Algoritmo 2.4: Recorrido en profundidad

función *recorrido_en_profundidad*(G, r):
- sea P una pila que sólo contiene a r
- **repetir mientras** $|P| > 0$ **:**
  - sea x el vértice que está en la cima de P
  - **si** x no ha sido visitado **:**
    - *previsitar*(x)
    - marcar x como visitado
  - **si** existe una arista x—y con y no visitado **:**
    - *procesar_arista*(x, y)
    - apilar y en P
  - **en otro caso:**
    - *posvisitar*(x)
    - desapilar x de P

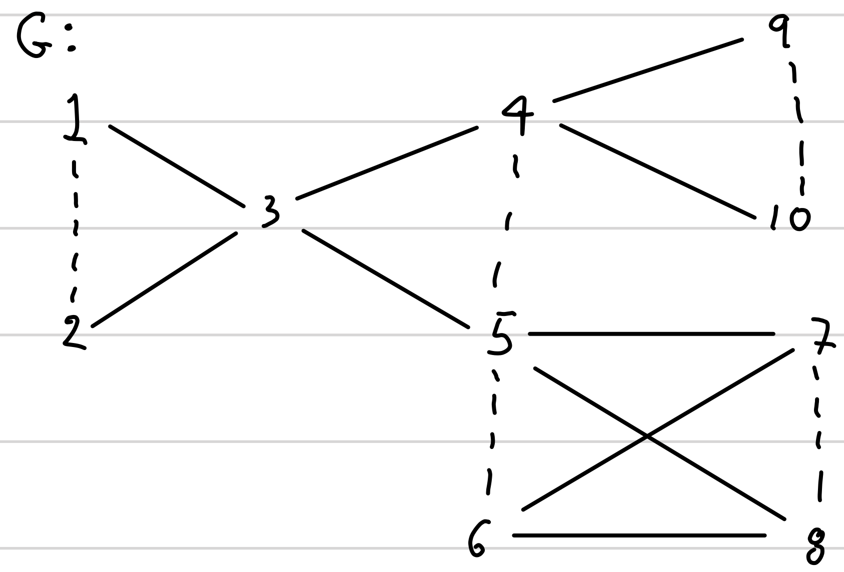
Algoritmo 2.7: Etiquetado en pre y postorden de vértices

función *previsitar*(u):
- $pre[u] := t$
- $t := t + 1$

función *posvisitar*(u):
- $pos[u] := t$
- $t := t + 1$

función *etiquetado_en_profundidad*(G):
- $t := 0$ ▶Tiempo inicial
- *recorrido_en_profundidad*(G, r)
- **devolver** $pre[\cdot], pos[\cdot]$

G:



Apliquemos el algoritmo 2.7 al bigrafo. Sea $r=1$, para $t=0$.

$P = [1]$ Pila con $r=1$.

$pre(1) = 0$, $t=1$, (1, visitado)

1 - - - - 2

$P = [2, 1]$, $pre(2) = 1$, $t=2$ (2, visitado)

2 —— 3 , $P = [3, 2, 1]$, $pre(3) = 2$, $t = 3$ (3, visitado)

3 —— 4 , $P = [4, 3, 2, 1]$, $pre(4) = 3$, $t = 4$ (4, visitado)

4 - - - 5, $P = [5, 4, 3, 2, 1]$, $pre(5) = 4$, $t = 5$ (5, visitado)

5 - - - 6, $P = [6, 5, 4, 3, 2, 1]$, $pre(6) = 5$, $t = 6$ (6, visitado).

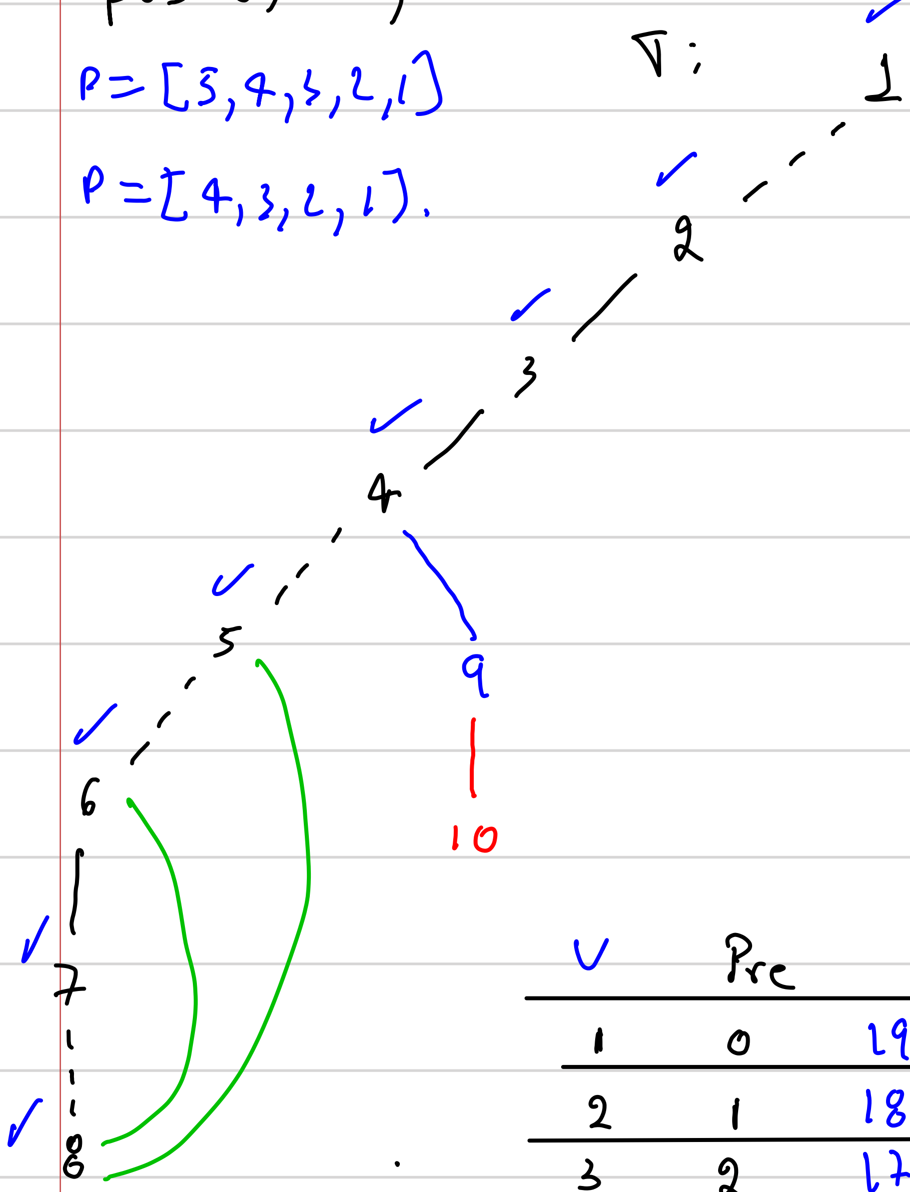
6 —— 7, P = [7, 6, 5, 4, 3, 2, 1], pre(7) = 6, (7, visitado) t = 7

7 - - - - 8, P = [8, 7, 6, 5, 4, 3, 2, 1], pre(8) = 7, t = 8, (8, visitado).

pos(8) = 8, t = 9.   P = [7, 6, 5, 4, 3, 2, 1]. ⟶ P = [6, 5, 4, 3, 2, 1]

P = [5, 4, 3, 2, 1]

P = [4, 3, 2, 1].



pos(7) = 9, t = 10

pos(6) = 10, t = 11

pos(5) = 11, t = 12

pre(9) = 12, t = 13   P = [9, 4, 3, 2, 1]

pre(10) = 13, t = 14, P = [10, 9, 4, 3, 2, 1].

pos(10) = 14, [9, 4, 3, 2, 1], t = 15

pos(9) = 15, [4, 3, 2, 1], t = 16

pos(4) = 16, [3, 2, 1], t = 17

pos(3) = 17, [2, 1], t = 18

pos(2) = 18, [1], t = 19

pos(1) = 19, P = ∅.

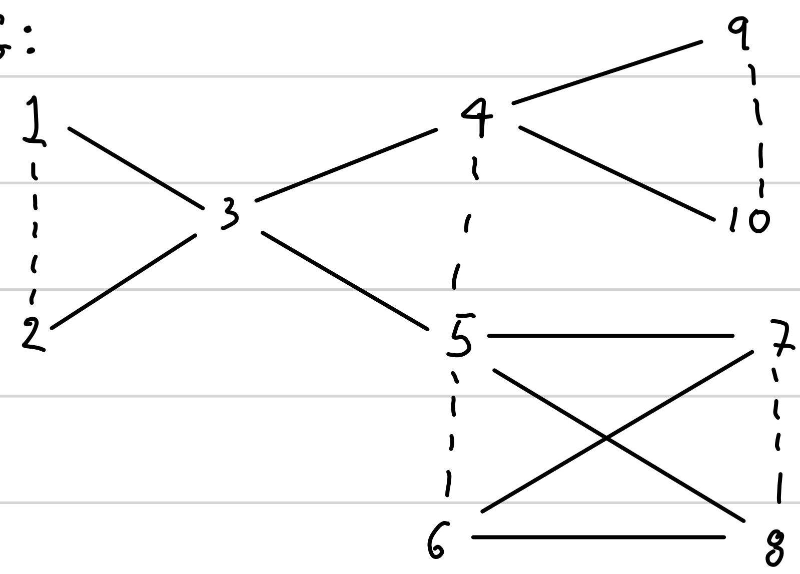
pre(9)

V	Pre	Pos
1	0	19
2	1	18
3	2	17
4	3	16
5	4	11
6	5	10
7	6	9
8	7	8
9	12	15
10	13	14

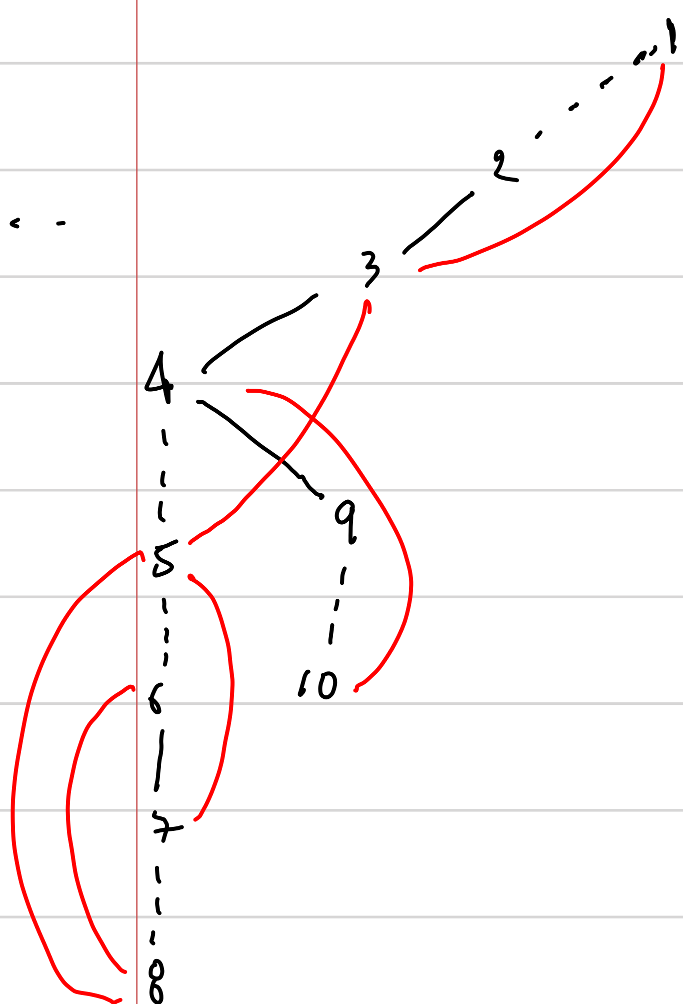
Para implementar sup[x] := mín $\begin{cases} \text{pre}(x) \\ \text{pre}(w) \text{ para cada arista de retroceso } x \to w \\ \text{sup}(y) \text{ para cada arista de árbol } x \to y. \end{cases}$

ya tienen que estar etiquetados los vértices con pre y pos

G:



v	Pre	Pos	sup
1	0	19	0
2	1	18	0
3	2	17	0
4	3	16	2
5	4	11	2
6	5	10	4
7	6	9	4
8	7	8	4
9	12	15	3
10	13	14	3



$$sup(x) = \min \begin{cases} pre(x) \\ pre(w) \text{ para cada arista de retroceso } x \to w \\ sup(y) \text{ para cada arista de árbol } x \to y \end{cases}$$