

3. Úkol

pátek 20. listopadu 2020

19:20

Úloha 1. Navrhněte algoritmus, který převede zadaný binární vyhledávací strom T na seřazený lineární seznam za použití pouze konstantního množství pomocné paměti, tj. bez použití pomocného pole, pouze za pomoci přepojování ukazatelů.

Algoritmus:

strukturní struktura

$T \rightarrow m_Parent$
$T \leftarrow m_Left$
$T \leftarrow m_Right$
? m_Value

```
(1) | MAKE LIST (  $T^* bvs$  )  
(2) |  $T^* start = FindMin(bvs);$   
(3) | for ( $T^* x = start; x;$ )  
|   |  $T^* next = FindNext(bvs);$   
|   |  $x \rightarrow m\_Right = next;$   
|   |  $x = next;$   
(7) | RETURN start;
```

Kde FindMin je operace nalezení minima BVS tak, jak byla uvedena na přednášce (vrátí ukazatel na min.) a FindNext je operace nalezení následníka v BVS tak, jak byla ur. na předn. (vrátí ukazatel /NULL, pokud nemá následníka)

Důkaz horečnosti a horečnosti:

Algoritmus bude korektní, pokud navštíví každý vrchol právě jednou a to v pořadí od nejmenšího po největší. Pak budou opravdu odkazy m-Right zpřístupňovat spojující seznam.

Toto správné pořadí navštívení vrcholů je podmíněno správností operace FindNext i po výměnách obněm přepojení ukazatelů.

Důkaz indukci: (Očíslovme vrcholy 1...n podle pořadí v seřazené posl.)

① Při prvním volání FOR cyklu bude strom BVS, FindNext tedy bude korektní. Přeponíme ukazatel na vrcholu 1.
(rozbití) m-right

② Při dalším volání pro n+1. vrchol bude operace FindNext(v) postupovat následovně (z def.):

A₁ IF ($v \rightarrow m\text{-Right}$) RETURN FindMin($v \rightarrow m\text{-Right}$);

B₁ Vrat̄ prvního rodiče, do kterého přijde zleva.

Krok B₁ využívá pouze odkazy M-LEFT a M-PARENT, které zůstaly nezměněny. ✓

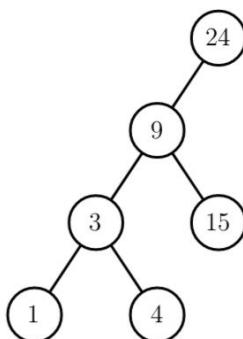
Krok A₁ využívá m-Right na n+1. vrcholu, přičemž tento ukazatel je rozbitý na nejvýše n vrcholech.

Dále v operaci FindMin využívá ukazatele M-RIGHT u potomků, a těch ale víme, že mají označení $> n+1$ (v pravém poněkudu jsou větší prvky) a tedy také fungují. ✓

Algoritmus bude konečný, jelikož bude každý vrchol navštíven právě jednou a máme konečný počet vrcholů.

Takéž bude horečkový, jelikož se vrcholy navštíví ve správném pořadí a následný seznam začne původním minimem a skončí původním maximem. □

Úloha 2. Uvažujte následující binární vyhledávací strom. Napište všechny možné posloupnosti představující pořadí, ve kterém mohly být klíče vkládány tak, aby vznikl takovýto BVS.



Vzhledem k tomu, jak funguje operace BVSInsert, vložený klíč může být vždy pouze list. Jako poslední tedy mohly být vloženy:

$$1 \overbrace{-15}^4 - \overbrace{3}^1 - \overbrace{15}^3 - \overbrace{3}^1 - \overbrace{9}^3 - \overbrace{24}^1$$

NEBO

$$4 \overbrace{-15}^1 - \overbrace{3}^1 - \overbrace{15}^3 - \overbrace{3}^1 - \overbrace{9}^3 - \overbrace{24}^1$$

NEBO

$$15 \overbrace{-4}^1 - \overbrace{3}^1 - \overbrace{4}^1 - \overbrace{3}^1 - \overbrace{9}^3 - \overbrace{24}^1$$

Mozné posloupnosti jsou tedy:

$$\boxed{24-9-3-4-1-15}$$

$$24-9-3-1-4-15$$

$$24-9-3-4-15-1$$

$$24-9-3-1-15-4$$

$$24-9-3-15-1-4$$

$$24-9-3-15-4-1$$

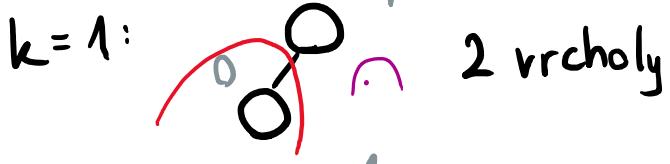
$$24-9-15-3-1-4$$

$$24-9-15-3-4-1$$

A je jich 8.

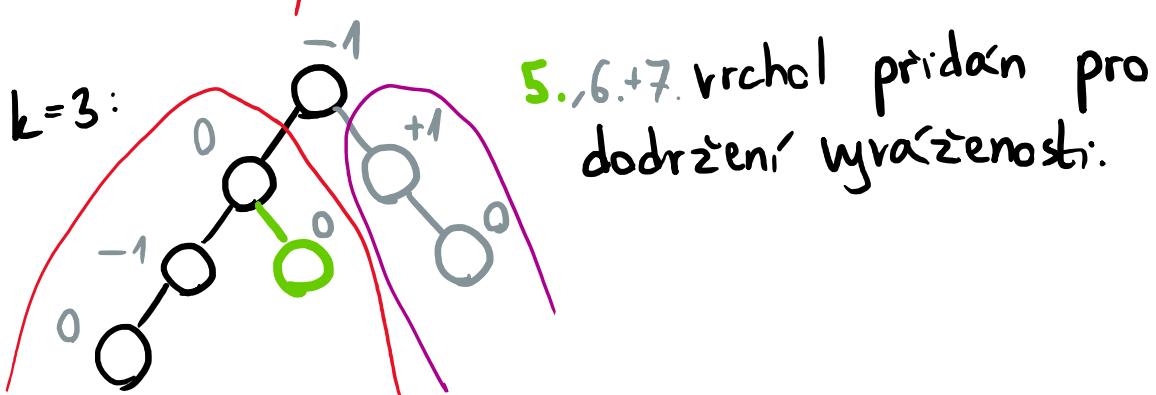
Úloha 3. Dokažte, že platí vztah $A_k = F_{k+3} - 1$, kde A_k je minimální velikost AVL stromu hloubky k a F_n je n -té Fibonacciho číslo.

Pojďme na to induktivně:



$k=2$:

Ctvrтí vrchol (šedě) jsme museli dodat, aby byla dodržena vyraženost AVL stromu.



Všimněme si, že při konstrukci vždy nakonec dojdeme k tomu, že se min. AVL strom hloubky k skládá z min. AVL stromů hloubky $k-1, k-2^+$ (KORENE) ^{1 vrcholu}

Proč? Při konstrukci min. AVL stromu se vždy chceme "priklonit na jednu stranu" aby bylo $\forall v \in V: |\ell(L(v)) - \ell(P(v))| = 1$ Toho dosáhneme

právě (díky znalosti toho, že každý podstrom musí být také AVL) níže popsanou konstrukcí.

Pro počet vrcholů min. AVL stromu hloubky k
tedy platí: VELIKOST

$$A_k := \left\{ \begin{array}{ll} 1 & \dots \quad k=0 \\ 2 & \dots \quad k=1 \\ 1 + A_{k-1} + A_{k-2} & \dots \quad \forall k \geq 2 \end{array} \right\}$$

$$A_{k-1} = \left\{ \begin{array}{ll} 0 & \dots \quad k=0 \\ 1 & \dots \quad k=1 \\ A_{k-1} + A_{k-2} & \dots \quad \forall k \geq 2 \end{array} \right\}$$

$$A_{k-1} = F_k \Leftrightarrow A_{k-1} = F_{k+1} - 0$$

$$\underline{A_k = F_{k+3} - 1} \Leftrightarrow A_{k-1} = F_{k+2} - 1$$

(Ze znalosti prvních 3 členů Fib. posloupnosti
tedy dostavíme rovnost)

□

Úloha 4. Skupina n trampů byla policisty chycena, jak právě prošla železničním tunelem, což je v naší zemi přísně zakázáno. Proto byla skupina okamžitě převezena na výslech, při kterém se policisté snaží zjistit, kdo z nich je autorem tohoto nápadu.

Trampové mají strážce zákona v hluboké úctě, tedy si nedovolí jím lhát. Podobně silné pouto je ale váže i k jejich zeleným bratřím. Z toho důvodu není žádný z uličníků ochotný sdělit pořadí, v jakém do tunelu vlezli, výpověď každého z nich tak sestává pouze z potenciálně prázdného seznamu několika málo přezdívek kolegů, kteří do tunelu vešli před ním.

Navrhněte algoritmus, který policii pomůže najít toho výtečníka, který do tunelu vlezl jako první.

Algoritmus:

1. Vytvoríme si booleovské pole o délce n NAPLNĚNO
HODNOTAMI TRUE
2. Každý, kdo mluvil, má hodnotu FALSE
3. První vešli lidé, kteří mají v poli hodnotu TRUE

Důraz horečnosti: máme pouze cykly s konečným počtem opakování (max n -krát), program skončí. ✓

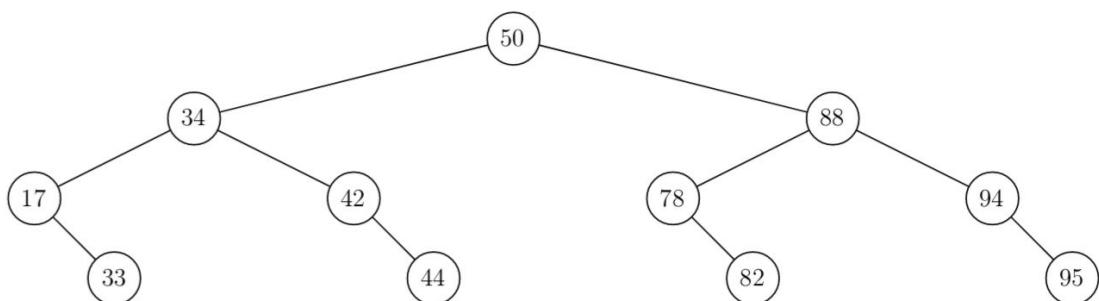
Důraz horečnosti:

Každý, kdo mluvil, řekl že před ním někdo vešel \Rightarrow mluvící osoba tedy nemohla být první.

O lidech, kteří nemluvili (pokud jich bylo více), nemůžeme s jistotou říct, kdo z nich byl první.

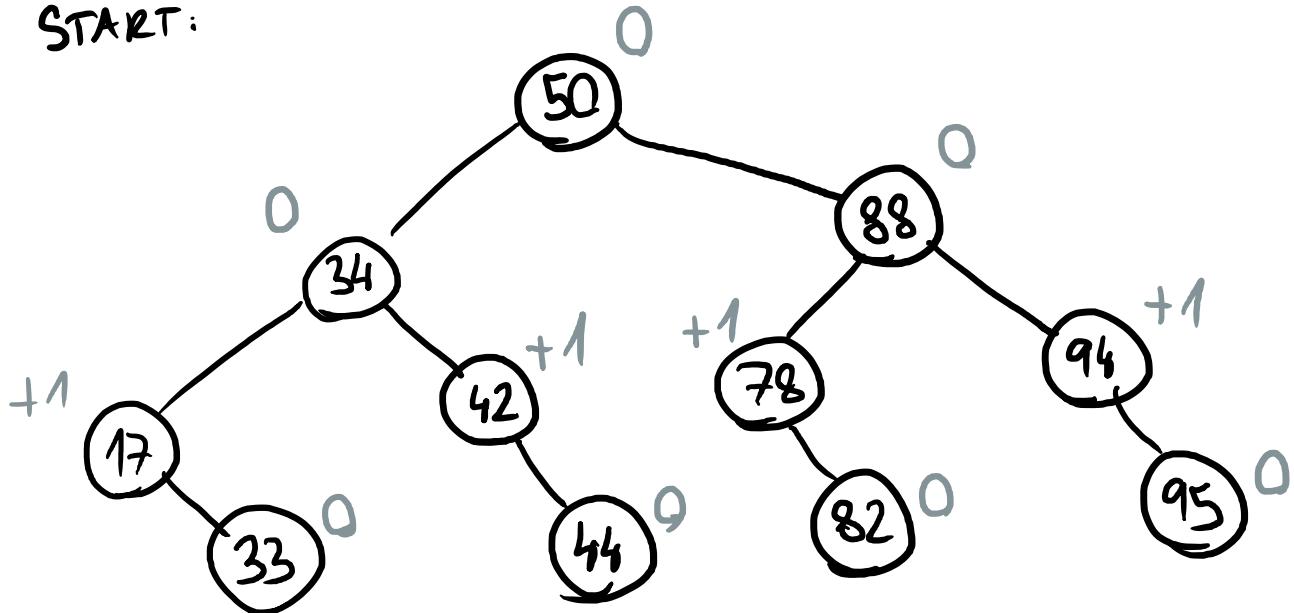
Určitě nenastane situace, kdy by všichni mluvili, protože pak by nastal "cyklus" a někdo by lhá, což je v rozporu se zadáním. \square

Úloha 5. Uvažujte binární vyhledávací strom T z Obrázku 2. Dokažte, že se jedná o AVL strom. Poté provedte následující operace: $AVLInsert(10)$, $AVLInsert(48)$, $AVLInsert(43)$, $AVLInsert(87)$, $AVLInsert(90)$, $AVLInsert(93)$, $AVLInsert(91)$, $AVLInsert(89)$, $AVLDelete(48)$, $AVLDelete(90)$, $AVLDelete(88)$. Zakreslete veškeré mezivýsledky (po každém vložení/smažení) a použité rotace pojmenujte. U každého vrcholu uvádějte jeho aktuální znaménko.



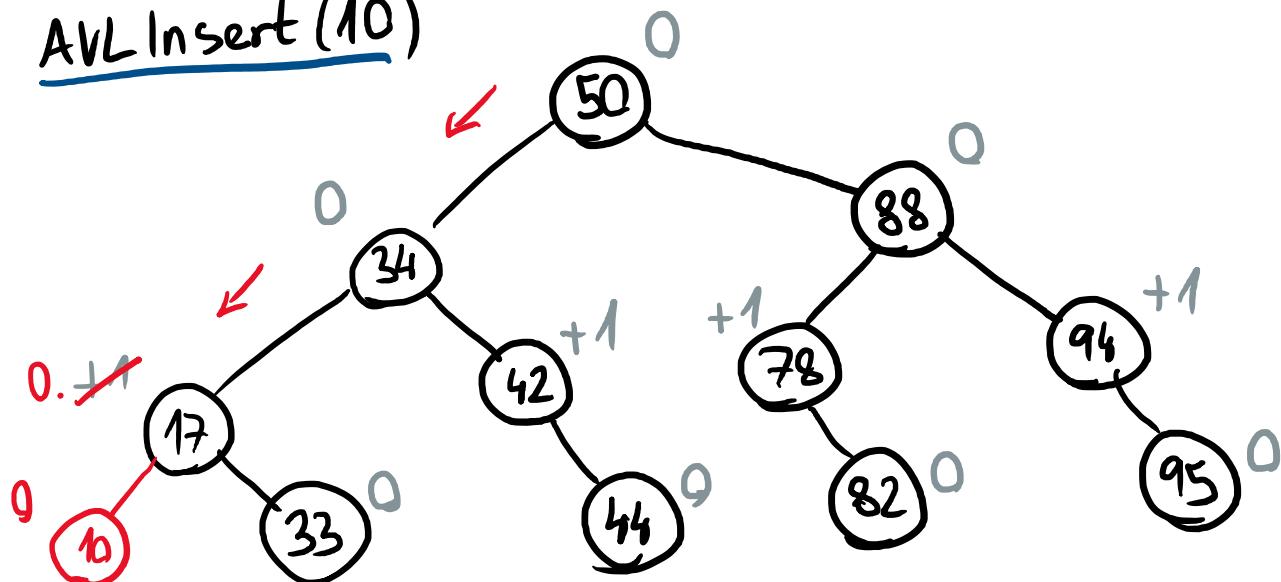
Obrázek 1: Binární vyhledávací strom ze zadání Úlohy 5.

START:

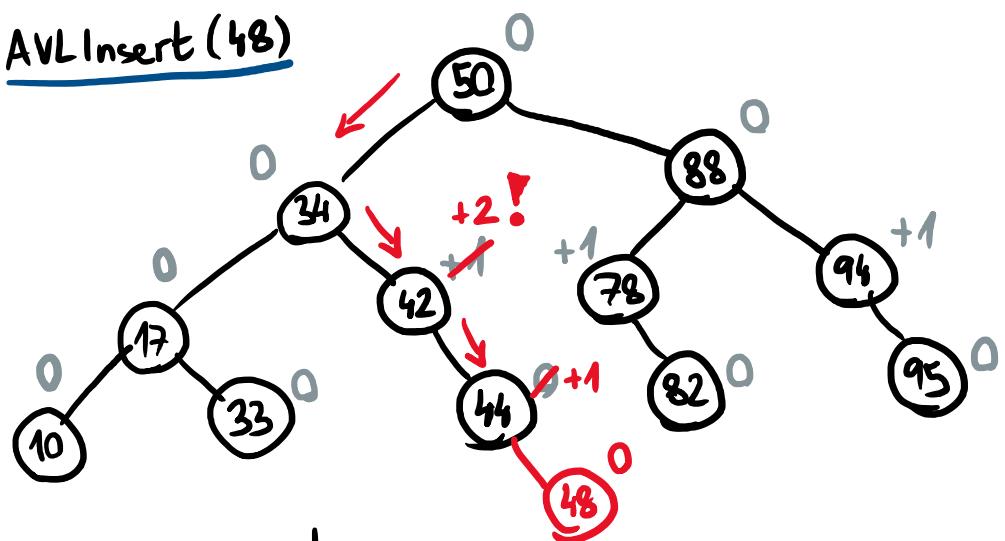


Sborem T je AVL, pokud $\forall v \in V$ $|h(L(v)) - h(P(v))| \leq 1$.
Z císel u vrcholeů vidíme, že je to splněno.

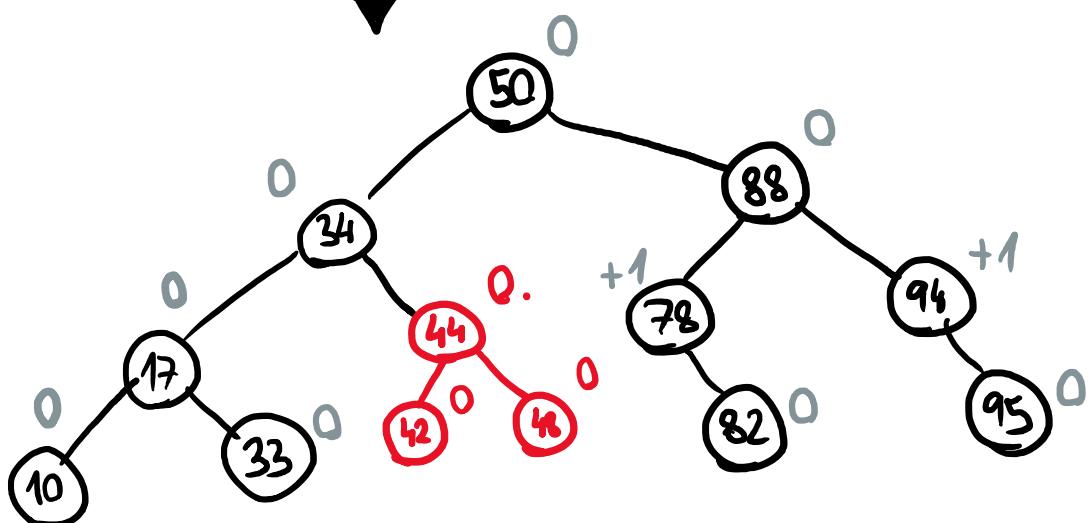
AVL Insert (10)



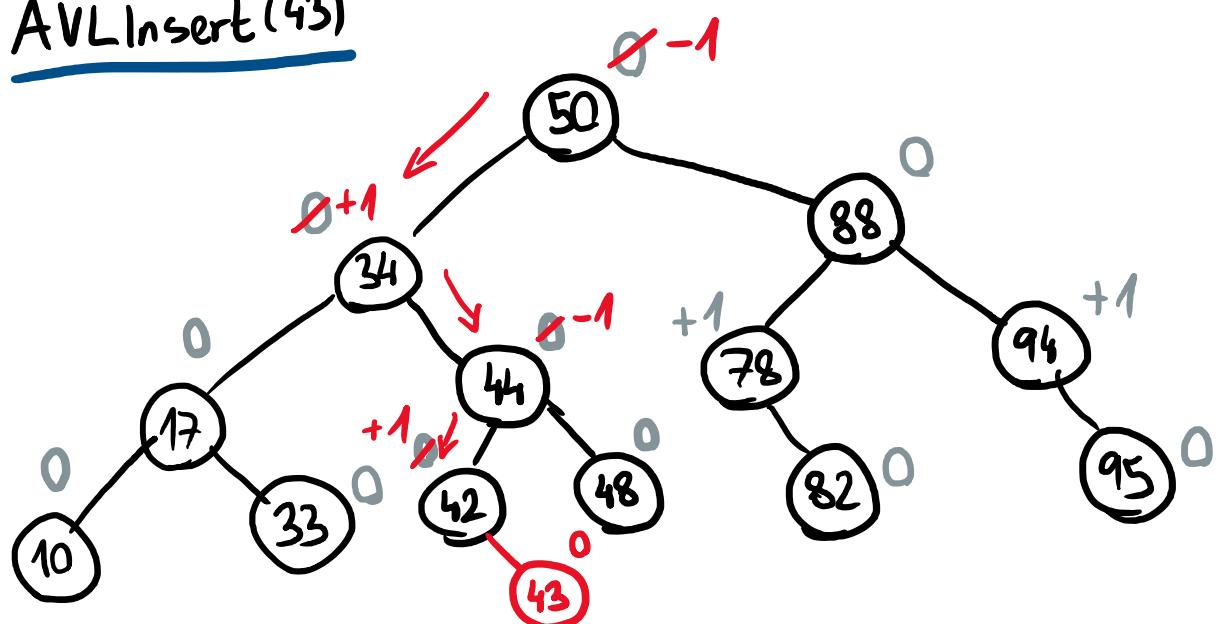
AVLInsert (48)



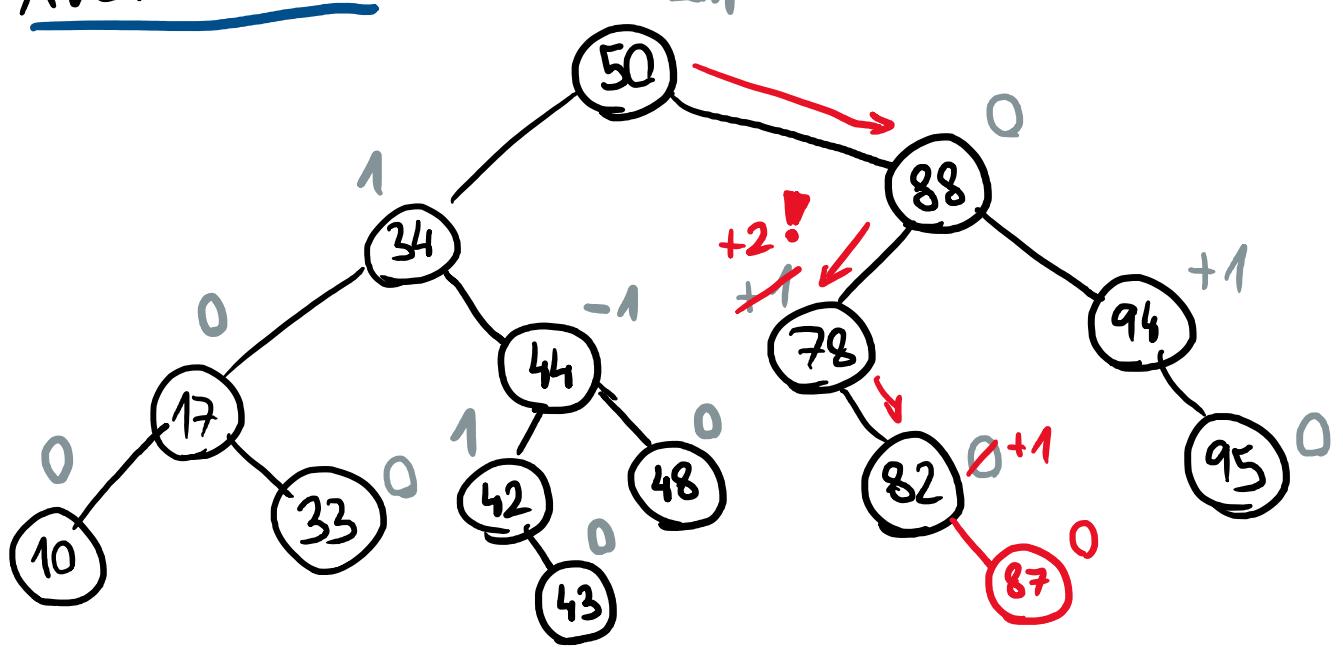
L, 42



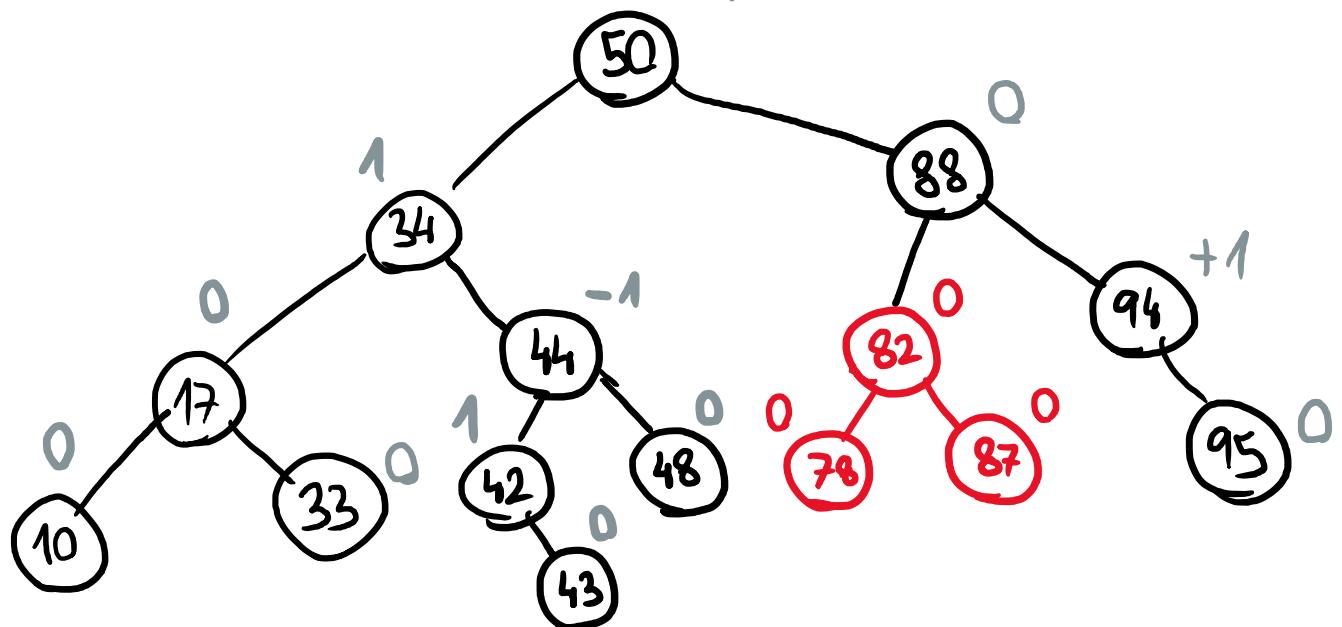
AVLInsert (43)



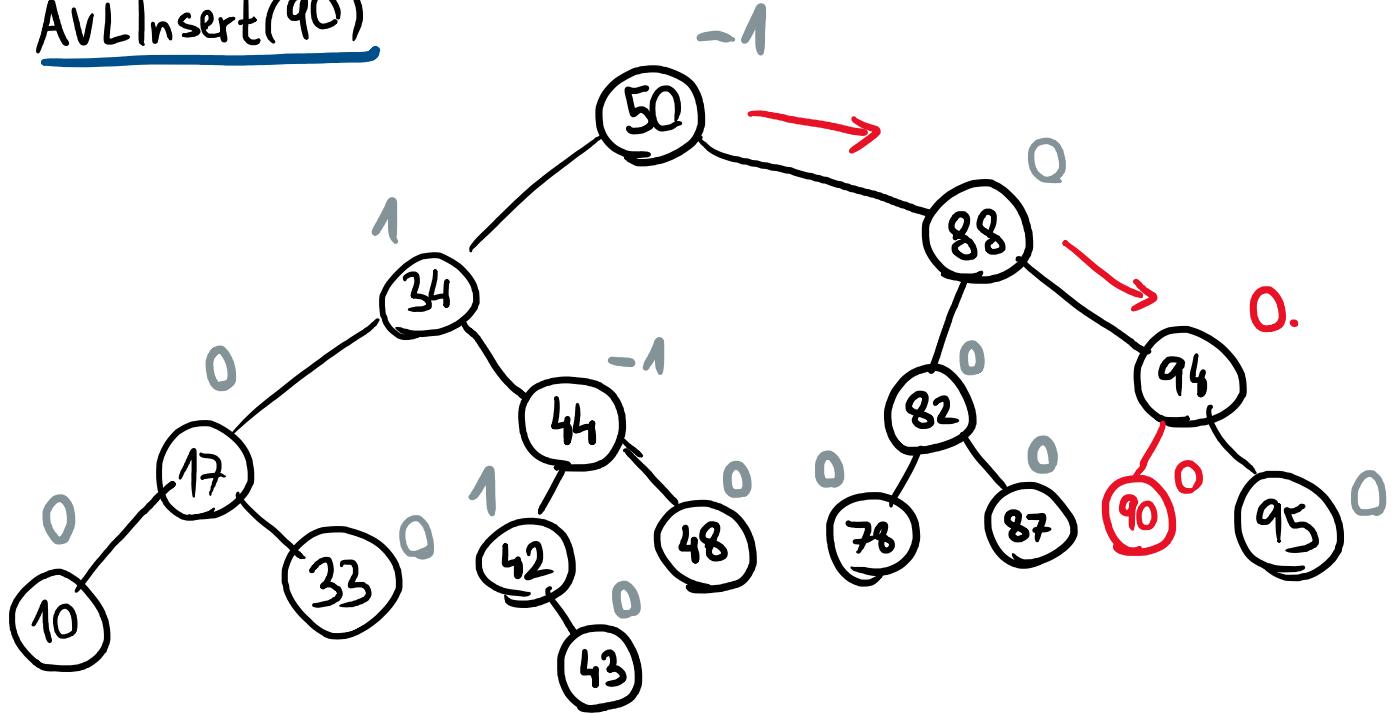
AVLInsert (87)



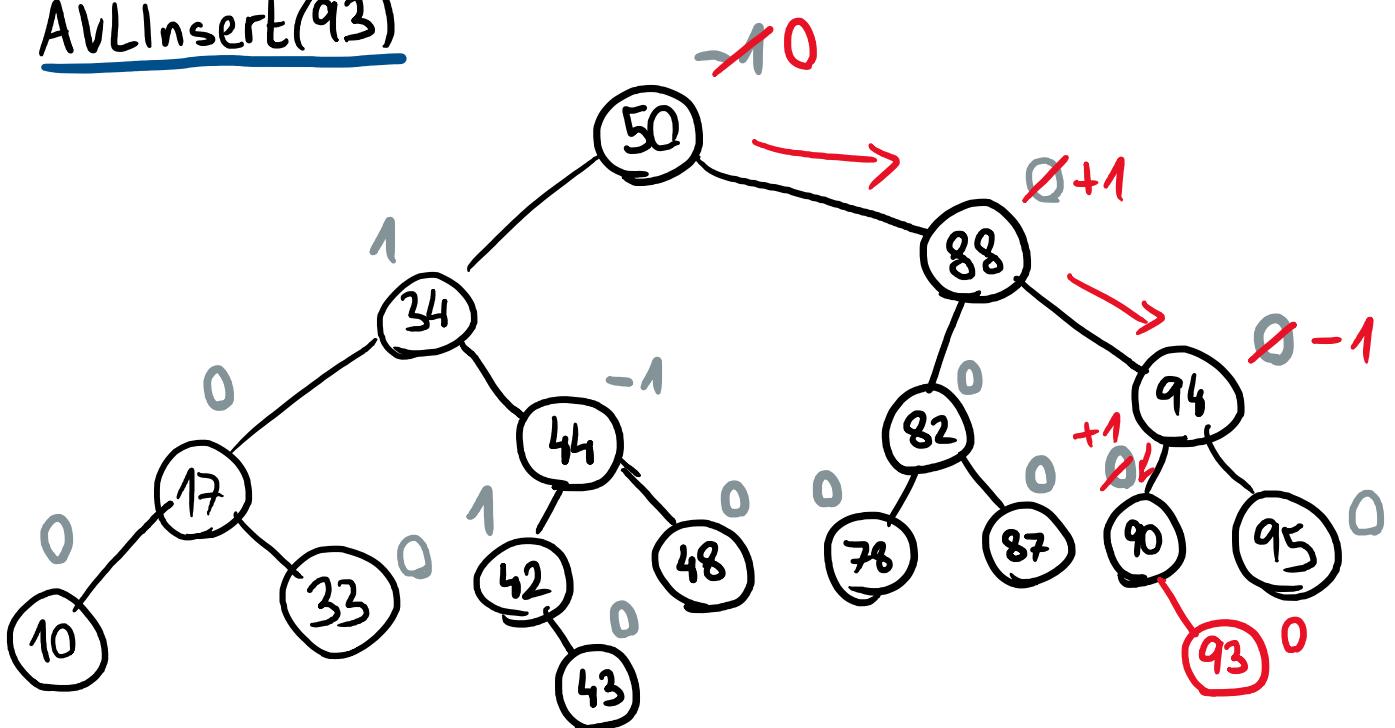
L, 78



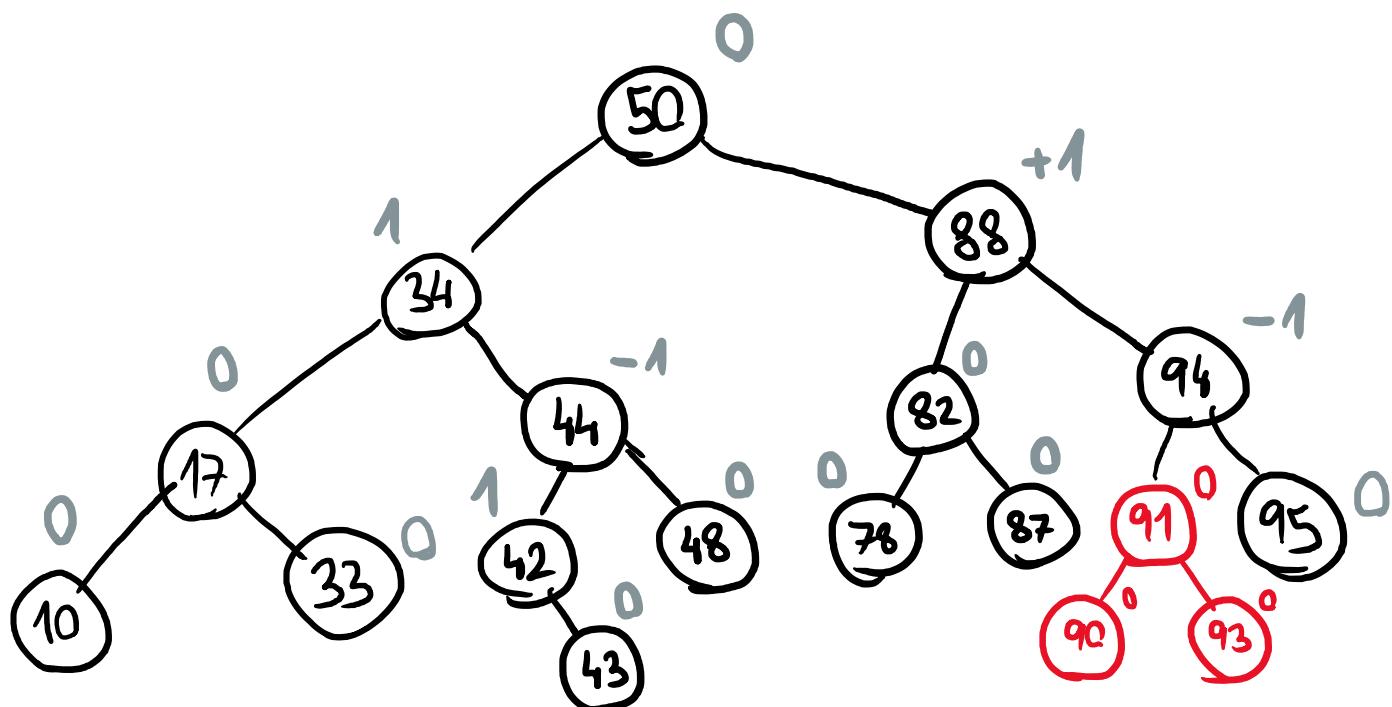
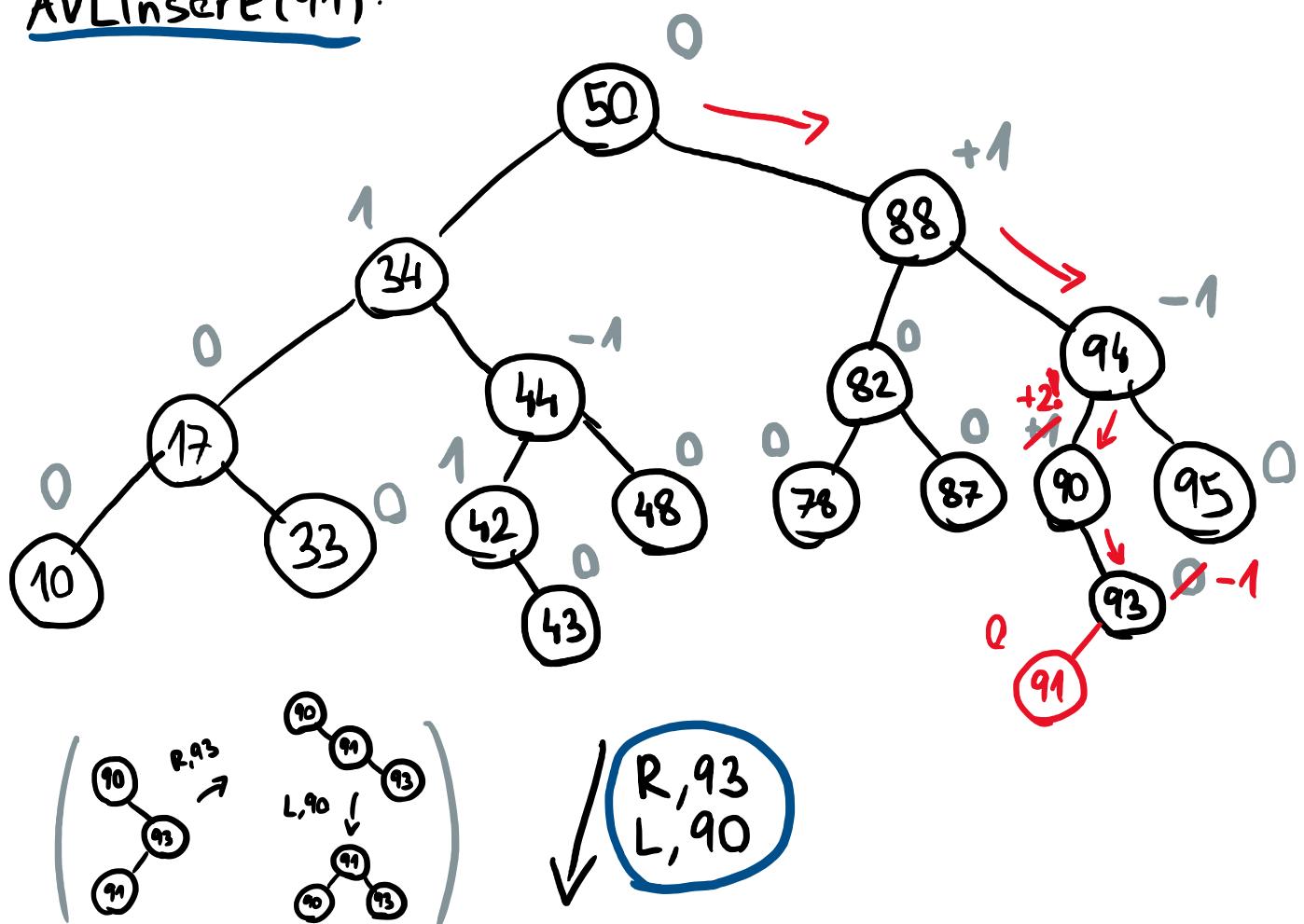
AVLInsert(90)



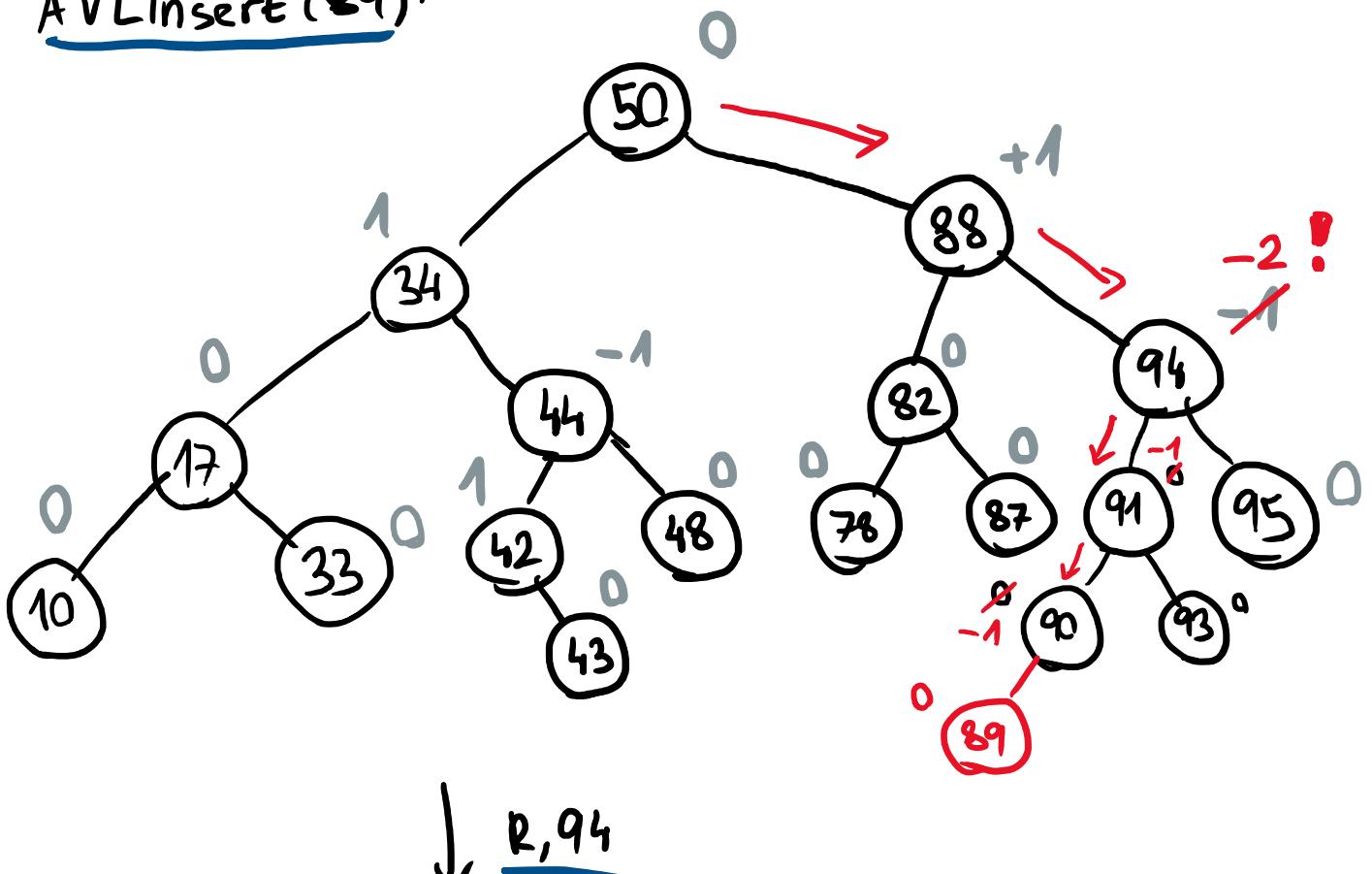
AVLInsert(93)



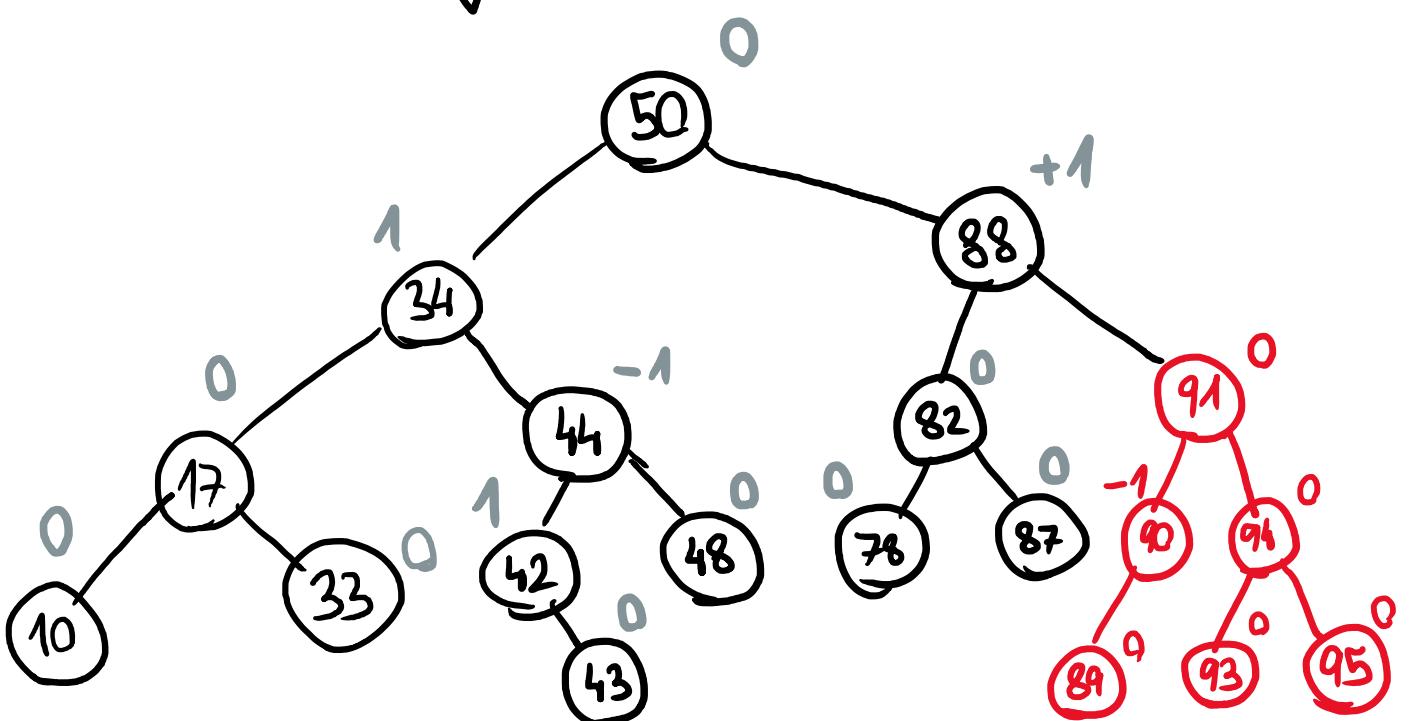
AVLInsert(91):



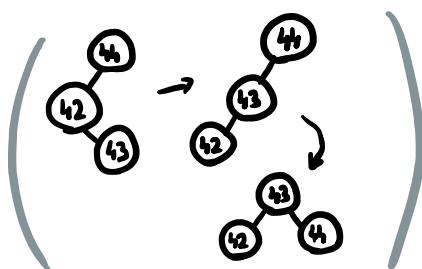
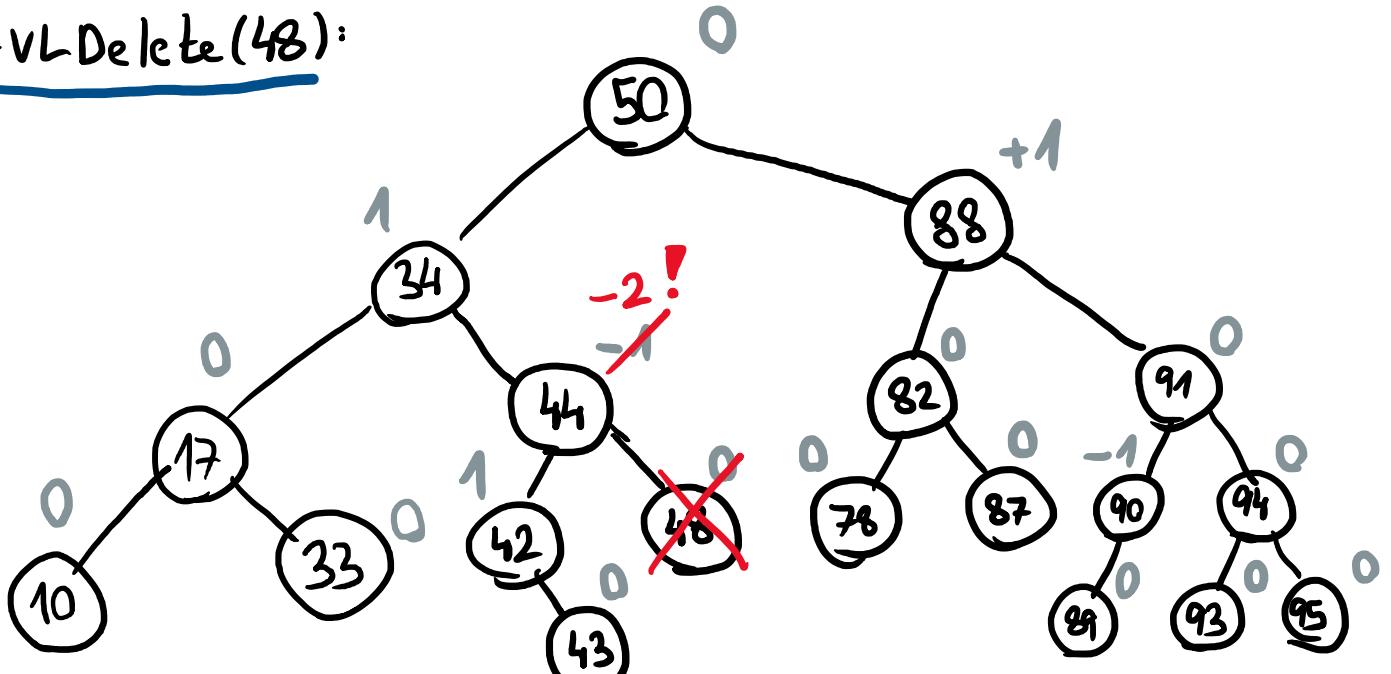
AVLInsert(89):



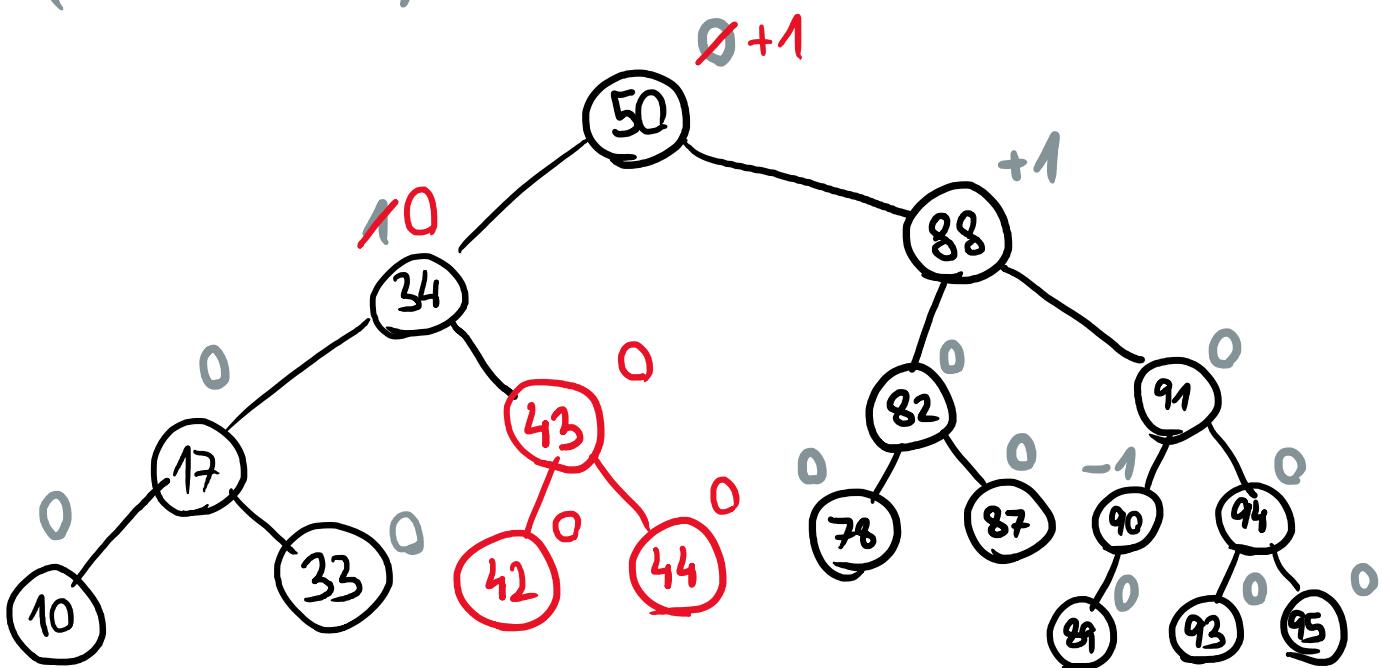
R, 94



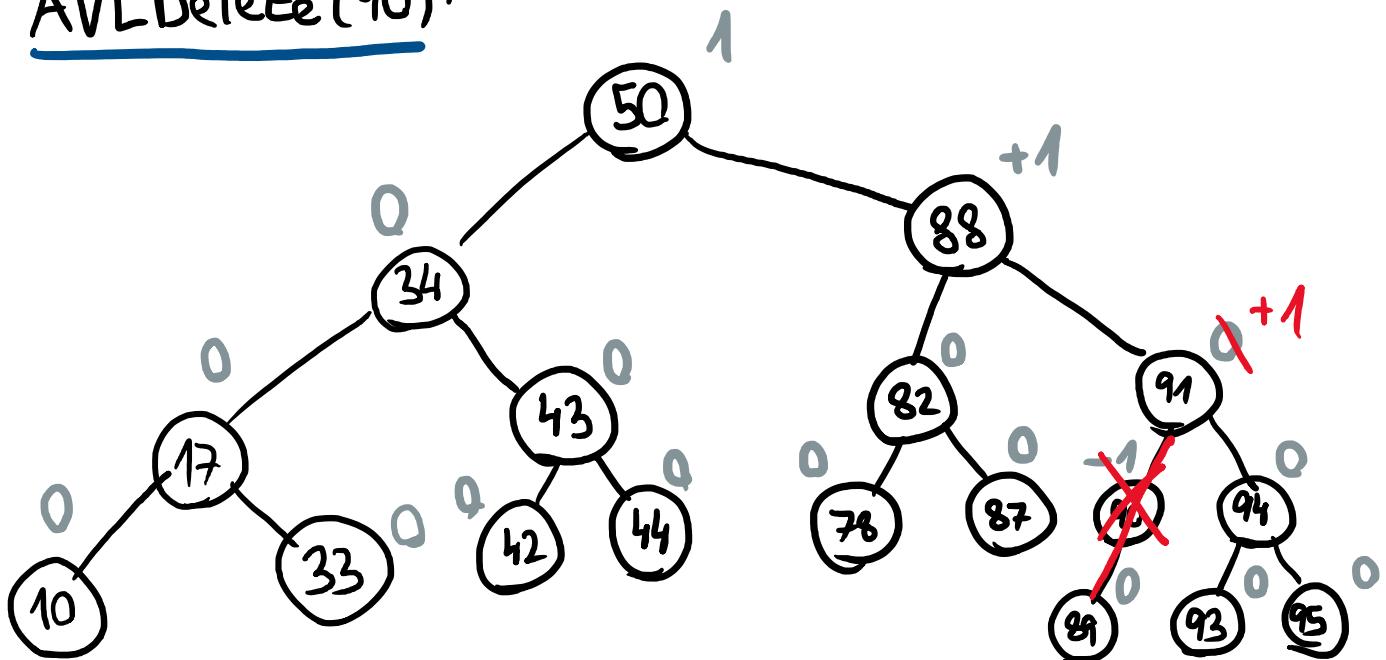
AVL Delete(48):



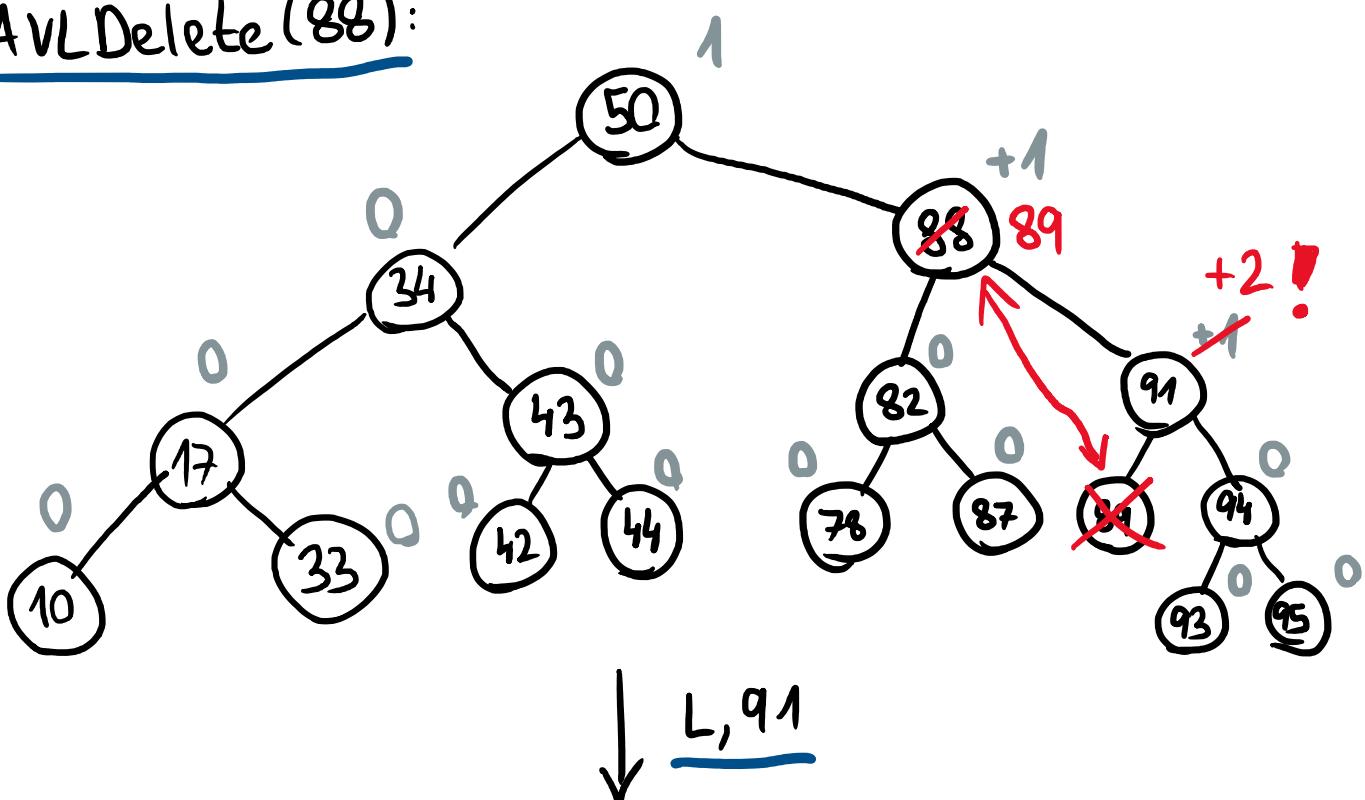
L, 42
R, 44

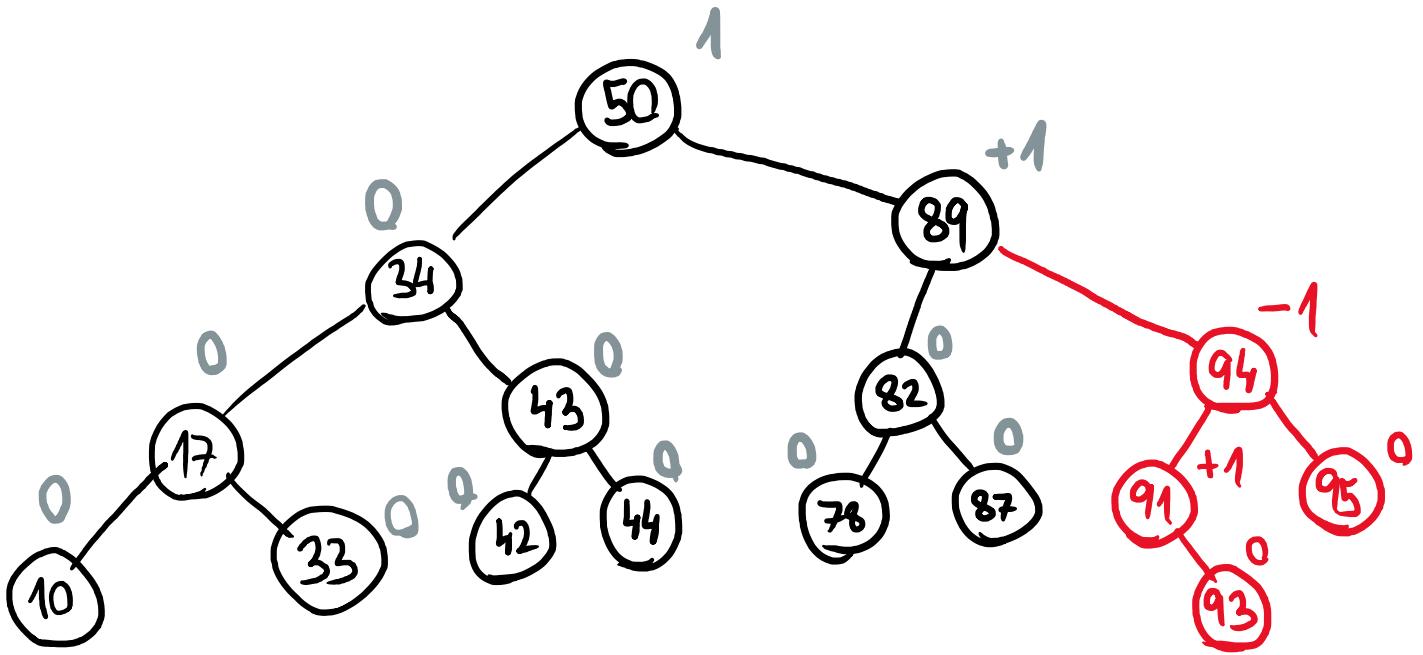


AVL Delete (90):



AVL Delete (88):





Výsledný AVL strom:

