

ZALG 4. cvičení

Zopakování

- Vyvážené binární stromy

STL knihovny pro binární stromy

- `<set>`
- `<map>`

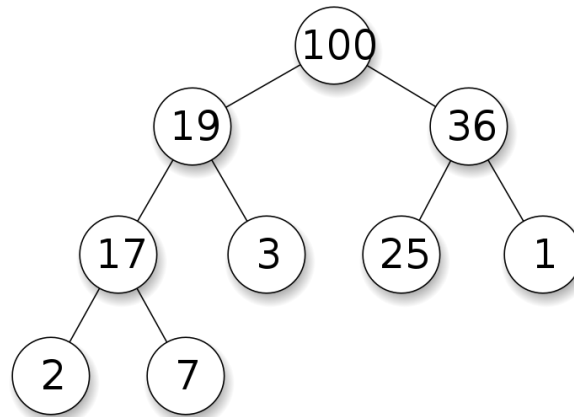
Halda (Heap)

- Specializovaná stromová struktura
- Skoro dokonalý strom – všechna patra se zaplňují postupně
- Splňuje takzvanou vlastnost haldy: Pro každý vrchol V , jehož rodič je vrchol P platí, že hodnota ve vrcholu P je větší než hodnota ve vrcholu V – někdy nazývaný max heap
- Min heap – opak max heap
 - Pro každý vrchol V , jehož rodič je vrchol P platí, že hodnota ve vrcholu P je menší než hodnota ve vrcholu V

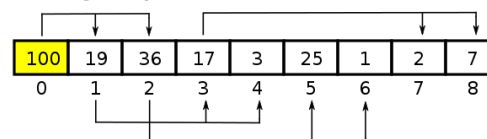
(Binární haldá)

- Často se používá k implementaci datové struktury prioritní fronta
- V haldě je prvek s největší prioritou (nebo nejmenší) vždy uložen v kořeni

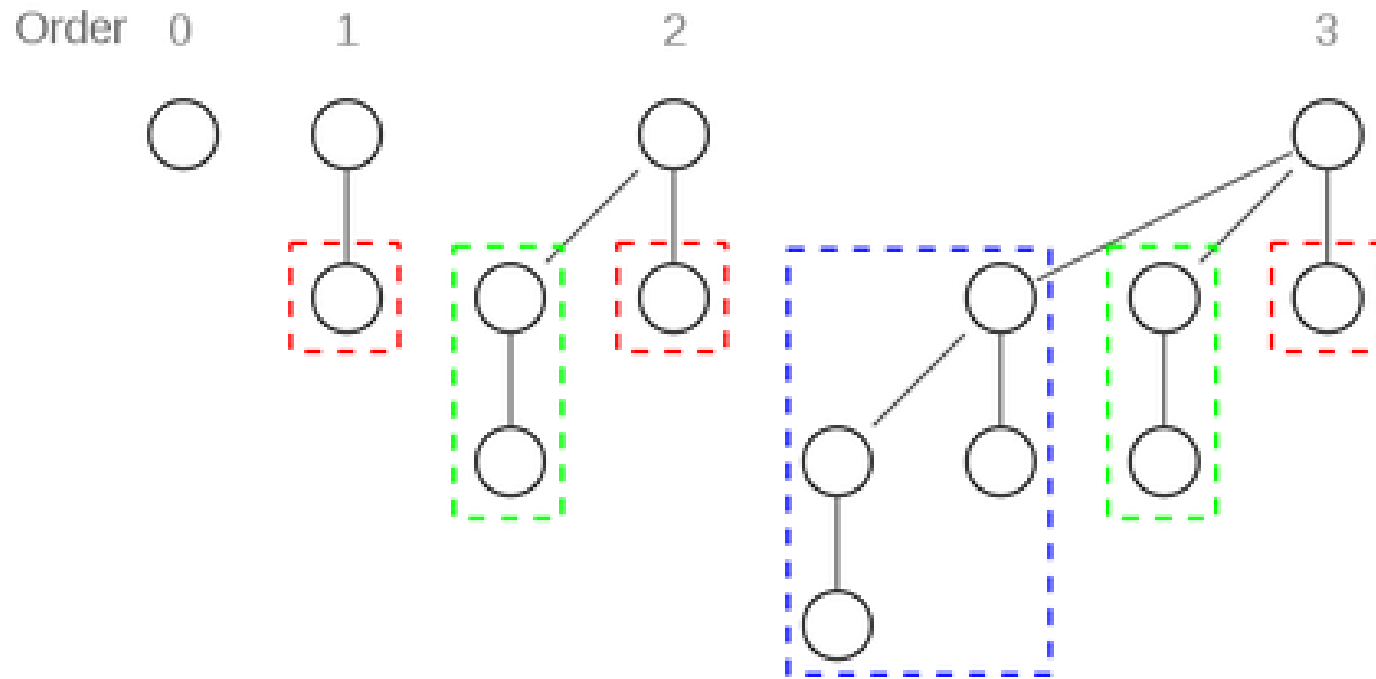
Tree representation



Array representation



5. Zápočtové téma – (~~Fibonacciho~~) Binomiální halda



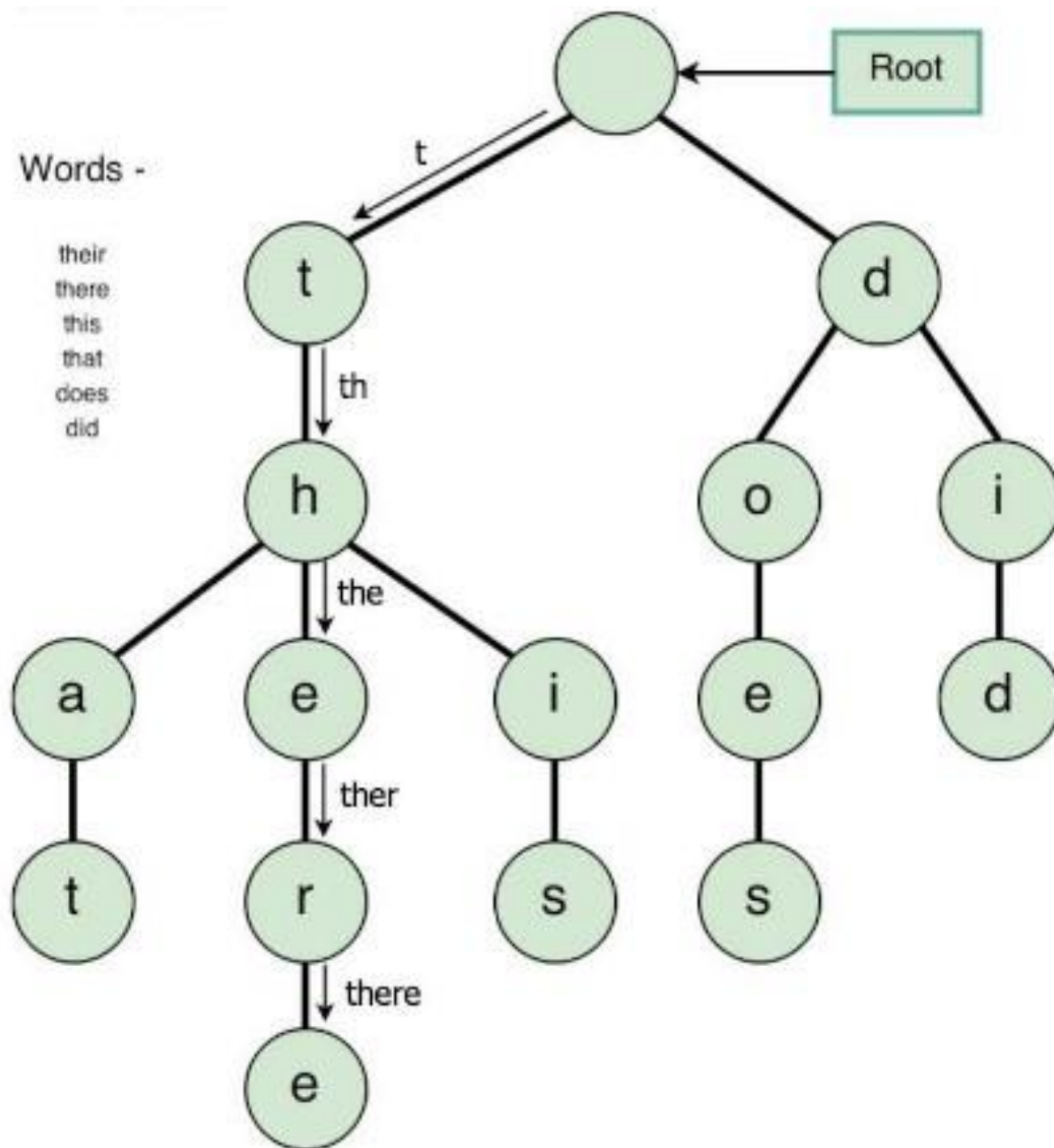
- There are a total of 2^k nodes in the tree.
- The tree has $\binom{k}{i}$ nodes at depth i .

6. Zápočtové téma (B-strom)

- Stačí 2-3-4 strom

Trie

- Datová struktura pro uchovávání dvojic klíč-hodnota, kde klíč je obvykle řetězec



7. Zápočtové téma – Trie a Suffix Tree

- Naprogramovat trie a popsat sufix tree

N-tice

- `std::tuple`
- `std::pair`
 - `first`
 - `second`

Přetěžování operátorů metod

- Budeme chtít přetížit operátor [] pro indexování datové struktury

Vracející L-hodnotu

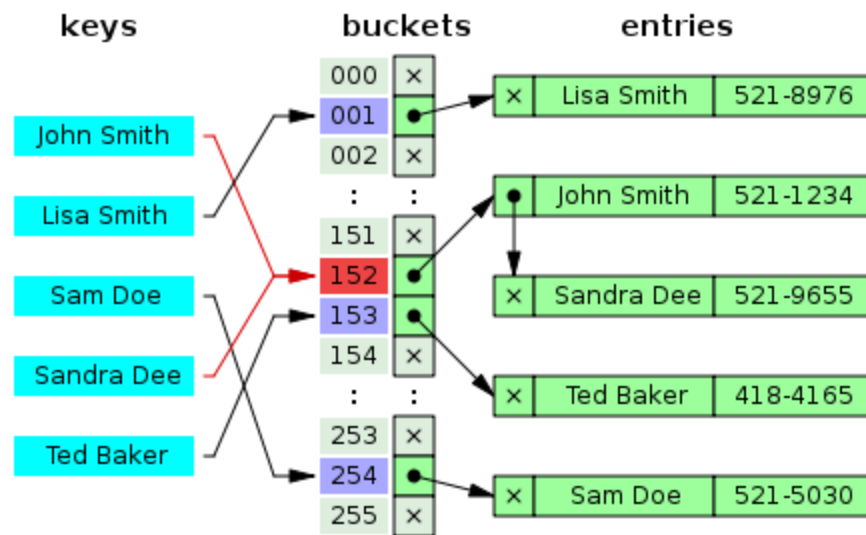
- `Value& operator[](const Key& key) {...}`

Vracející R-hodnotu

- `const Value& operator[](const Key& key) const {...}`

Hešová tabulka (Hash map nebo Hash table)

- Datová struktura implementující asociativní pole nebo slovník
- Abstraktní datový typ, který mapuje nějaký klíč k hodnotě (například stringový klíč jméno k integerové hodnotě věku)



Hešová tabulka

- Implementace je obvykle pomocí pole spojových seznamů – to slouží k obcházení kolizí, že různé klíče mohou být namapované na stejný index
- Dále využívá hešovací funkci, která spočte index, do kterého seznamu se vloží dvojice klíč-hodnota
- Pod stejným klíčem může být v hešové tabulce jen jeden záznam

Hešovací funkce

- Prvky v určitém seznamu v poli nemají stejné klíče, ale hešovací funkce se vstupní hodnotou těchto klíčů vrátí stejný index – proto je nutné ukládat nejen hodnotu, ale i klíč
- Chceme využívat hešovací funkci, která dobře distribuuje klíče, aby nedocházelo ke kolizím, což by snižovalo rychlost vyhledávání prvku s daným klíčem

Hešová tabulka

- Data tedy nejsou ukládána v seřazeném pořadí
- Složitost ukládání nebo vyhledávání prvků se může blížit $O(1)$ když zanedbáme nějaké kolize, které mohou nastat

Implementace

- Privátní složky
 - Počet kyblíků
 - Vektor spojových seznamů (pro nás přesněji vektor vektorů), do kterých se vkládá dvojice klíč-hodnota
 - Hešovací funkce
- Public složky
 - Konstruktor – alokuje vektor o maximální velikosti
 - Metoda pro vložení
 - Přetížený operátor [] – budou 2, jeden vracející referenci a jeden konstantní referenci
 - Metodu pro odstranění prvku podle klíče
 - Metoda, která zkontroluje, zda se prvek s daným klíčem v tabulce nachází