Введение в дискретную математику и математическую логику

•

Лекция №8
st-нумерация

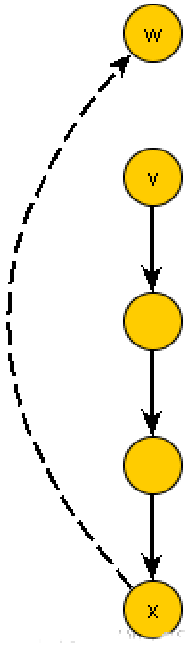
Апанович Зинаида Владимировна

Напоминание: Low(v)

$v.Low$ = MIN({v} U { w|существует обратное ребро { x, w } $\in B$ такое, что

x является потомком v,

а w предком v в глубинном остовном лесу (V, T)}) (1)



Low(v) и двусвязность

По лемме 2, если вершина v *не* является корнем, то v является точкой сочленения ⇔ у v есть сын s такой что $s.Low \geq v$.

Как переформулировать это утверждение для двусвязных графов?

st-ориентация

Пусть $G = (V, E)$ — неориентированный двусвязный граф, имеющий n вершин и m ребер.

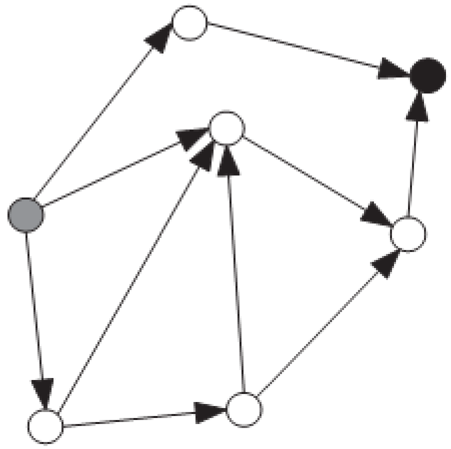
Существуют различные алгоритмы ориентации рёбер *G.*

На самом деле, есть 2^m способов этого добиться.

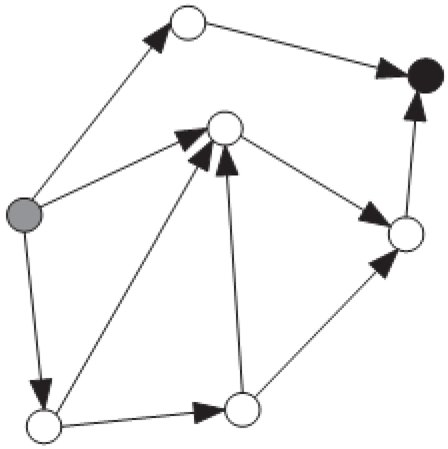
Однако во многих приложениях очень полезно иметь возможность создавать st-ориентированные ориентированные графы, которые удовлетворяют двум различным свойствам:

1. Они имеют единственный исток и единственный сток .
2. Они не содержат циклов.

Такая ориентация ребер графа *G* называется *st* -ориентацией или биполярной ориентациеi



st-ориентация



st-ориентированные графы обладают многими интересными свойствами. Прежде всего, мы можем запустить на них несколько алгоритмов с полиномиальным временем выполнения (например, поиск самого длинного пути и топологическую сортировку) и сделать некоторые полезные выводы.
Но как мы можем вычислить *st*- ориентацию ?
Все ли неориентированные графы допускают такую ориентацию?

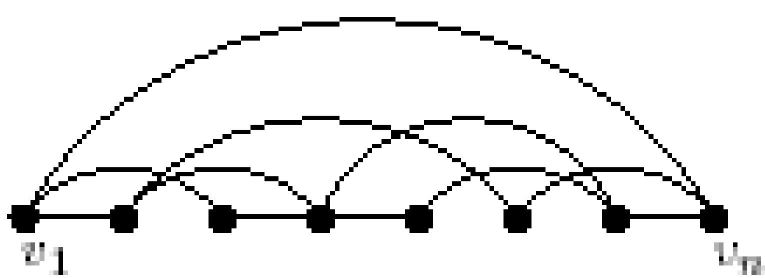
st-нумерация

В 1967 году Lempel, Even и Cederbaum впервые подошли к решению этой проблемы, представив алгоритм вычисления

st - нумерации вершин неориентированного графа с целью проверки того, является ли граф планарным или нет.

Они доказали, что для любого ребра { *s, t* } двусвязного графа *G* вершины *G* можно пронумеровать от 1 до *n* , так что

- вершина *s* получает номер 1 ,
- вершина *t* получает номер *n*
- все остальные вершины смежны как с вершиной с меньшим номером, так и с вершиной с большим номером.



st-нумерация и двусвязность

На самом деле, неориентированный граф $G = (V, E)$ имеет st-*нумерацию* $\Leftrightarrow$ граф $G' = (V, E \cup (s, t))$ двусвязен.

st –ориентация и *st* -нумерация

Легко доказать, что *G* имеет *st* -ориентацию ⇔ он имеет *st* -нумерацию , и мы можем вычислить любую из них из другой за время

$O(m + n)$, как показано ниже.

=>Для данной *st* -ориентации мы нумеруем вершины *G* в топологическом порядке.

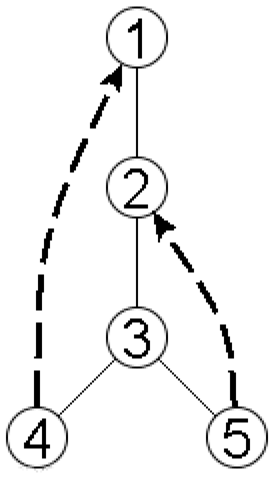
Это создает *st* -нумерацию.

<= Для данной *st* -нумерации, мы ориентируем каждое ребро от его концевой вершины с меньшим номером к его вершине с большим номером.

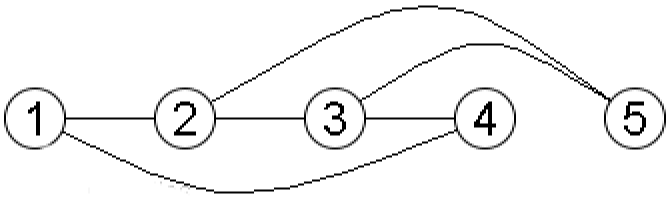
Это создает *st* -ориентацию.

st - нумерация : Пример

1) Является ли показанный граф двусвязным?
2) Является ли его нумерация *st* - нумерацией ?



st - нумерация : Пример



Мы можем перенумеровать вершины следующим образом:

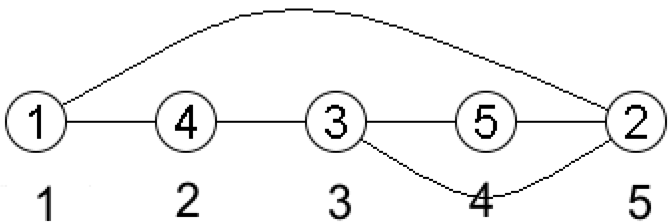
1 -> 1

2 -> 5

3 -> 3

4 -> 2

5 -> 4



12- нумерация

Простой алгоритм вычисления *st* -нумерации (Тарьян 1986)

Алгоритм основан на поиске в глубину исходного двусвязного графа.

Алгоритм состоит из 2 проходов.

Первый проход — это DFS(s).

В ходе DFS(s) вершина s получает номер 1 .

В ходе DFS(s) вершина t получает номер 2 .

Также для каждой вершины $v \in V$ вычисляются:

v – dfs_number(v) ;

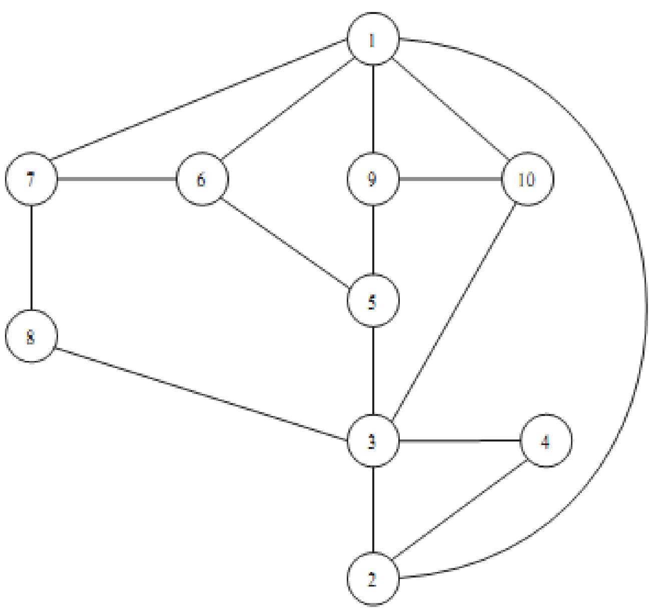
$v.Low$ – нижняя точка (v);

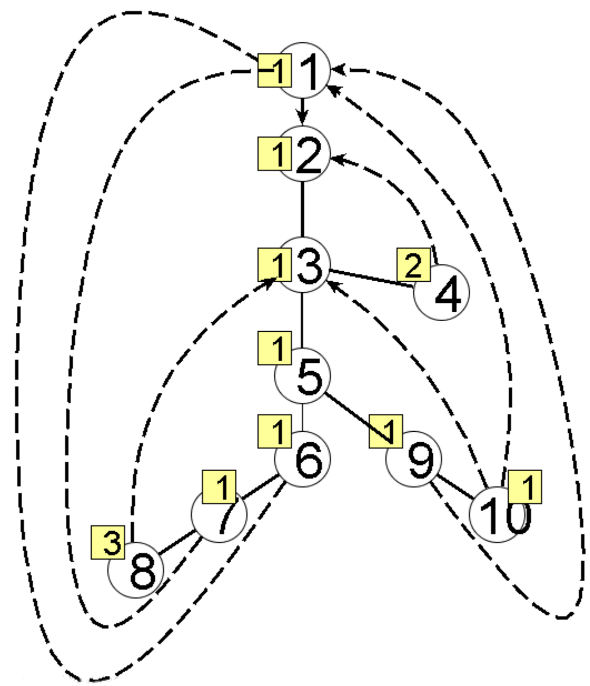
$v.p$ – отец вершины v

.

st –нумерация. Пример

Предположим, мы хотим вычислить 2-1-нумерацию для двусвязного графа, показанного ниже.
Сначала мы выполняем DFS, вычисляем дерево DFS и значения *v.Low, v.p.*





Простой алгоритм вычисления *st* -нумерации (второй проход)

Второй проход создает список *L* вершин, причем если вершины пронумерованы в том порядке, в котором они встречаются в *L*, то получается st-нумерация .

На самом деле второй проход представляет собой предварительный обход остовного дерева.

Инициализация : $L = \{ s , t \}$; *знак* (*s*) = « - »;

Второй проход алгоритма состоит из повторения следующего шага для каждой вершины $v \neq s, t$ в прямом порядке :

1. **если** знак (*v.Low*) == “ + ” **то**
2. Вставить *v* после *v.p* в *L*
3. знак(*p*(*v*)) = “ - ” ;
4. **конец если**
5. **если** знак(*v.Low*) == «-» **, то**
6. Вставить *v* перед *v.p* в *L*
7. знак(*v.p*) = « + » ;
8. **конец если**

st-нумерация Пример

L = [1$^-$, 2]

Вершина 3:

3.Low = 1,

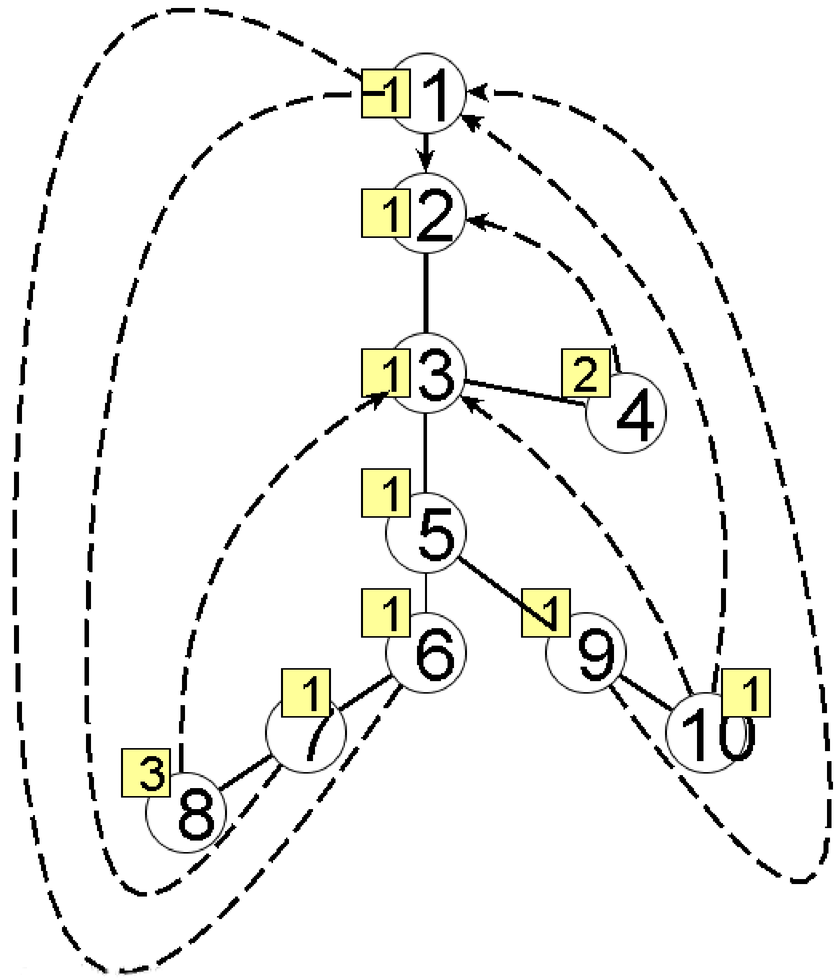
знак (1) = «-»

3.p = 2

Ставим 3 перед 2 в L,

знак(2):="+"

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



st- нумерация Пример

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

Вершина 4:

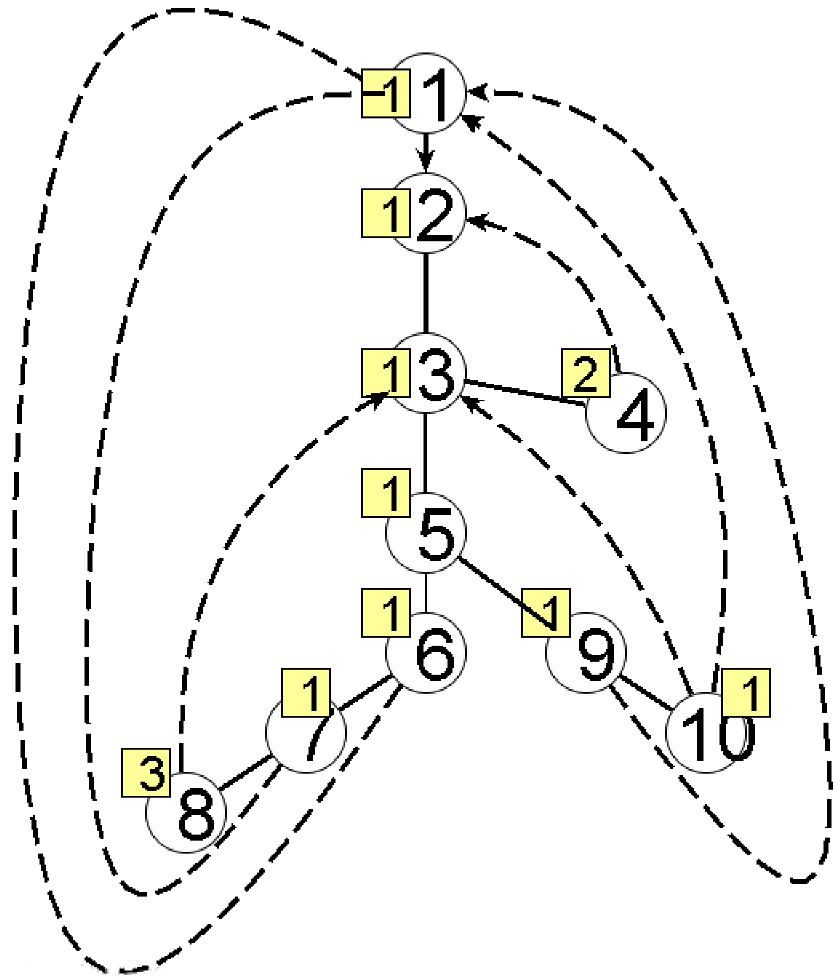
4.Low = 2, знак(2) = «+»

4.p = 3

Ставим 4 после 3 в *L*,

знак (3):=" - "

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



st- нумерация. Пример

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

Вершина 5:

5.Low = 1 ,

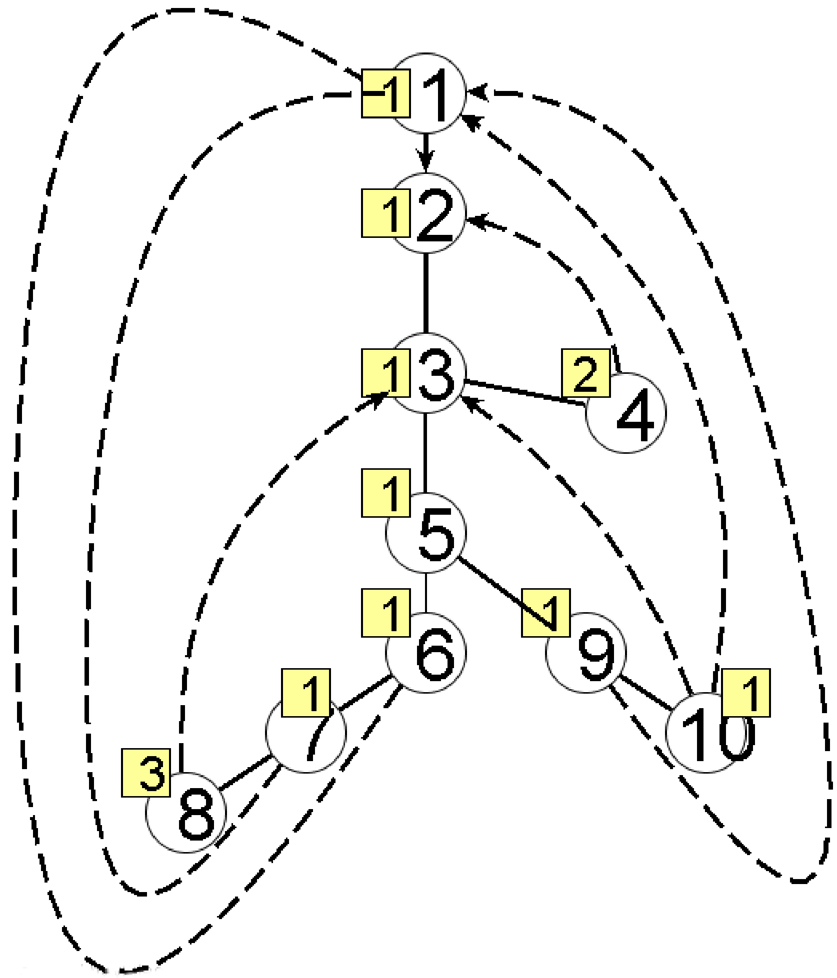
знак (1) = « - »

5.p = 3

Ставим 5 перед 3 в *L* ,

знак (3):=“ + ”

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



st- нумерация Пример

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

Вершина 6 :

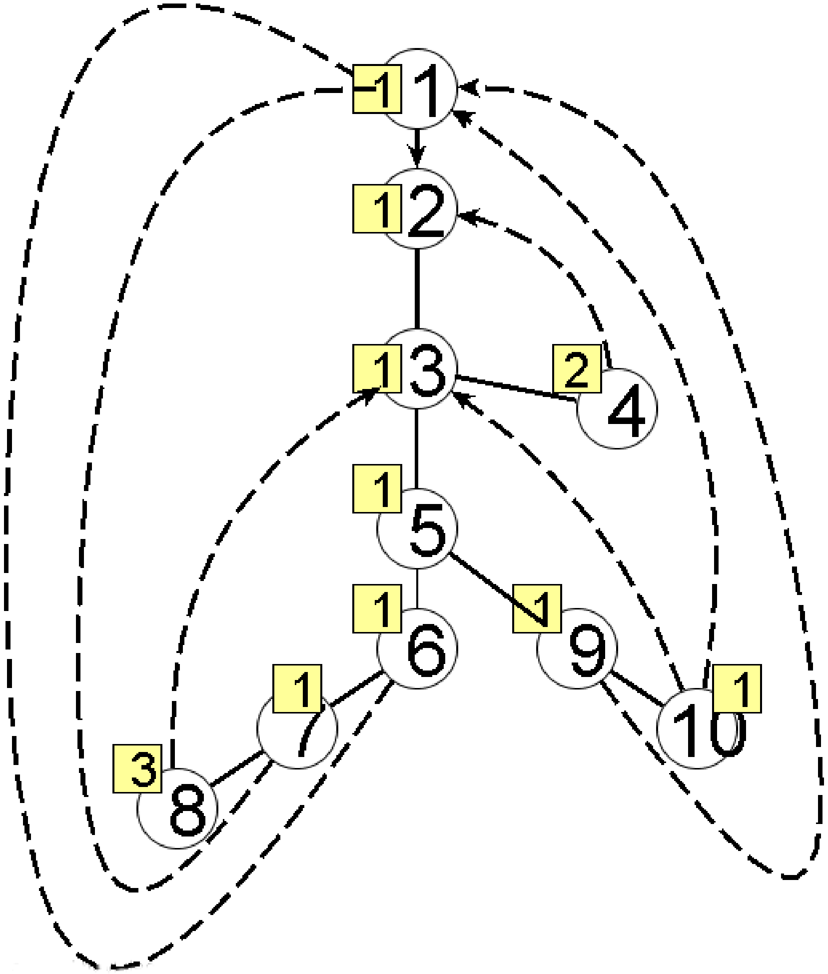
6.Low = 1 , знак(1) = « - »

6.p = 5

Ставим 6 перед 5 в *L* ,

знак (5):=“ + ”

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



st-нумерация Пример

L = [1⁻ , 6, 5⁺ , 3⁺ , 4, 2⁺]

Вершина 7:

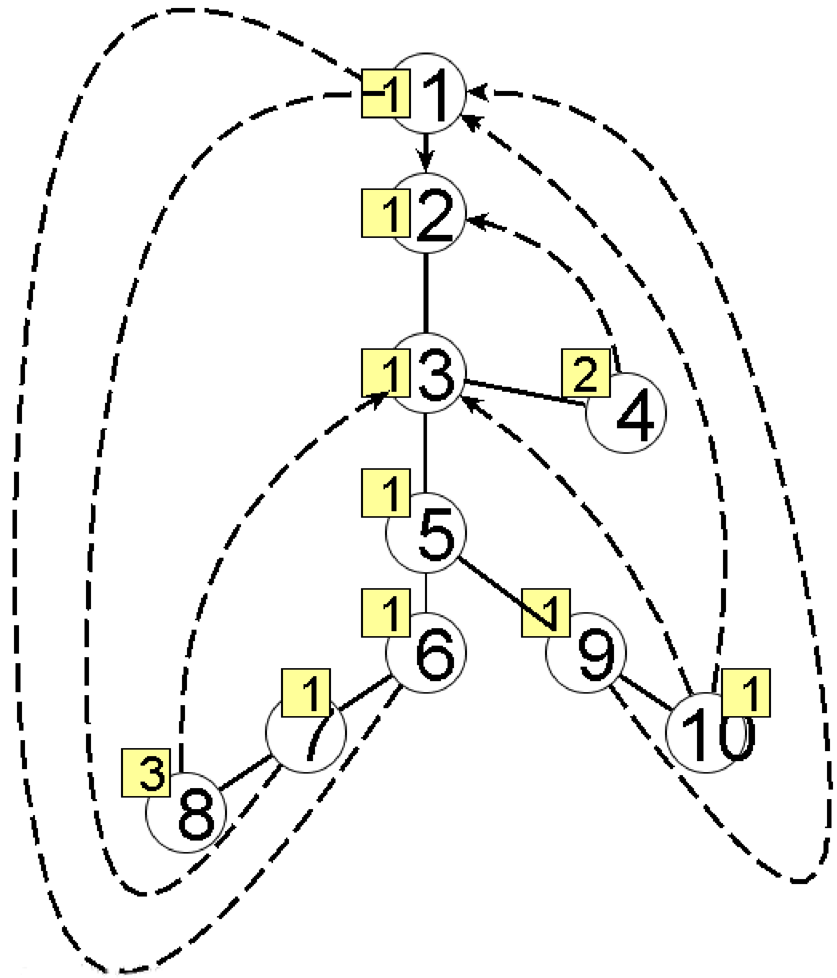
7.Low = 1 , знак(1) = « - »

7.p = 6

Ставим 7 перед 6 в *L* ,

знак (6):=“ + ”

L = [1⁻ , 7 , 6⁺ , 5⁺ , 3⁺ , 4, 2⁺]



st-нумерация Пример

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

Вершина 8:

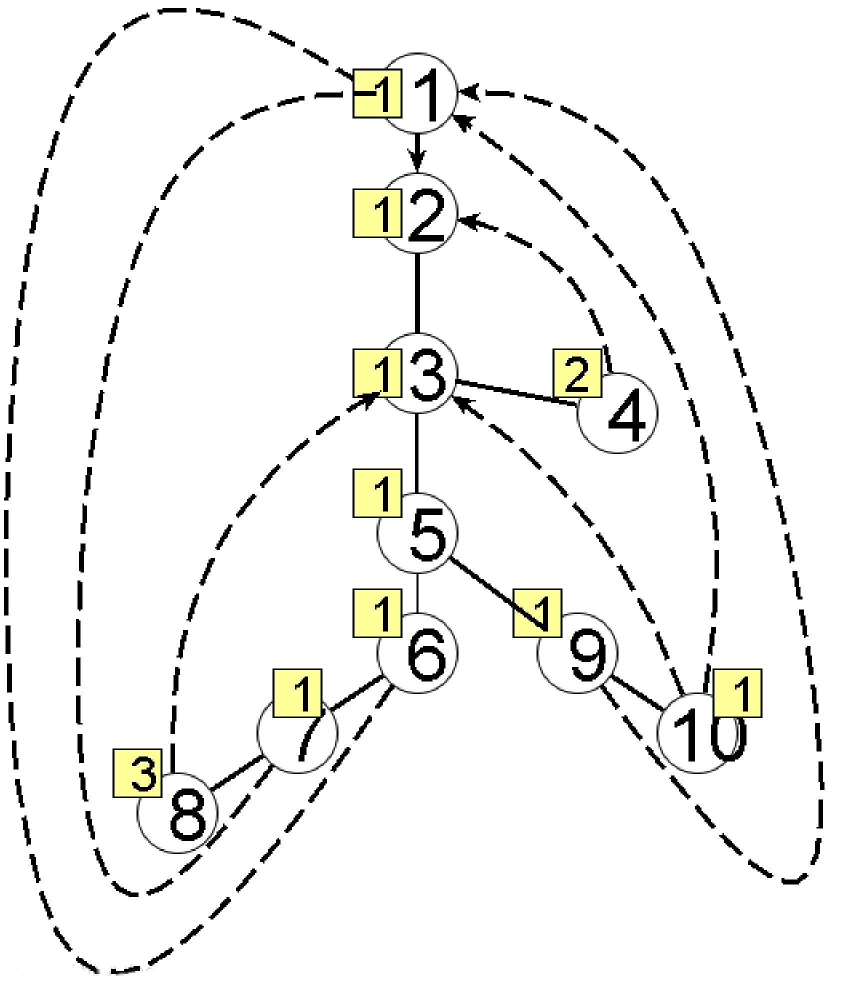
8.Low = 3 , знак(3) = « + »

8.p = 7

Ставим 8 после 7 в *L* ,

знак (7):=" - "

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



st-нумерация Пример

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

Вершина 9:

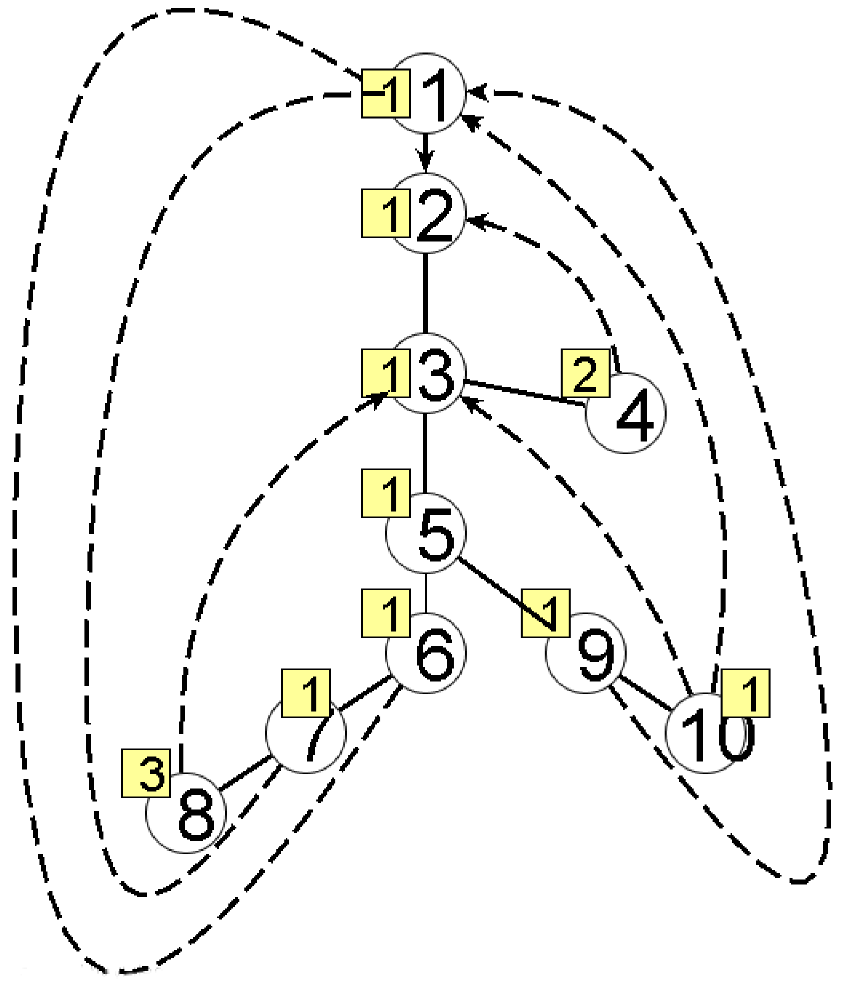
9.Low = 1 , знак(1) = « - »

9.p = 5

Вставьте 9 перед 5 в *L* ,

знак (5):=“ + ”

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



st-нумерация Пример

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

Вершина 10:

10.Low = 1 ,

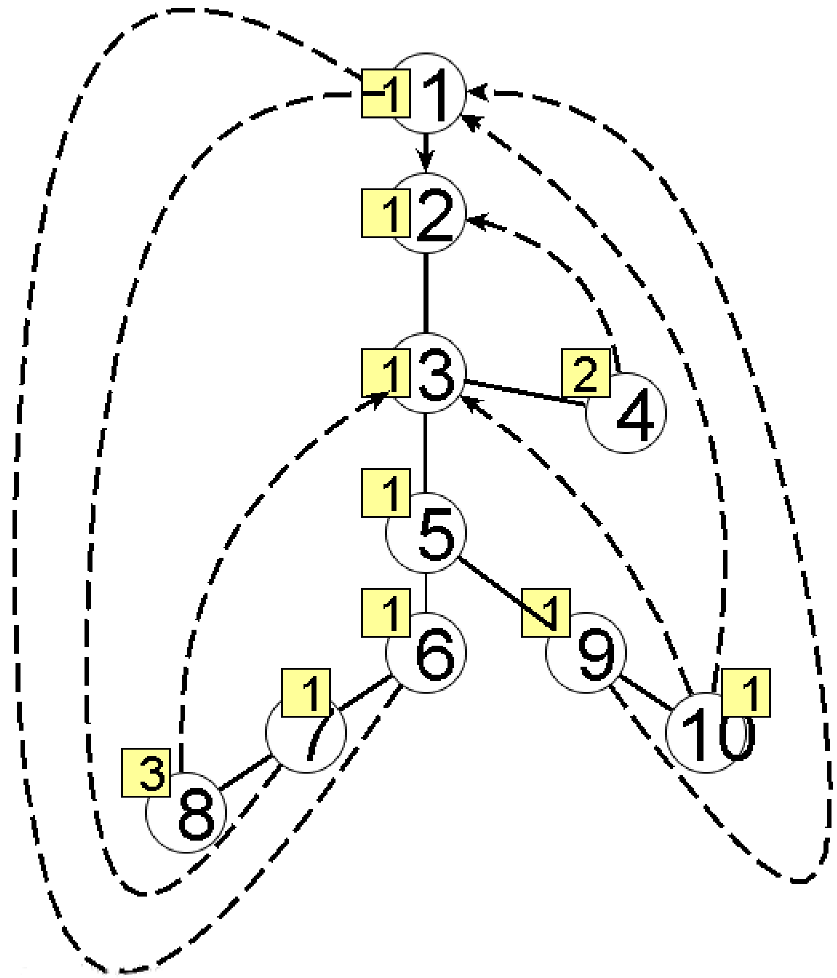
знак (1) = « - »

10.p = 9

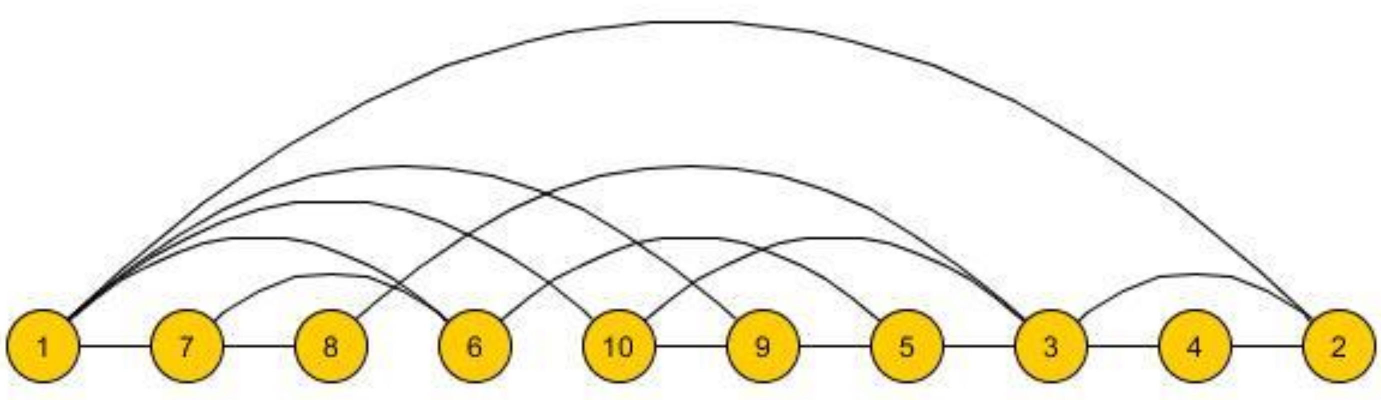
Вставьте 10 перед 9 в *L* ,

знак (9):=“ + ”

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



st-нумерация. Пример



st-нумерация

Теорема. Нумерация st корректна.

Доказательство . Рассмотрим второй проход алгоритма.

Мы должны показать, что если вершины пронумерованы в том порядке, в котором они встречаются в *L*, то получится *st* -нумерация.

st -нумерация

Для (1) предположим, что $s = x_0$, $t = x_1$, $x_2 \ldots x_l$ — путь по дереву от s до вершины x_l , последней добавленной к L.

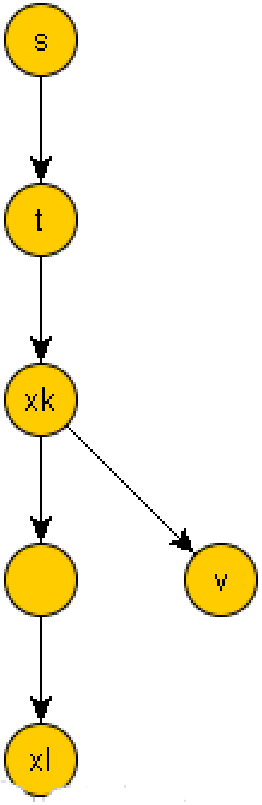
и пусть v с отцом x_k будет следующей вершиной, которая будет добавлена к L .

Предположим в качестве индукционной гипотезы, что для всех $0 \le i < j < l$, *знак* (x_i) = «+» ⇔

x_i следует за x_j в L ,

т.е. $x_i = x_j .p$

Так как знак (x_k) устанавливается на «-» , если v вставлено после x_k в L и в «+», если v вставлено перед x_k в L, то индуктивное предположение остается в силе после добавления v .



Следовательно, индукция верна.

$[x_j , x_i^{\,+}]$

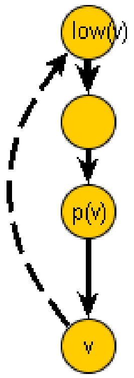
Корректность st- нумерации

(2) Пусть $v \neq s, t$.

Если (*v*, *low* (*v*)) — обратное ребро, то вставка *v* между

v.p и *v.Low* в *L* гарантирует, что в нумерации, соответствующей *L*,

v смежна как с вершиной с меньшим номером, так и с вершиной с большим номером.



Если знак (*v.Low*) = “ –” =>

([*v.Low* , …, *v* , *v.p*]

Если знак (*v.Low*) = “ +” =>

[*v.p* , *v* ,…, *v.Low*]

Корректность st- нумерации

В противном случае должна быть вершина *w* такая, что *w*. *p = v* и *w*. *Low* = *v*.*Low* .

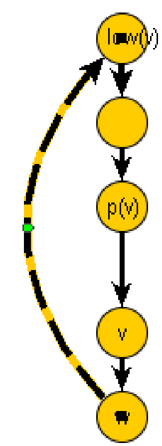
По лемме 2 мы имеем, что *v.Low является* собственным предком *v*, что означает, что знак

(*v*.*Low*) остается постоянным в течение времени, когда *v* и *w* добавляются к *L.*

Из этого следует, что *v* появляется между *v.p* и *w* в завершенном списке *L* ,

А это значит, что в нумерации, соответствующей *L*, *v* смежна как с вершиной с меньшим, так и с вершиной с большим номером.

Таким образом, имеет место второй случай.



[$v.Low^-$,..., *w* , *v* , *v.p*]

[*v.p*, *v*, *w* ,..., $v.Low^+$]

- Ваши вопросы?
- Контакты лектора:
  apanovich_09@mail.ru