1^-, 3^-, 4, 2^+]

Vertex 5:

$Low(5) = 1$,

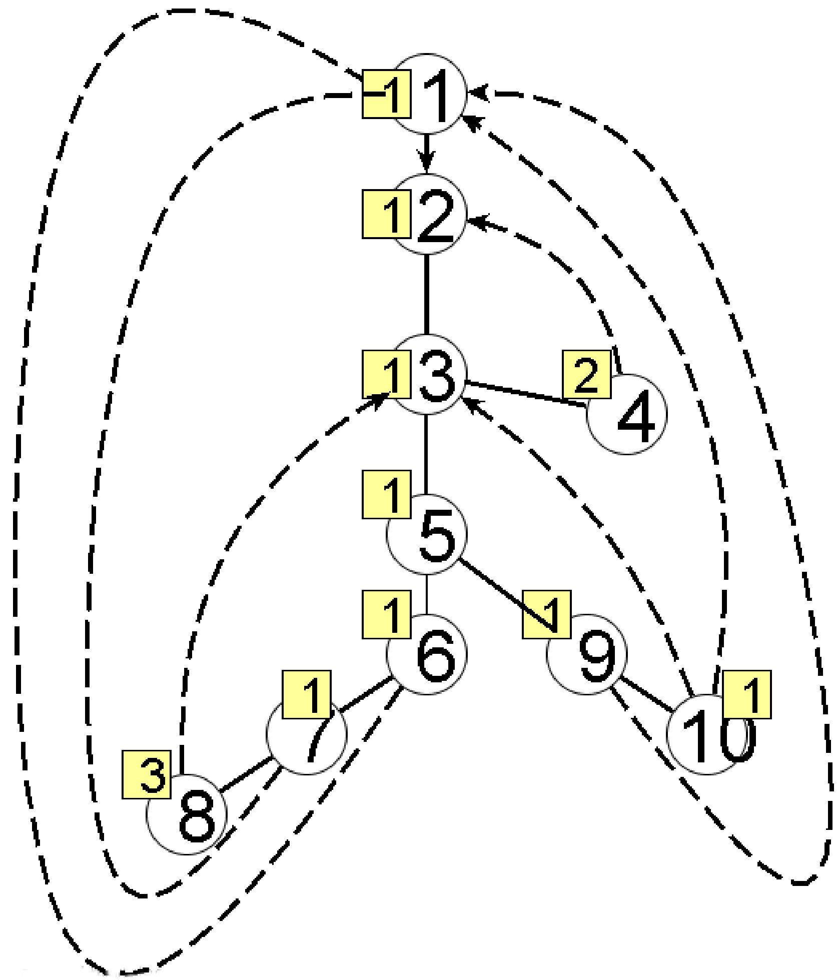
$sign(1)$ = "-"

$P(5) = 3$

Insert 5 before 3 in L,

$sign(3)$:="+"

L = [1^-, 5, 3^+, 4, 2^+]



Example of st-numbering

$L = [1^-, 5, 3^+, 4, 2^+]$

Vertex 6:

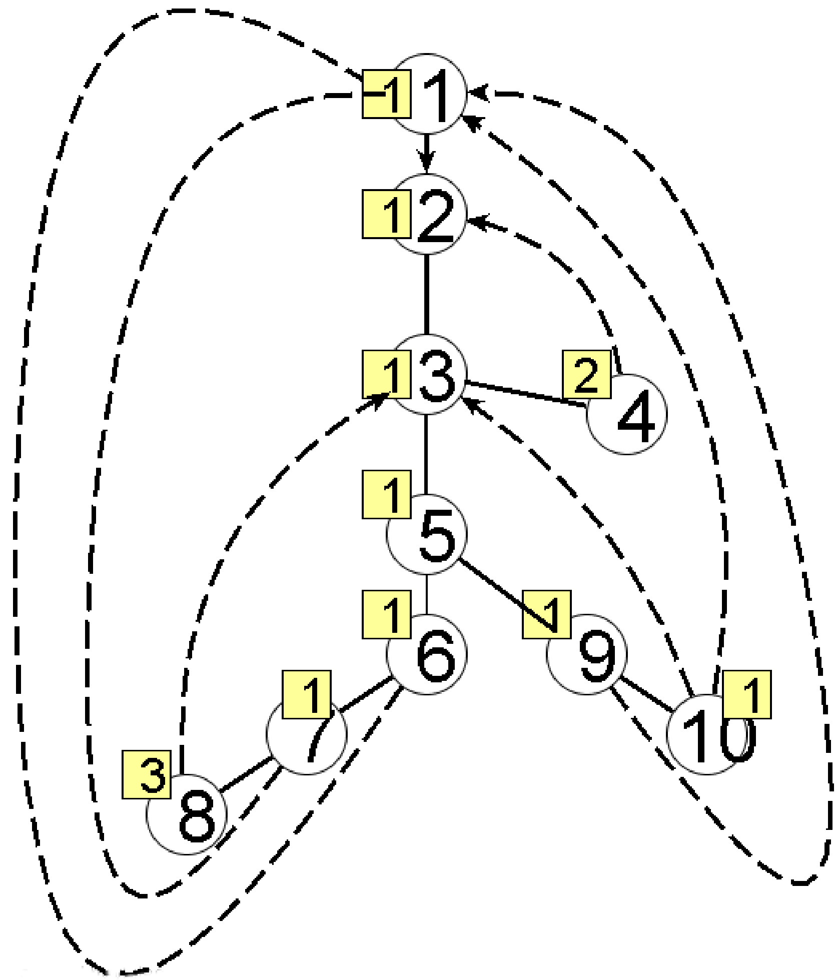
$Low(6) = 1$, sign(1) = "-"

$P(6) = 5$

Insert 6 before 5 in L,

$sign(5)$:="+"

$L = [1^-, 6, 5^+, 3^+, 4, 2^+]$



Example of st-numbering

L = [1^-, 6, 5^+, 3^+, 4, 2^+]

Vertex 7:

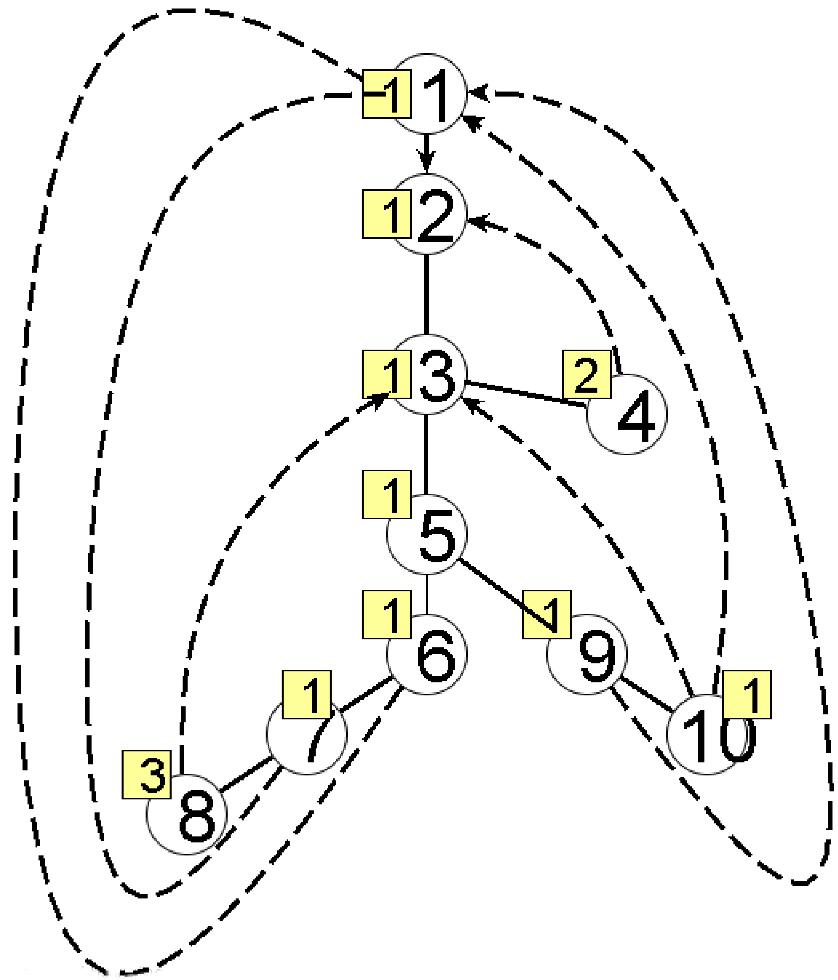
$Low(7) = 1$, sign(1) = "-"

$P(7) = 6$

Insert 7 **before** 6 in L,

$sign(6)$:="+"

L = [1^-, 7, 6^+, 5^+, 3^+, 4, 2^+]



Example of st-numbering

$L = [1^-, 7, 6^+, 5^+, 3^+, 4, 2^+]$

Vertex 8:

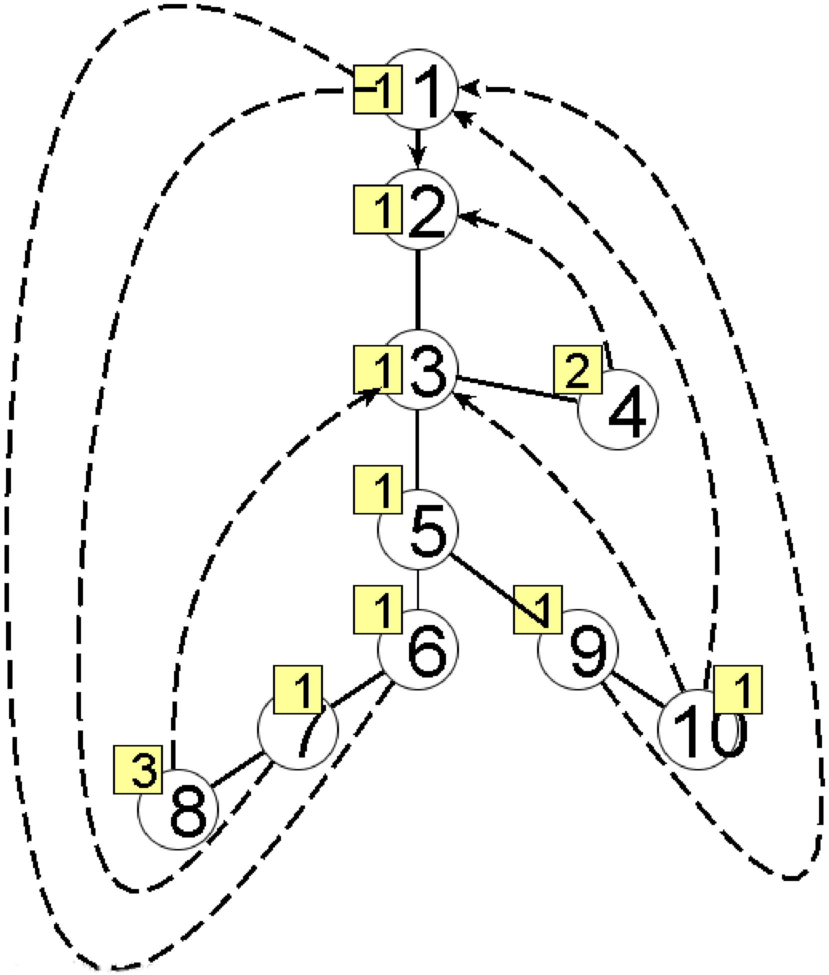
$Low(8) = 3$, sign(3) = “+”

$P(8) = 7$

Insert 8 after 7 in L,

$sign(7)$:=“-”

$L = [1^-, 7^-, 8, 6^+, 5^+, 3^+, 4, 2^+]$



Example of st-numbering

L = [1^-, 7^-, 8, 6^+, 5^+, 3^+, 4, 2^+]

Vertex 9:

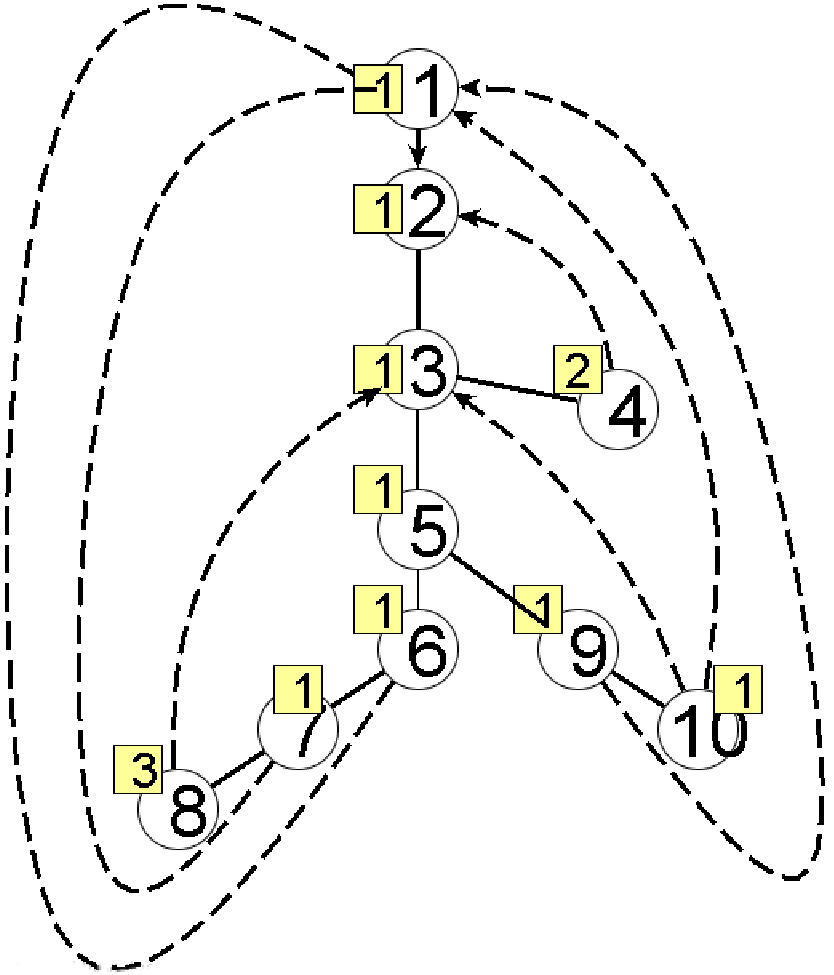
Low(9) = 1, sign(1) = "-"

P(9) = 5

Insert 9 before 5 in *L*,

sign(5):="+"

L = [1^-, 7^-, 8, 6^+, 9, 5^+, 3^+, 4, 2^+]



Example of st-numbering

$L = [1^-, 7^-, 8, 6^+, 9, 5^+, 3^+, 4, 2^+]$

Vertex 10:

Low(10) = 1,

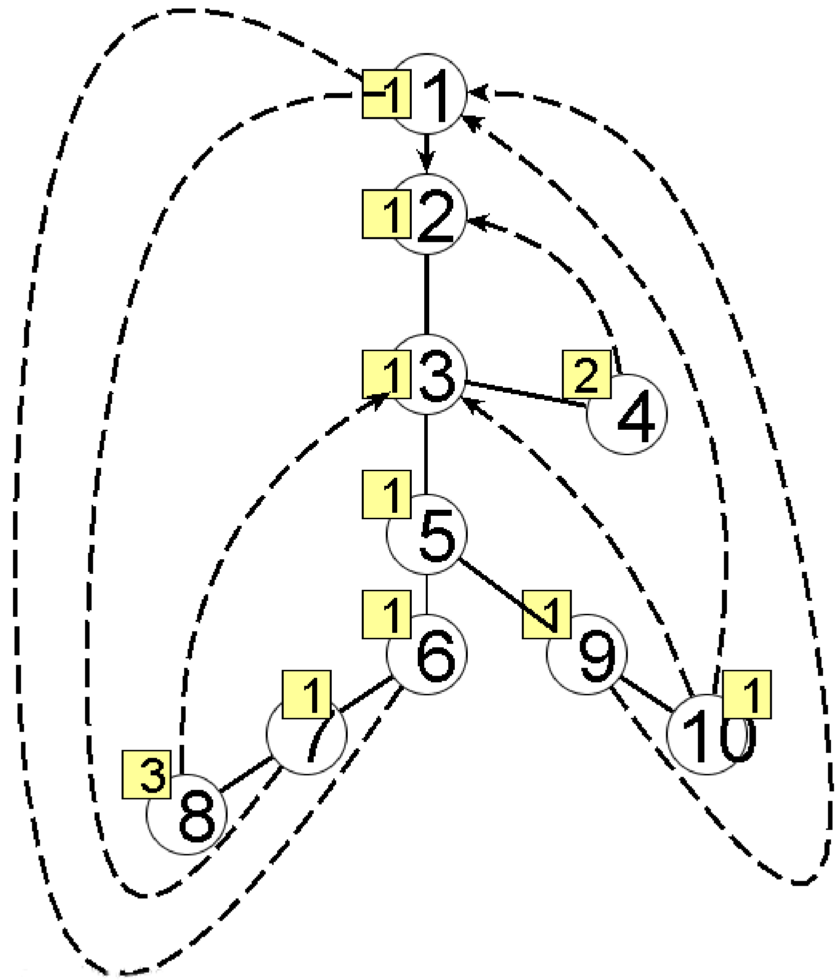
sign(1) = "-"

P(10) = 9

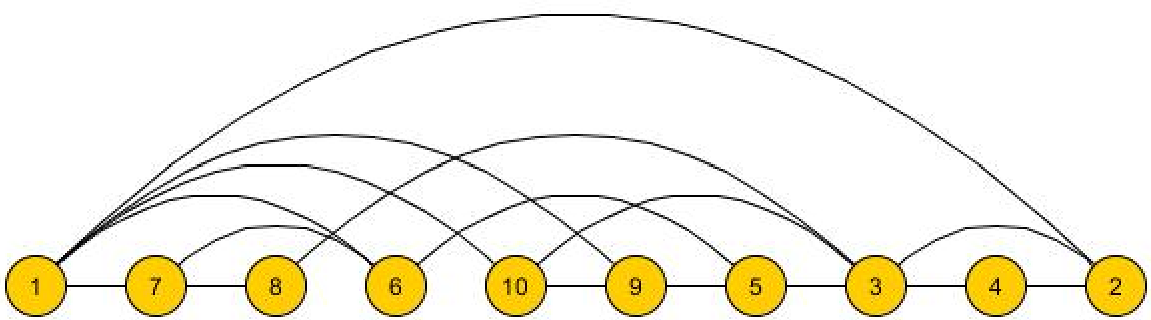
Insert 10 before 9 in *L*,

sign(9):="+"

$L = [1^-, 7^-, 8, 6^+, 10, 9^+, 5^+, 3^+, 4, 2^+]$



The final st-numbering



st-numbering

Theorem. The st-numbering is correct.

Proof. Consider the second pass of the algorithm.

We must show that:

1. The signs assigned to the vertices have the claimed meaning
2. If vertices are numbered in the order they occur in L, an st-numbering results.

st-numbering

For (1) suppose $s = x_0$, $t = x_1, x_2 \ldots x_l$ be the tree path from s to the vertex x_l most recently added to L

and let *v* with parent x_k be the next vertex to be added to *L*.

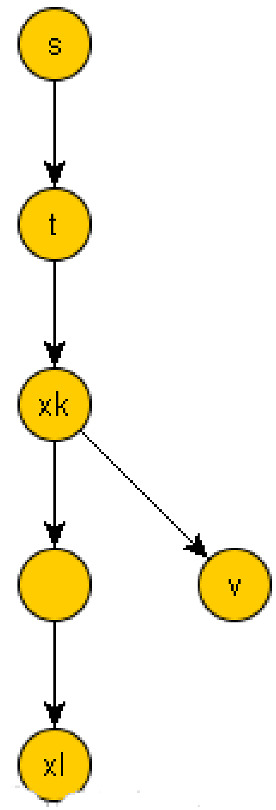
Assume as an induction hypothesis that for

all $0 \leq i < j < l$, $sign(x_i)$ = "+" if and only if x_i follows x_j in *L*,

i.e., $x_i = p(x_j)$.

Since sign (x_k) is set to "-" if *v* is inserted after x_k in *L* and to "+" if *v* is inserted before x_k in *L*, the induction hypothesis holds after *v* is added.

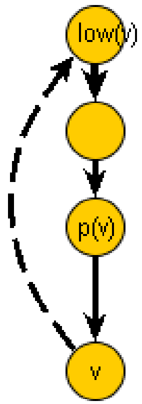
Hence the induction holds.



$[x_j, x_i^+]$

st-нумерация

(2) Let v ≠ s, t. If $(v, low(v))$ is a back edge, the insertion of v between $p(v)$ and $low(v)$ in L guarantees that in the numbering corresponding to L, v is adjacent to both a lower-numbered and a higher-numbered vertex.



$[low(v)^{-}, v, p(v)]$

$[p(v), v, low(v)^{+}]$

st-нумерация

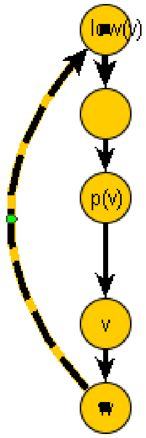
Otherwise, there must be a vertex w such that $p(w) = v$ and $low(w) = low(v)$.

By lemma 2.1 we have that $low(v)$ is a proper ancestor of v, which means that sign ($low(v)$) remains constant during the time v and w are added to L.

It follows that v appears between $p(v)$ and w in the completed list L,

Which implied that in the numbering corresponding to L, v is adjacent to both a lower-numbered and higher-numbered vertex.

Thus, the second case holds.



$[low(v)^{-}, w, v, p(v)]$

$[p(v), v, w, low(v)^{+}]$