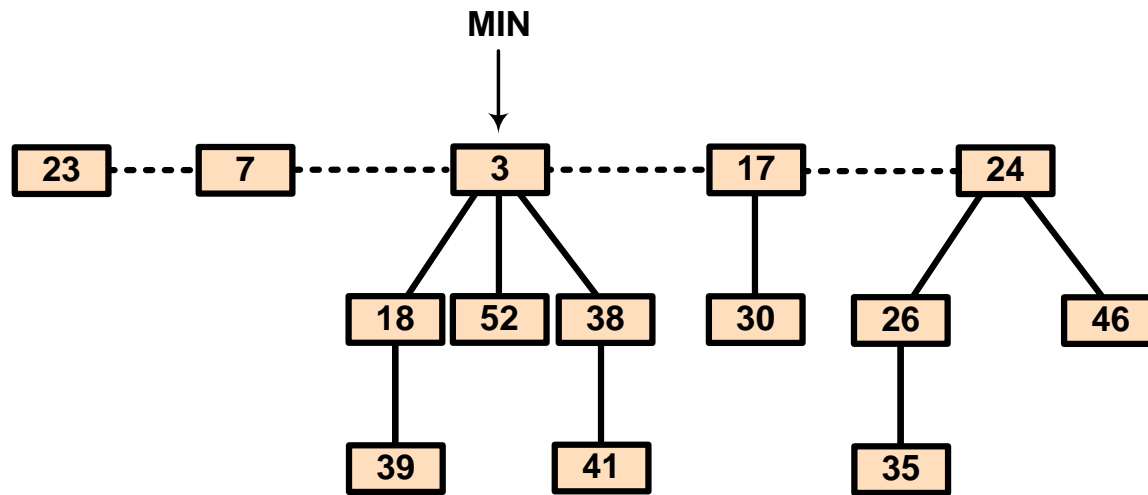


Fibonacciho halda

(Fibonacci heap)

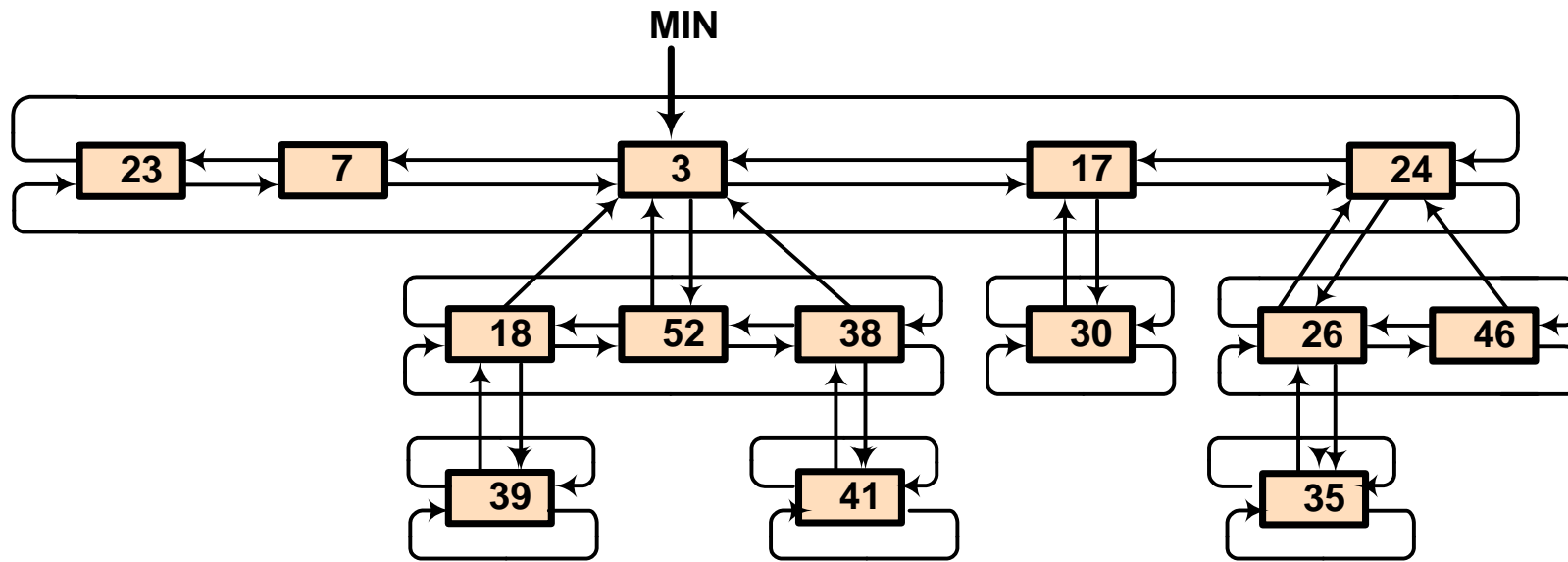
- je implementáciou prioritného frontu
- kľúč sa nazýva **priorita**
- halda je strom, v ktorom pre všetky vrcholy platí, že priorita ich kľúča je vyššia ako priorita kľúčov ich priamych potomkov
- v príkladoch bude vyššiu prioritu predstavovať nižšie číslo
- vo všeobecnosti prioritou môže byť akýkoľvek dátový typ, je potrebné definovať, čo znamená vyššia priorita
- stupeň uzla predstavuje počet jeho synov
- každý uzol má stupeň najviac $\log n$
- veľkosť podstromu vychádzajúceho z uzla stupňa k je najmenej F_{k+2} , kde F_k je k -té Fibonacciho číslo

Fibonacciho halda predstavuje les haldovo usporiadaných viaccestných stromov, v ktorom sa vždy udržiava explicitná referencia na prvok s maximálnou prioritou.



Odporúčaná pamäťová reprezentácia (formát vrcholov je homogénny napriek tomu, že strom je viaccestný):

- uzol má maximálne 4 referencie: ľavý sused, pravý sused, otec, jeden zo synov



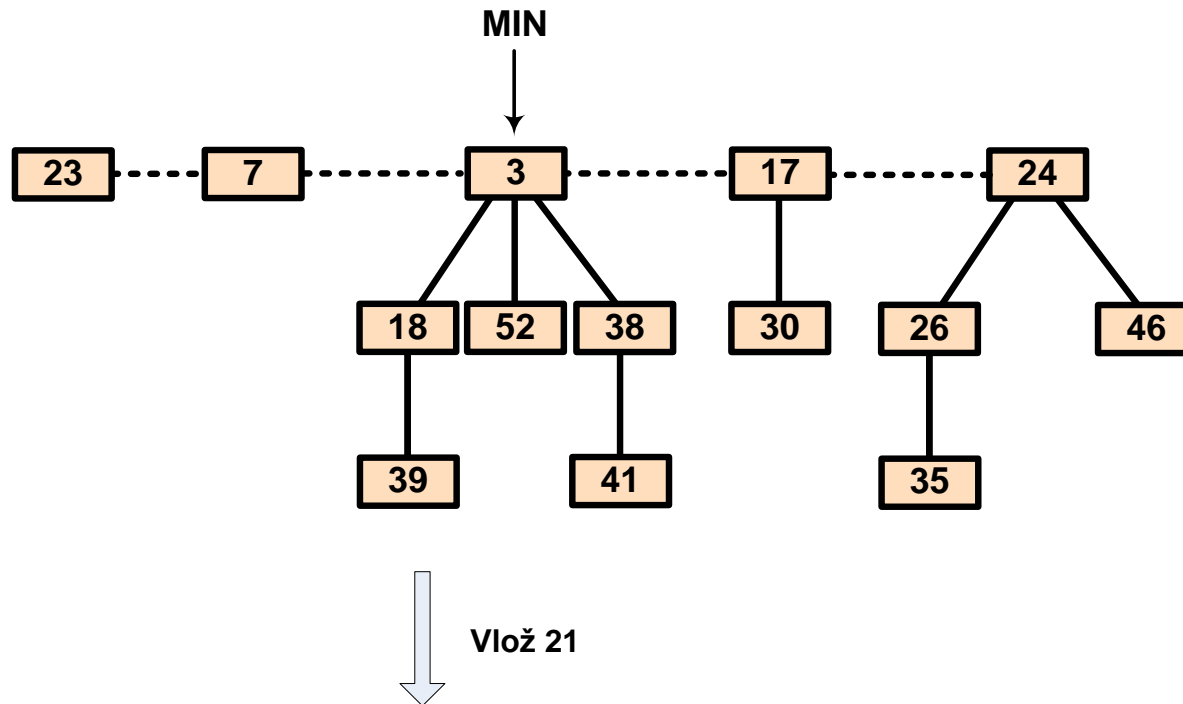
Princíp „značenia“

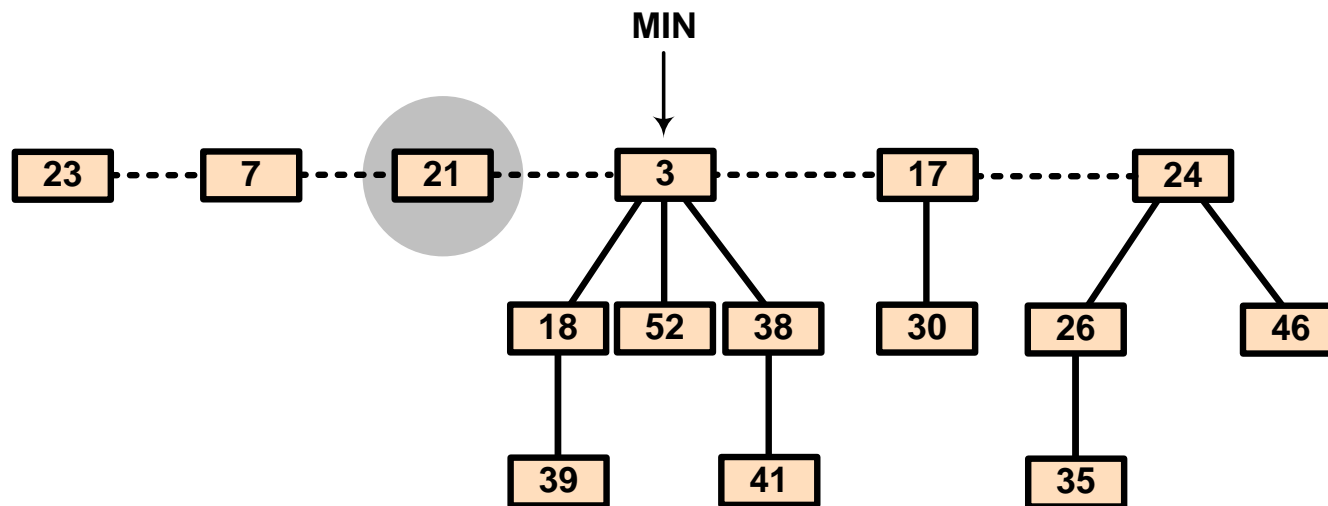
- v každom prvku X je rezervované pamäťové miesto pre uloženie „značky“ značka (X)
- značka (X) = true (vrchol X je označený) znamená, že X stratil syna od doby, keď bol naposledy ustanovený ako syn nejakého uzla
- novo vložený prvok je vždy koreňom, nastaví sa mu hodnota značka(X) = false (koreň je vždy neoznačený)
- prvok sa označí: keď nie je koreň a stratí syna
- prvok sa odznačí: keď stratí druhého syna
- značka sa využíva pri operáciách zmeny priority

Operácie

Operácia Vlož - priemerná aj najhoršia zložitosť $O(1)$:

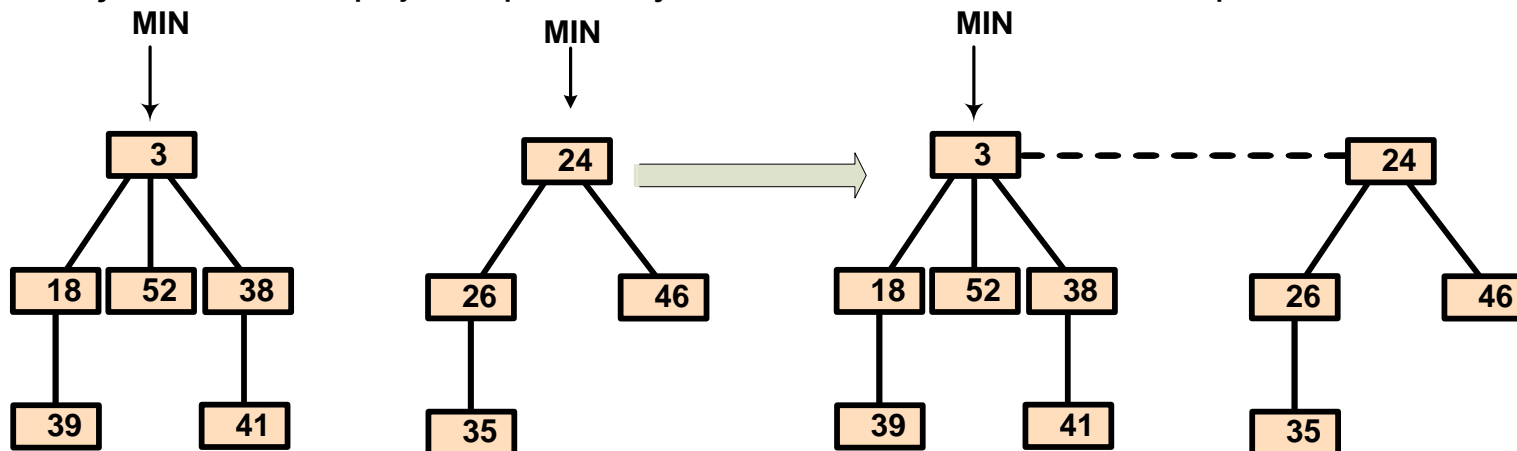
- vytvorí sa nový (jednoprvkový) strom, ktorý sa vloží do zoznamu koreňov (haldovo usporiadaných stromov) ako „ľavý sused“ prvku (koreňa) s maximálnou prioritou (v celej Fibonacciho halde)
- nastavenie značky prvku na *false* (odkazujú sa na ňu ďalšie operácie)
- prípadná aktualizácia referencie na prvok (koreň) s maximálnou prioritou



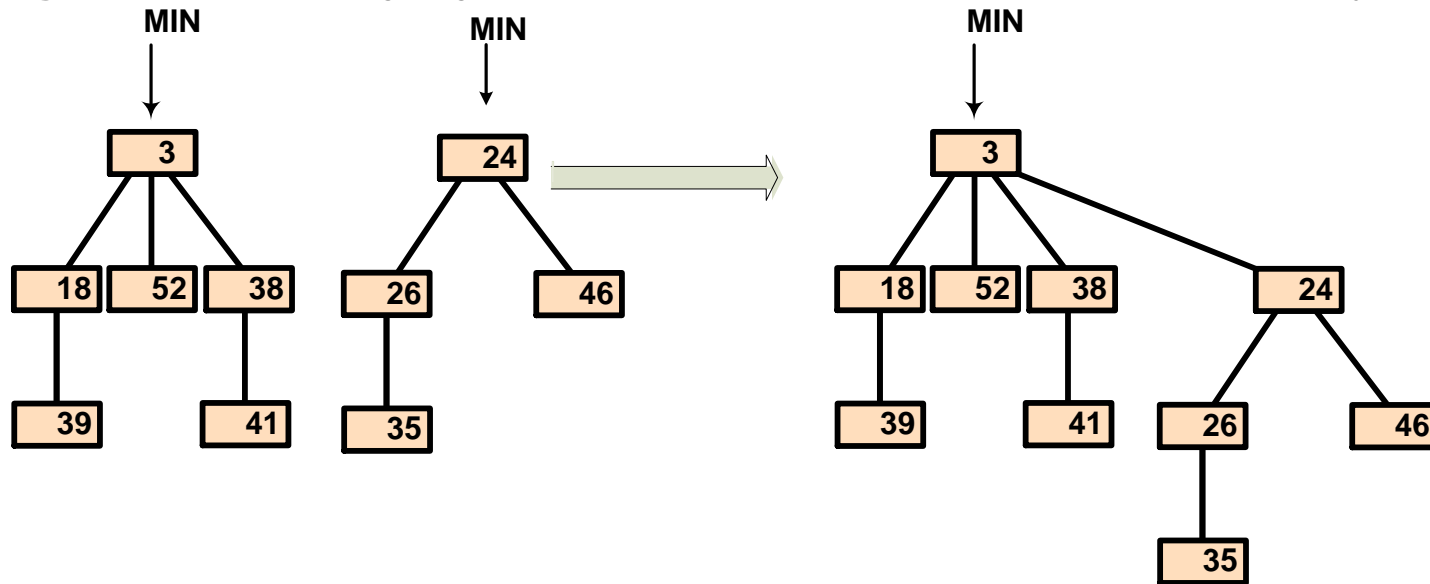


Zjednotenie dvoch háld - priemerná aj najhoršia zložitosť $O(1)$:

- jednoduché spojenie príslušných koreňov, nastavenie MIN pointera



Merge - priemerná aj najhoršia zložitosť $O(1)$: (väčší koreň sa stane synom menšieho)



Operácia Vymaž minimum - priemerná zložitosť $O(\log n)$, najhoršia zložitosť $O(n)$:

1. fáza algoritmu - odobratie

- odobratie prvku P_{MIN} (koreňa) s maximálnou prioritou zo zoznamu koreňov haldovo usporiadaných stromov
- jednoduché spojenie zoznamu synov prvku P_{MIN} so zoznamom koreňov (operácia zjednotenia dvoch háld)
- nastavenie referencie na prvok (koreň) s maximálnou prioritou

2. fáza algoritmu - konsolidácia

- les stromov Fibonacciho haldy upravíme tak, aby žiadne dva korene stromov v ňom neboli rovnakého stupňa:
 - zoznam koreňov stromov Fibonacciho haldy postupne prechádzame od koreňa s maximálnou prioritou

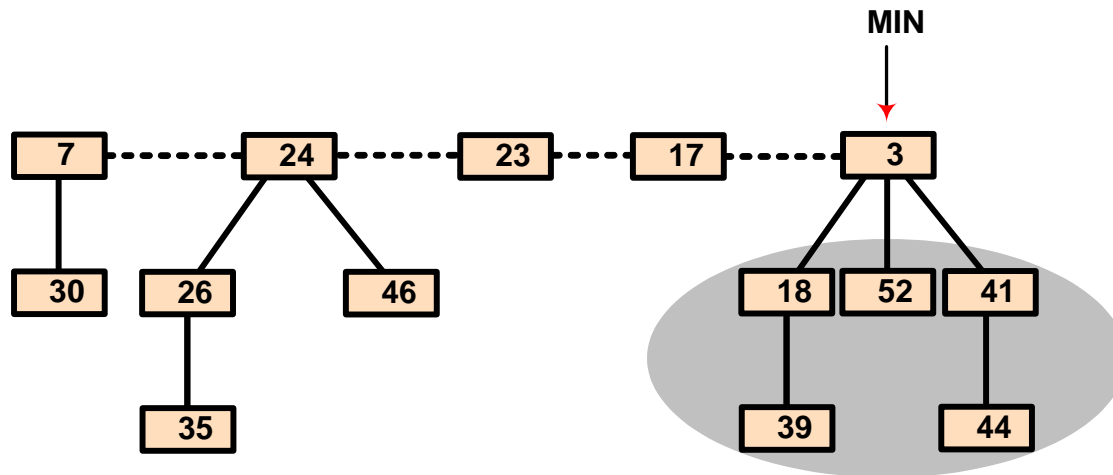
- ak identifikujeme dva stromy, ktorých korene sú rovnakého stupňa, spojíme ich operáciou **Merge** (tým zvýšime stupeň o 1)

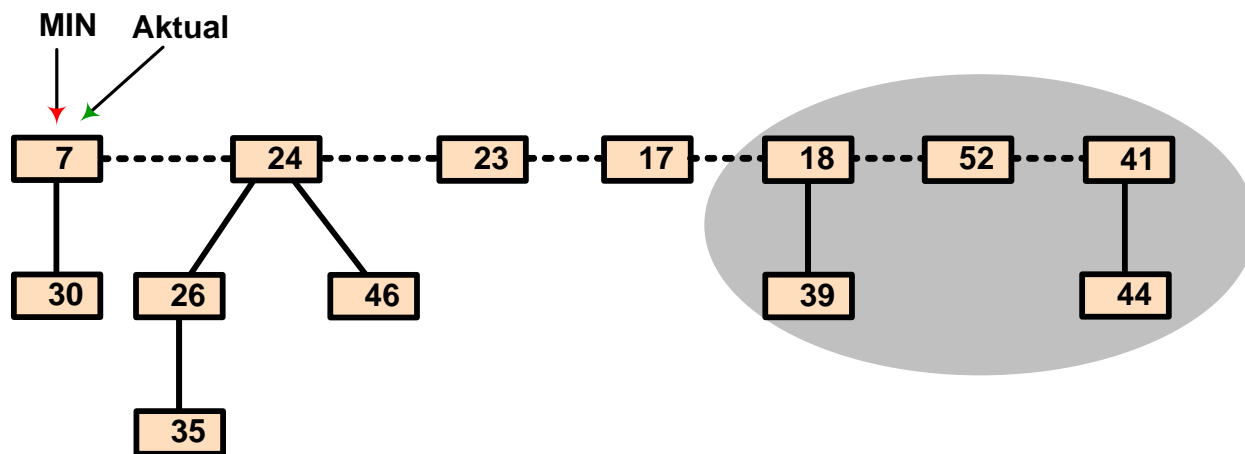
Pre potreby tejto fázy algoritmu sa používa pomocné pole, ktorého i -tá položka indikuje, či sa vo Fibonacciho halde aktuálne vyskytuje koreň so stupňom i . Stupeň koreňa je ohraničený \Rightarrow vieme ohraničiť veľkosť poľa.

V prípade aplikácie operácie Merge sa prvku s väčšou hodnotou (ktorý sa stal synom) nastaví Značka na true (označí sa). Odteraz sa sleduje, či stratil synov. Ak stratí druhého, odznačí sa.

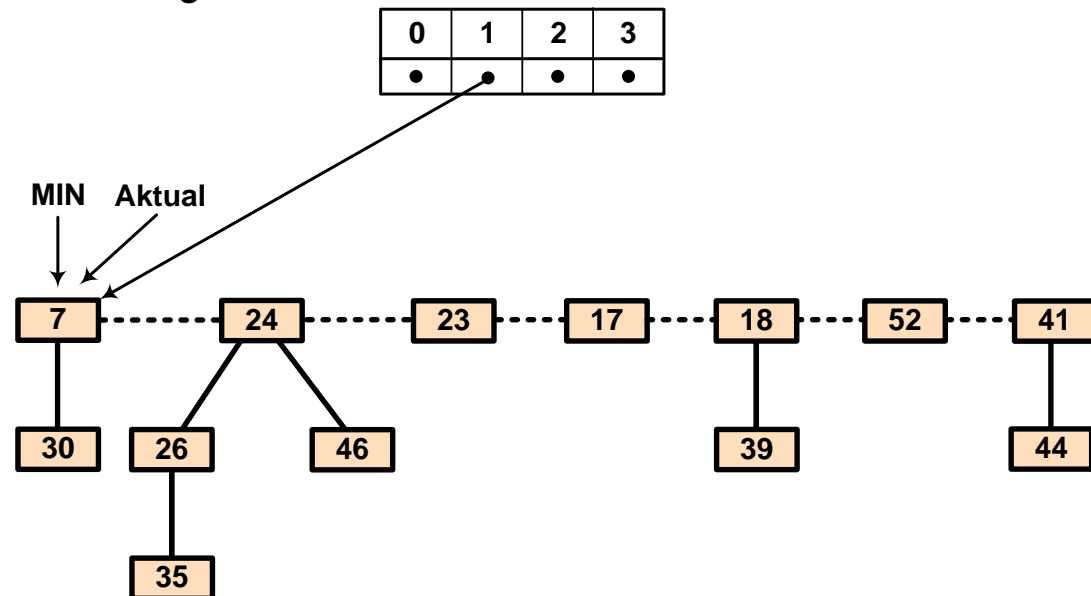
Príklad:

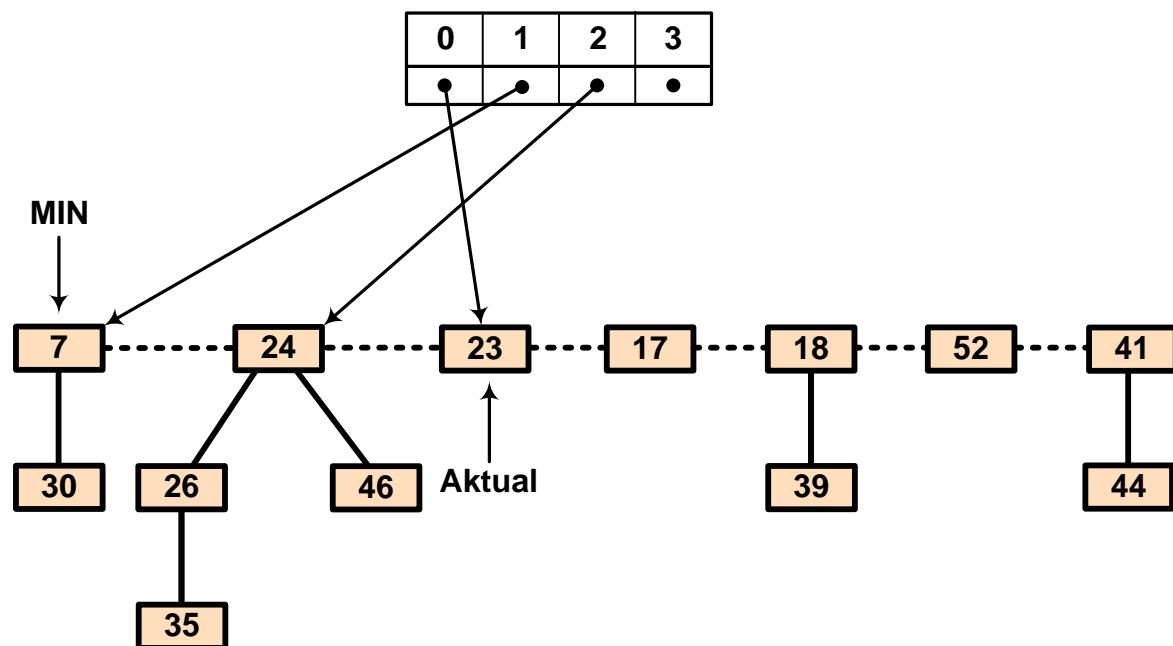
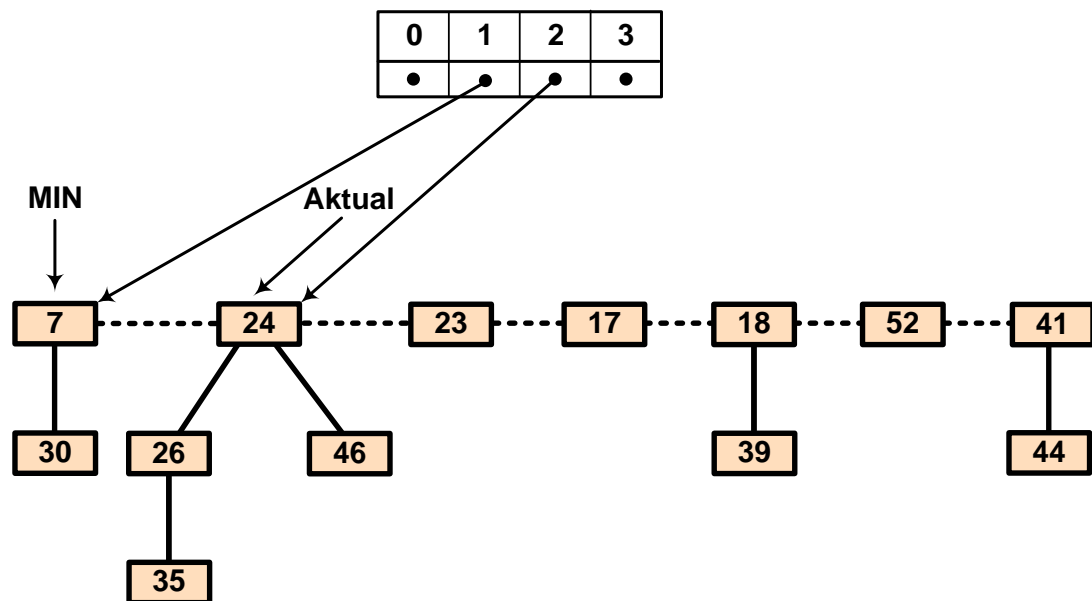
1.fáza algoritmu - odobratie: (v ďalšom texte sa nezobrazuje značenie)

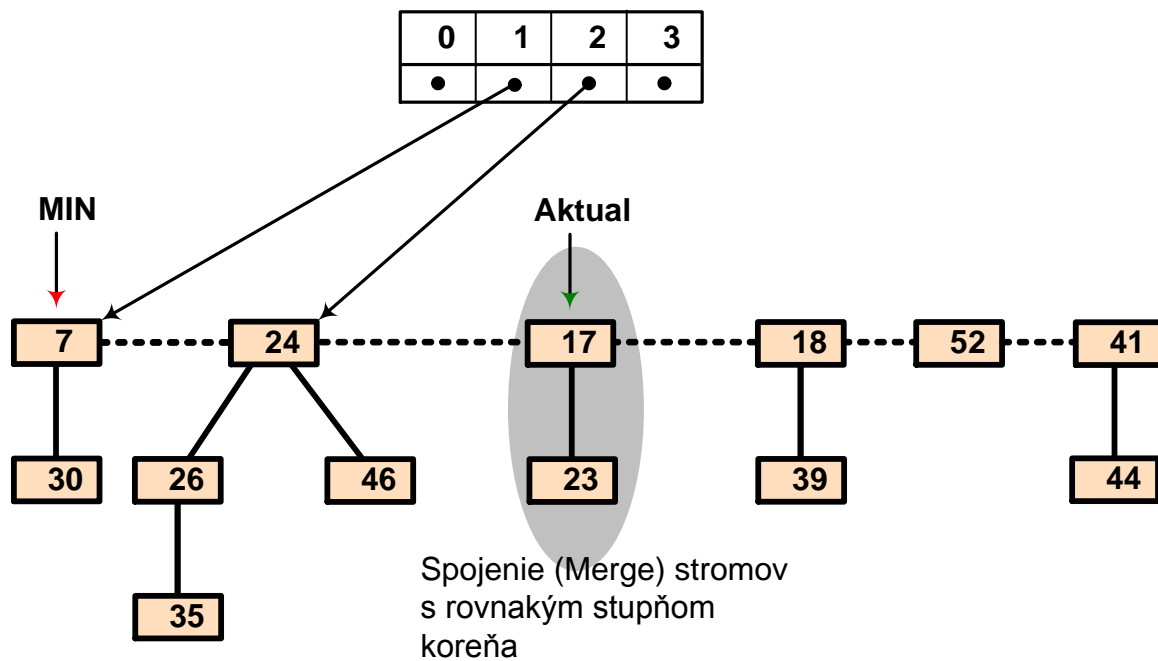
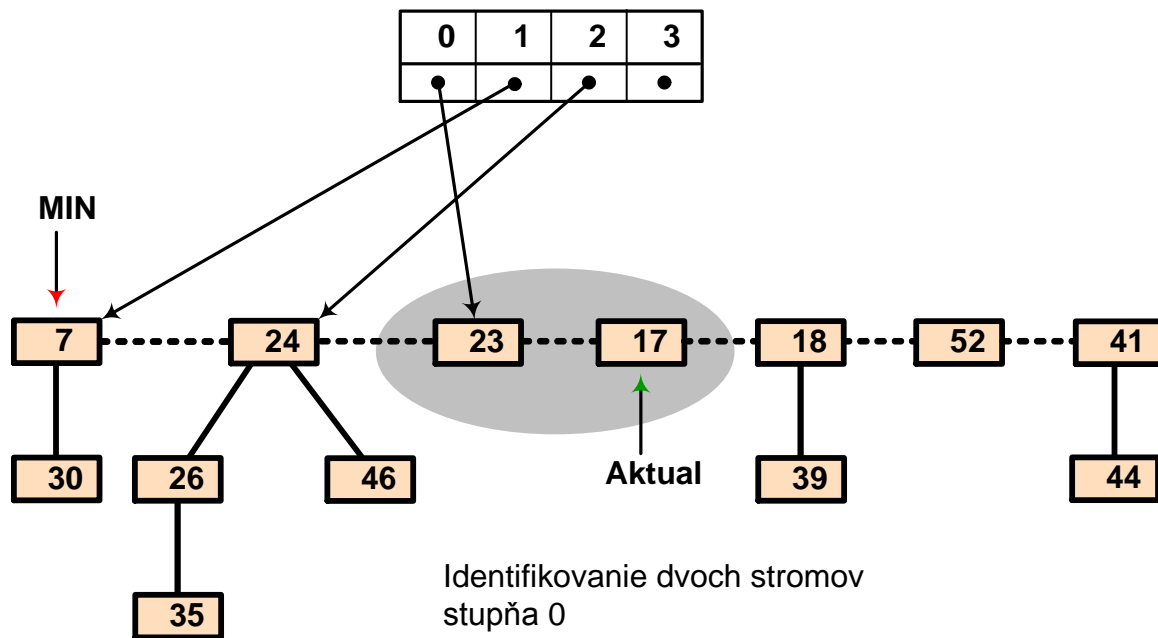


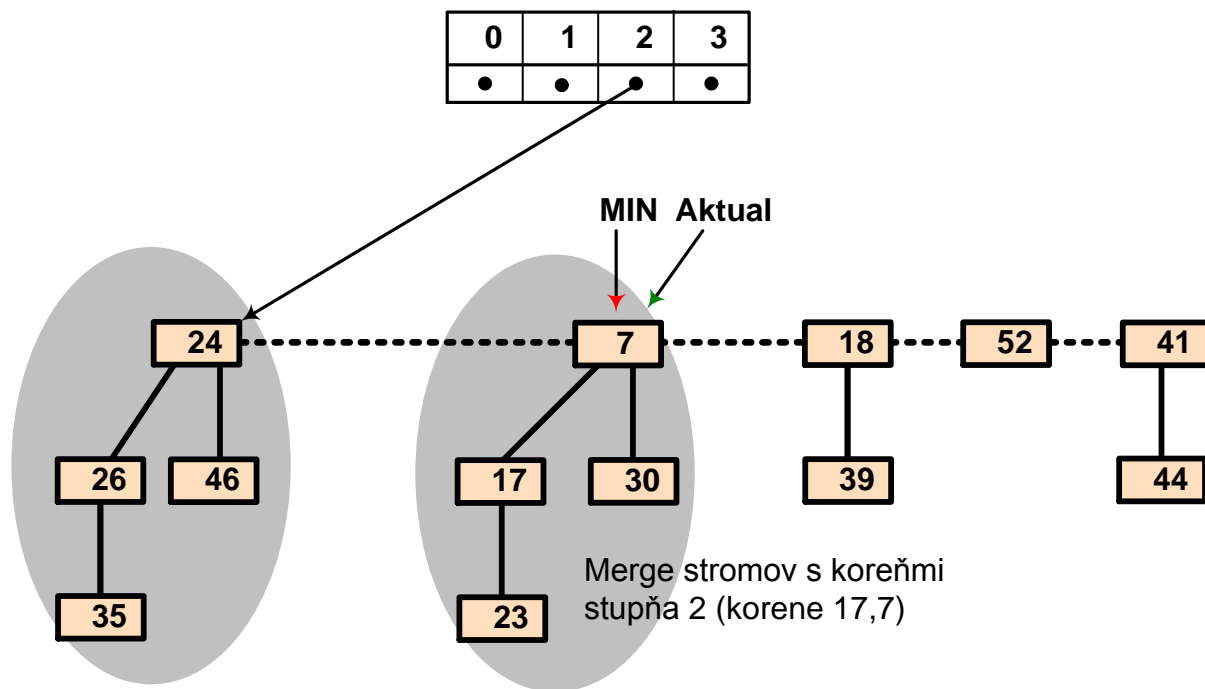


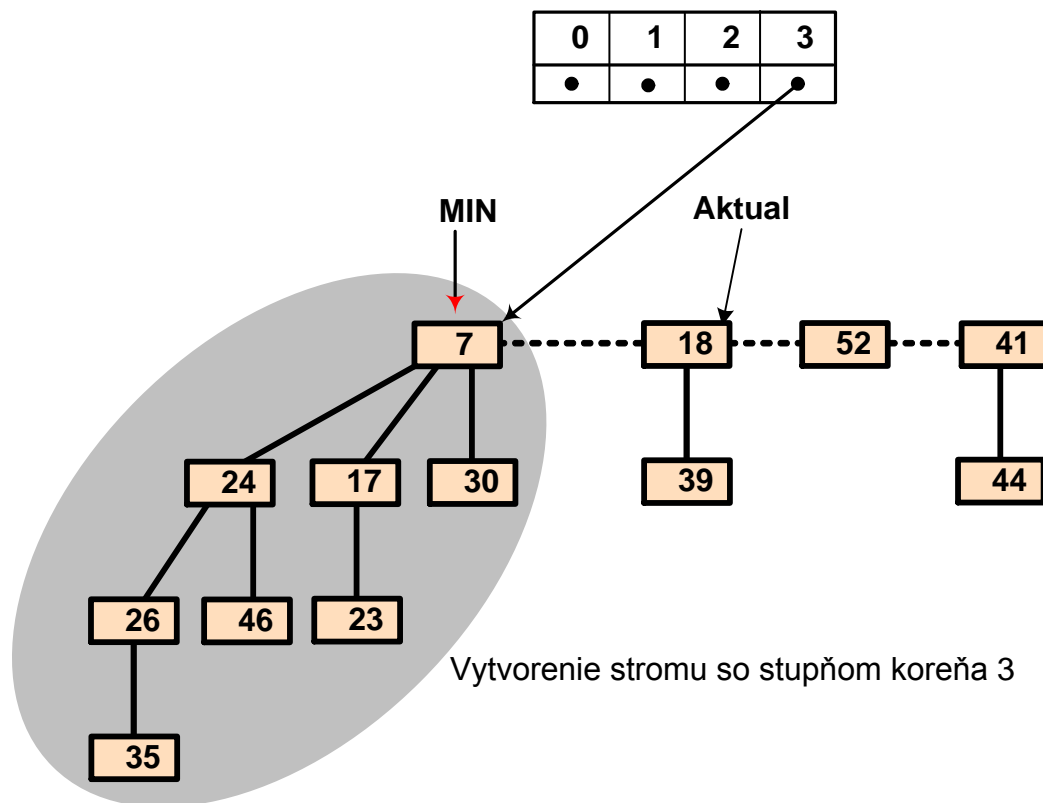
2. fáza algoritmu - konsolidácia:

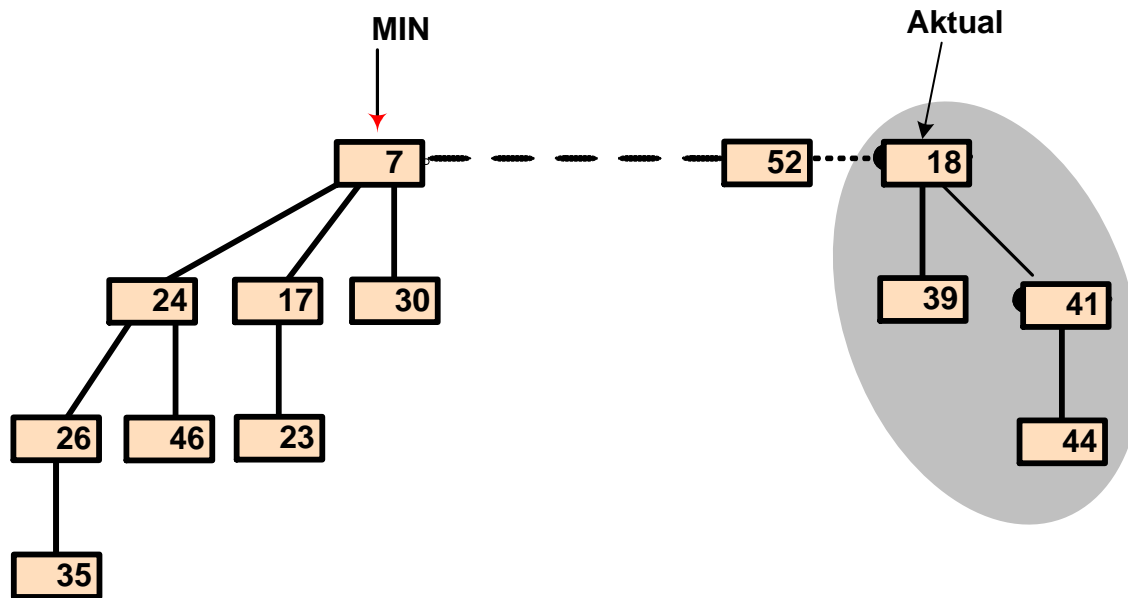
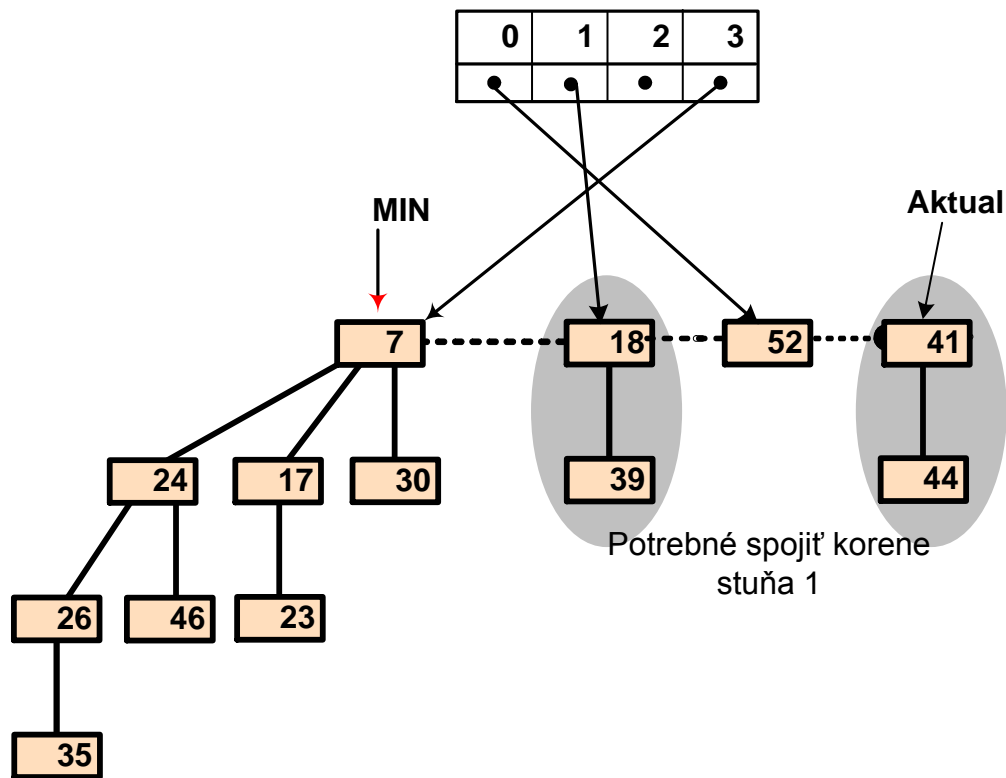












Materiál slúži výlučne pre študentov FRI ŽU, nie je dovolené ho upravovať, prípadne ďalej šíriť.

Fibonacciho halda podporuje efektívnu implementáciu ďalších dvoch „rozširujúcich“ a často potrebných operácií:

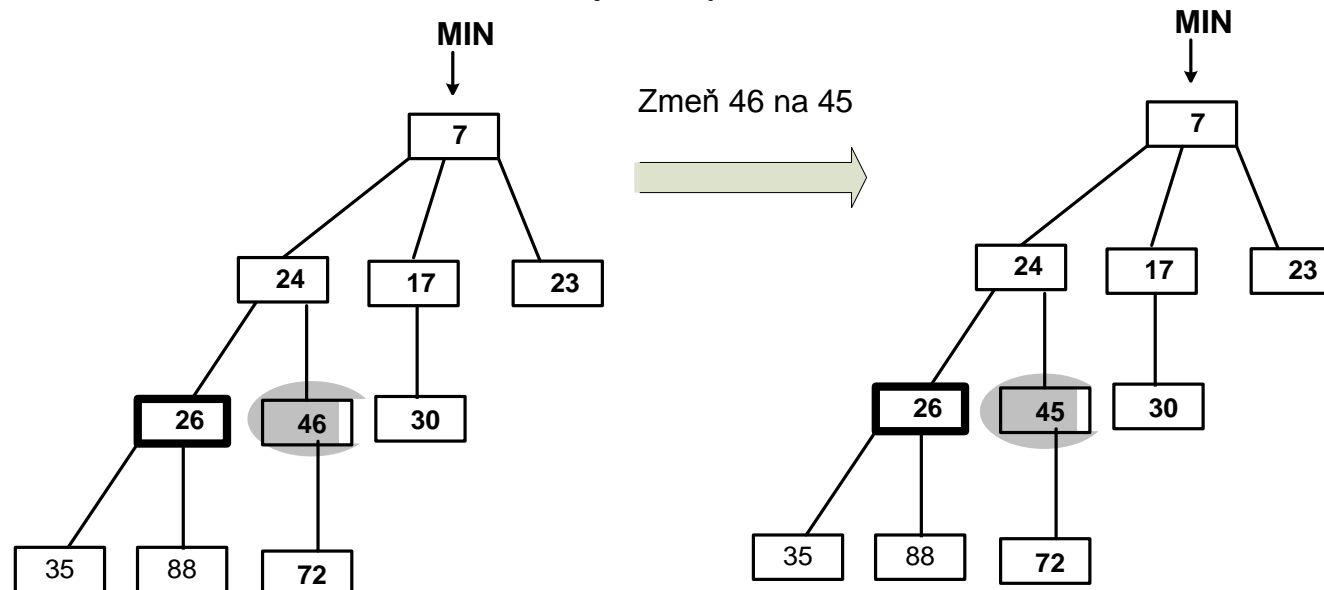
- zvýšenie priority zadaného (ľubovoľného) prvku,
- odobratie ľubovoľného prvku.

Operácia zvýšenia priority – priemerná (amortizovaná) zložitosť $O(1)$, najhoršia zložitosť $O(n)$:

- **v zložitosti nie je započítané získanie (nájdanie) samotného prvku**
- nech k je nová hodnota priority

Prípad 0 : Zmenou priority sa vlastnosti haldy nenarušia

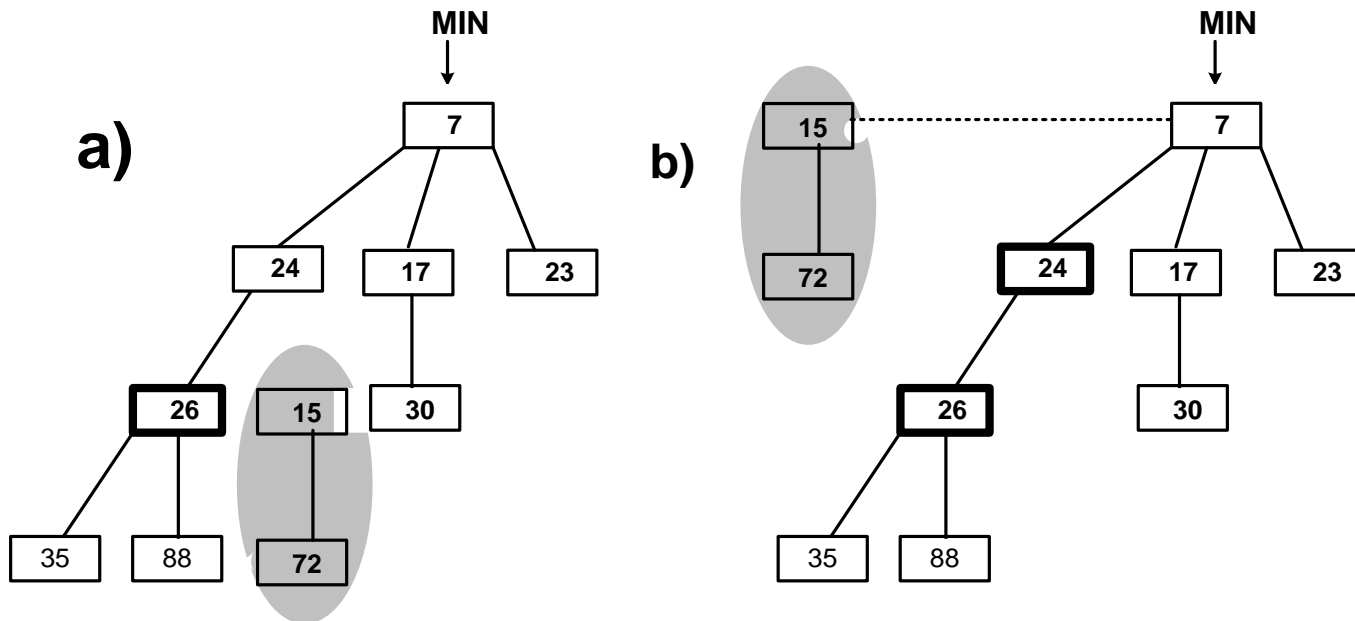
- zvýš priority uzla X na k
- ak treba, aktualizuj MIN pointer



Prípad 1: Narušenie vlastností, otec X je neoznačený

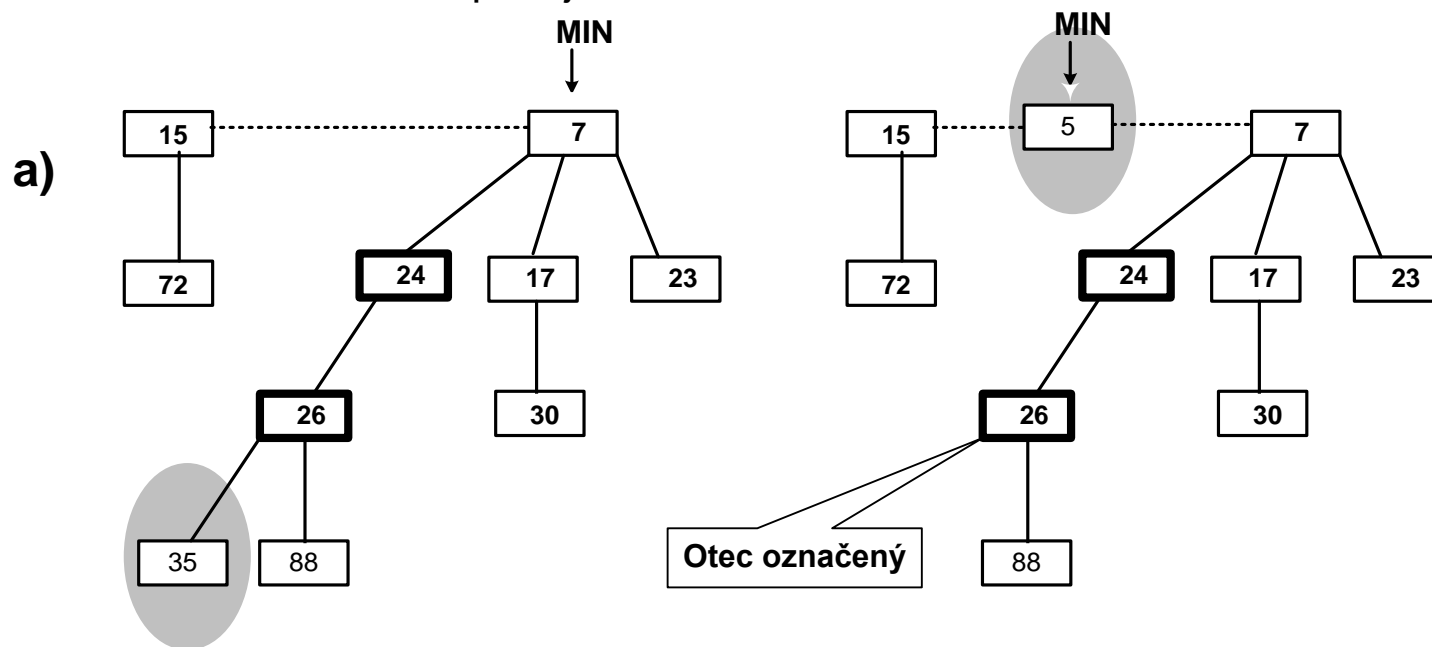
- a) - zvýš prioritu uzla X na k
 - preruš spojenie X s jeho otcom
- b) - označ otca
 - strom s koreňom X vlož do zoznamu koreňov (operácia zjednotenia dvoch háld), prípadne zmenš MIN pointer

Zvýš prioritu 45 na 15:

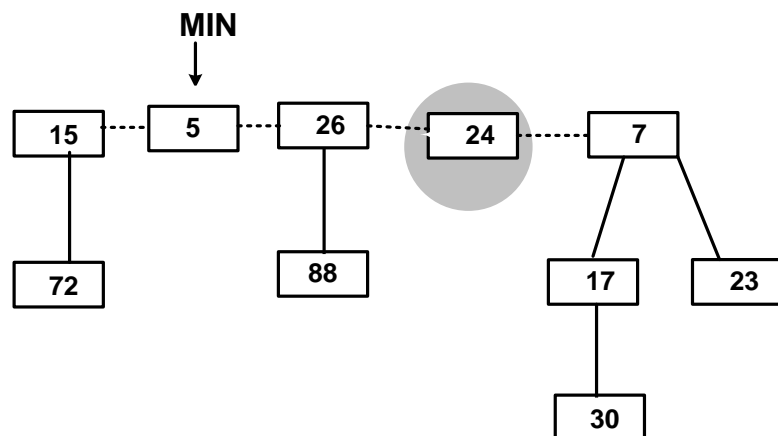
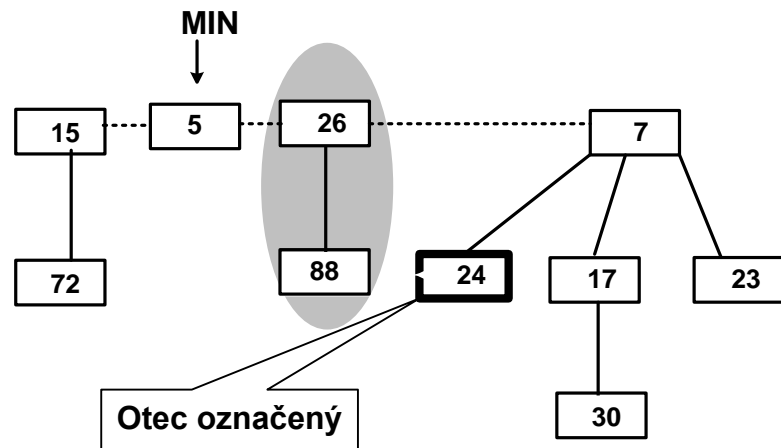


Prípad 2: Narušenie vlastností, otec X je označený
(ilustrácia: zvýš prioritu 35 na 5)

- a) zvýš prioritu uzla X na k
zruš spojenie X a otca $p(X)$,
vlož strom s koreňom X do zoznamu koreňov (operácia zjednotenia dvoch háld)
- b) zruš spojenie medzi $p(X)$ a jeho otcom $p(p(X))$
vlož $p(X)$ do zoznamu koreňov, nastav MIN pointer
ak $p(p(X))$ neoznačený, označ ho
ak $p(p(X))$ označený, odpoj ho, odznač ho, vlož ho do zoznamu koreňov a opakuj



b)



Operácia zníženia priority – priemerná (amortizovaná) zložitosť $O(1)$, najhoršia zložitosť $O(n)$:

V prípade narušenia vlastností haldy (syn/synovia) majú vyššiu prioritu sa títo synovia postupne odpoja, ich otcovia sa označia a sú presunutí do zoznamu koreňov. Ak bol otec označený, tak sa tiež odpojí a vloží do zoznamu koreňov, pričom sa odznačí.

Operácia vymaž prvok – priemerná zložitosť $O(\log n)$, najhoršia zložitosť $O(n)$:

- zvýš prioritu odoberaného prvku na maximálnu možnú (napr. $-\infty$),
- aplikuj operáciu **Vymaž minimum**
- **v zložitosti nie je započítané získanie (nájdenie) samotného prvku**