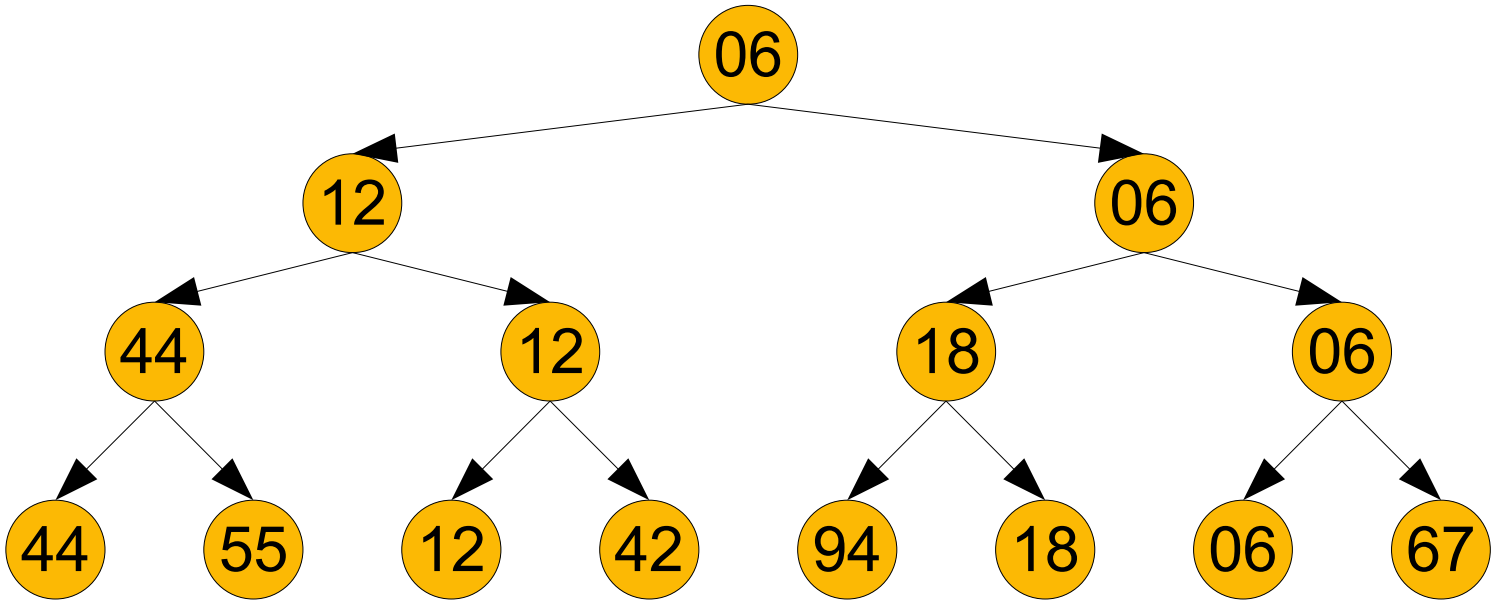
Heap Sort

Co lze vylepšit viz Selection Sort a další

- Jak by šlo vylepšit třídění výběrem?
- Vybírá vždy nejmenší prvek z n prvků, n-1 prvků, ...
- Musíme si zapamatovat více informací při výběru
  - pomocí n/2 porovnání můžeme zjistit menší klíč každé dvojice prvků
  - pomocí n/4 porovnání můžeme zjistit menší klíč z každé dvojice menších prvků ...

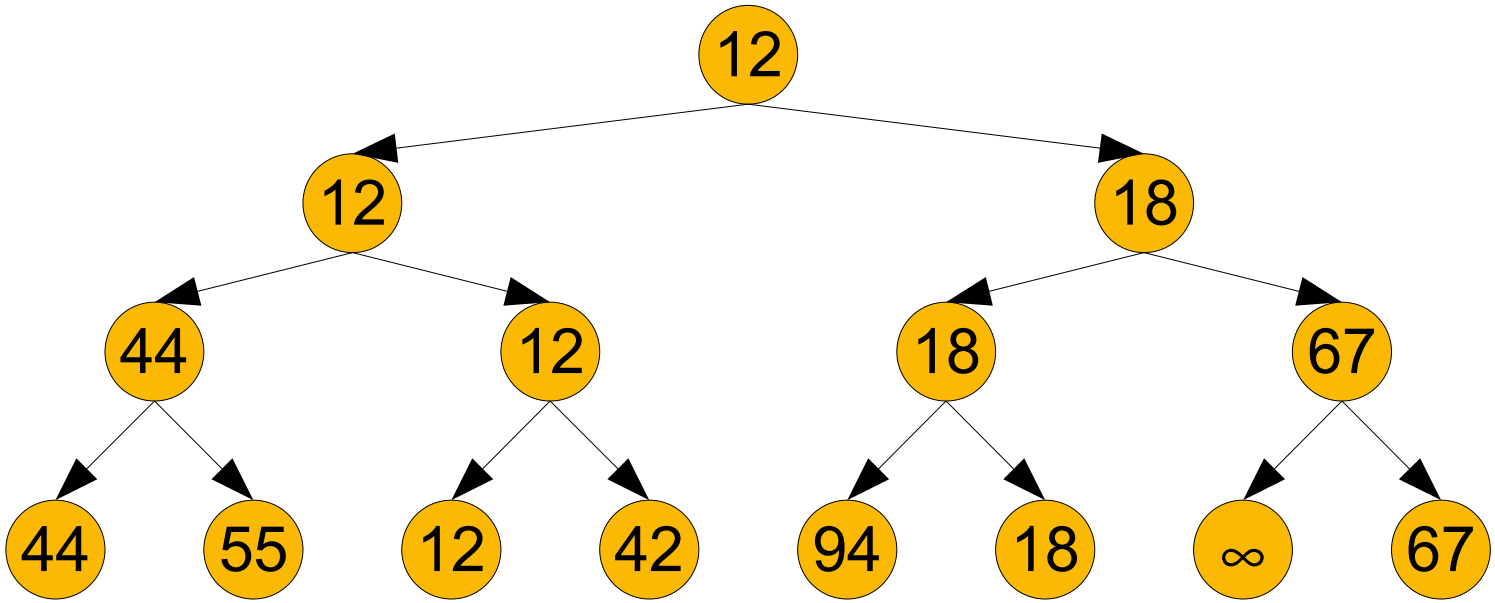
Strom prvků

- Do listů stromu dáme všechny prvky, do uzlů dáme vždy menší z obou prvků



Strom prvků

- Do listů stromu dáme všechny prvky, do uzlů dáme vždy menší z obou prvků
- Při řazení odebereme vždy nejmenší prvek, nahradíme jej ∞ a opravíme strom



Třídění

- Každý krok výběru – $\log n$ porovnání
- Potřebujeme n krát opakovat
- To je dobré, ale. . . .
  - nesplňujeme základní požadavek třídění na místě
  - potřebujeme pomocnou strukturu
- Řešení: pokusit se vhodně uspořádat data v poli

Halda

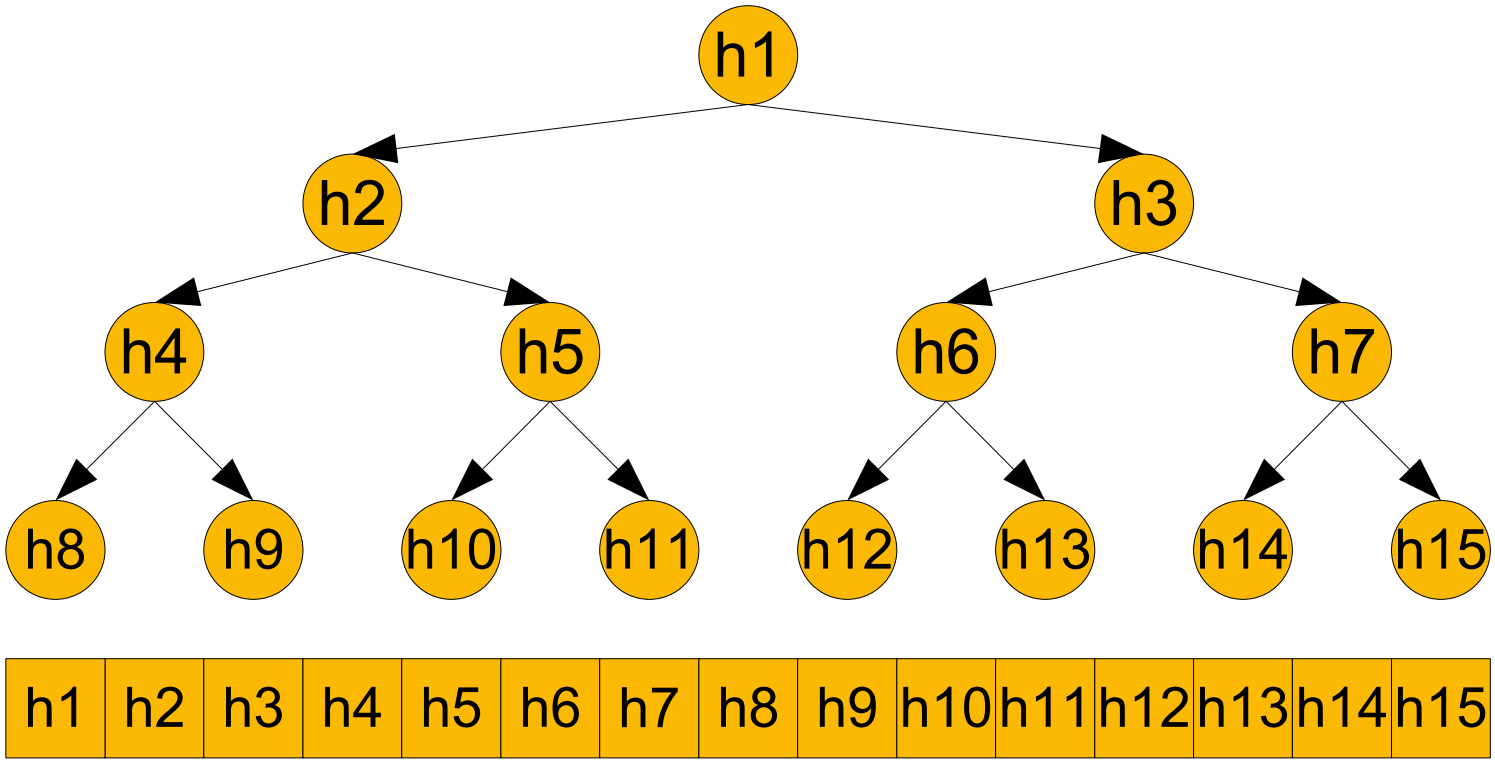
- Halda – posloupnost klíčů $h_d, \ldots, h_h$ takových že
  - $h_i \leq h_{2i}$
  - $h_i \leq h_{2i+1}$
- Co to znamená?

Halda

- Halda – posloupnost klíčů $h_d, \ldots, h_h$ takových že
  - $h_i \leq h_{2i}$
  - $h_i \leq h_{2i+1}$
- Co to znamená?
  - binární strom můžeme reprezentovat polem

Příklad

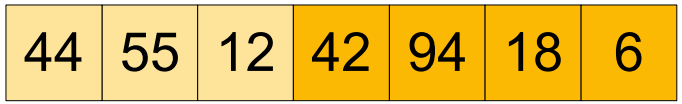
- Binární strom a odpovídající pole



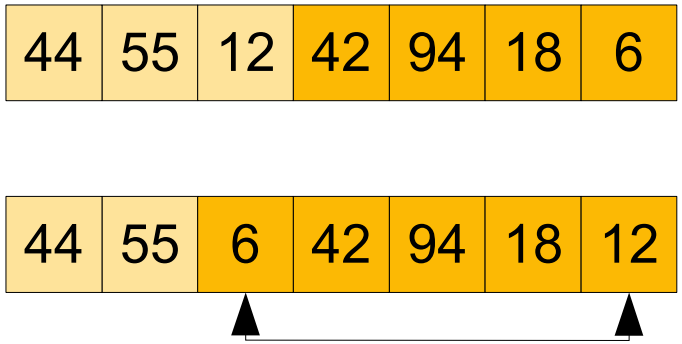
Sestrojení haldy

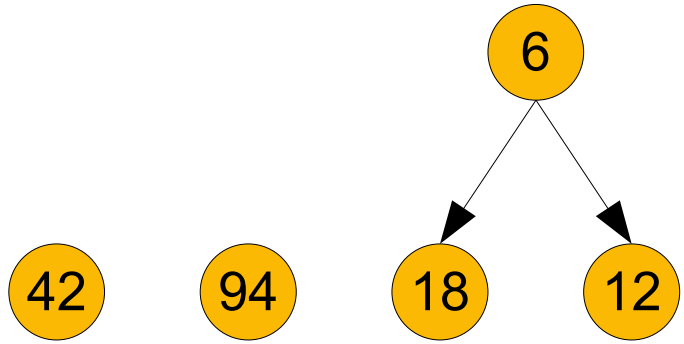
- Jak vlastně haldu sestrojit?
- Máme pole $h_0..h_{n-1}$, prvky $h_{n/2}..h_{n-1}$ už tvoří haldu (spodní vrstva stromu)
- Halda se rozšiřuje každým krokem doleva o nový prvek, který se musí dostat na správnou pozici

Příklad sestrojení haldy

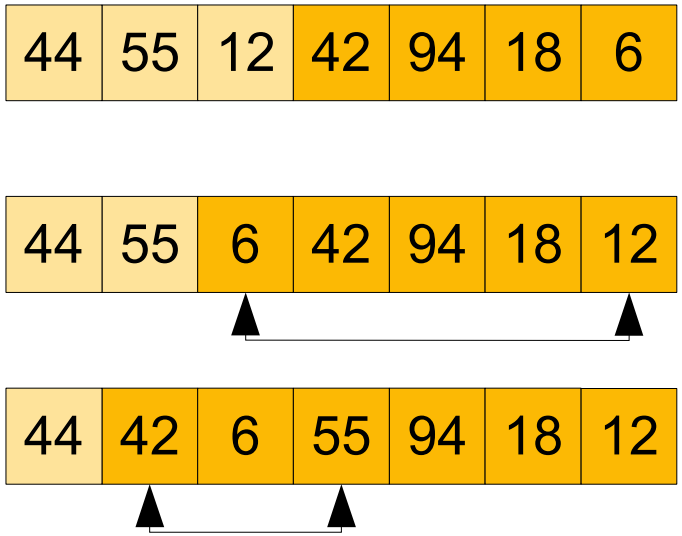


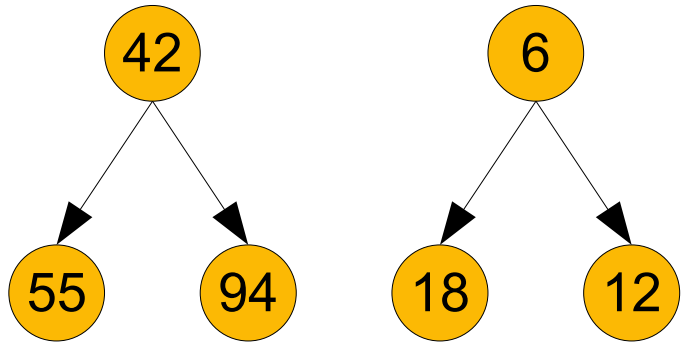
Příklad sestrojení haldy



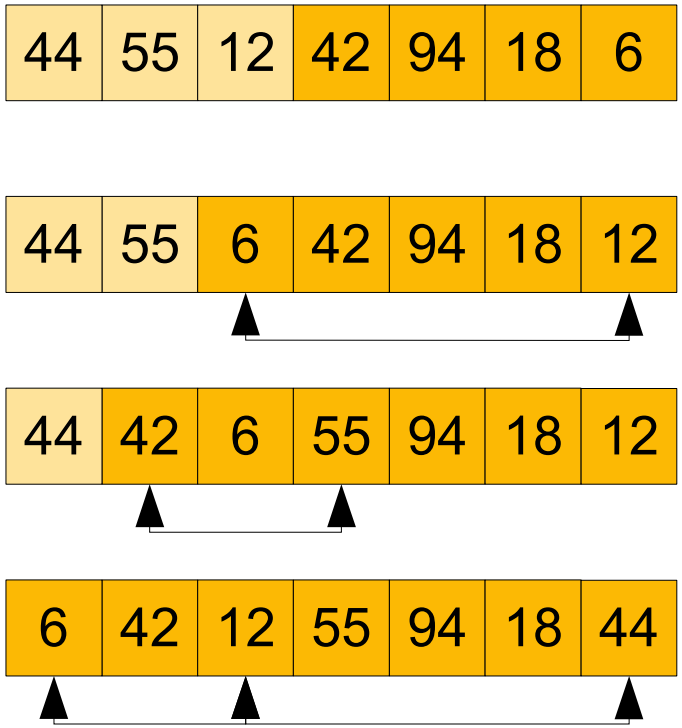


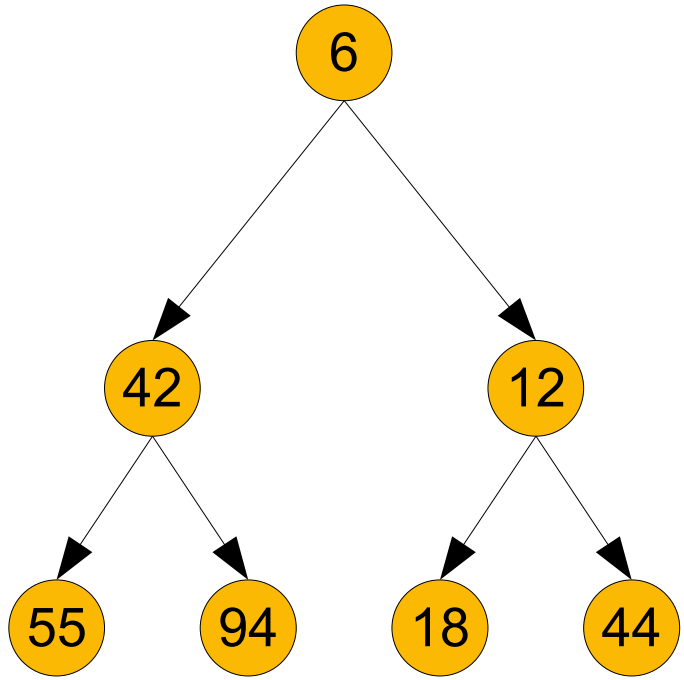
Příklad sestrojení haldy





Příklad sestrojení haldy



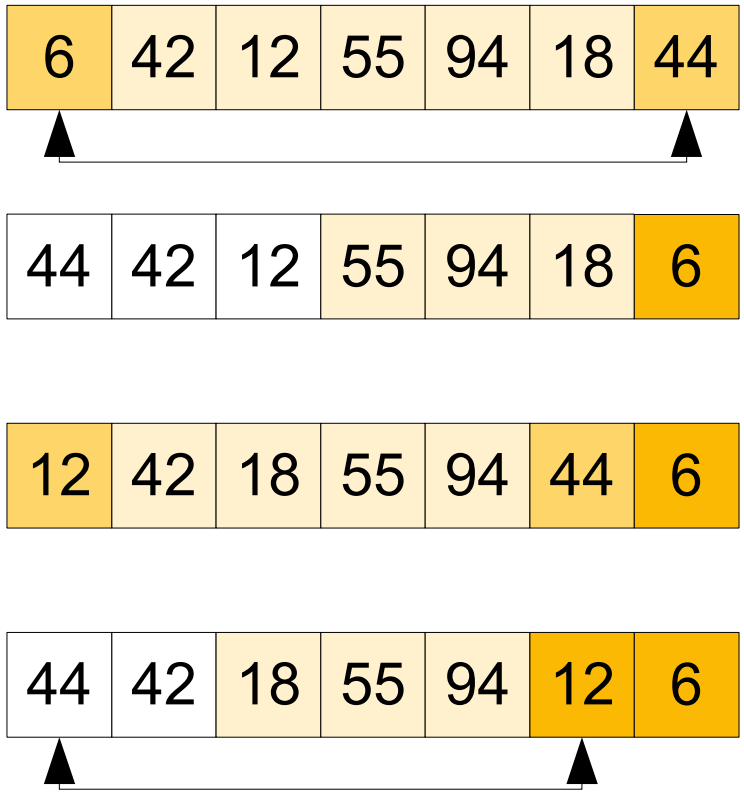


Třídění haldou

- Když už umíme vytvořit haldu haldu, zbývá poslední otázka – jak toho využít ke třídění?
  - v rámci každého kroku třídění:
    - vybereme poslední prvek z haldy (x)
    - na jeho místo dáme kořen
    - znovu vytvoříme haldu, z polem o velikosti n-1 a zatřídíme tím x na správné místo
- Bude nám to fungovat dobře?

Příklad třídění

- Bude nám to fungovat dobře?
  - ne úplně



- Pole je setříděné opačně
  - stačí zaměnit relace $<$ a $>$

Heapsort

- Složitost $O(nlogn)$
- Kompletní algoritmus:
  - Např. http://en.wikipedia.org/wiki/Heapsort

Konec