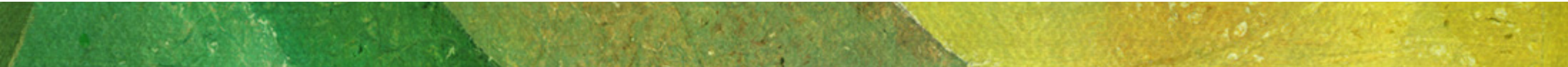
Dynamic Hashing

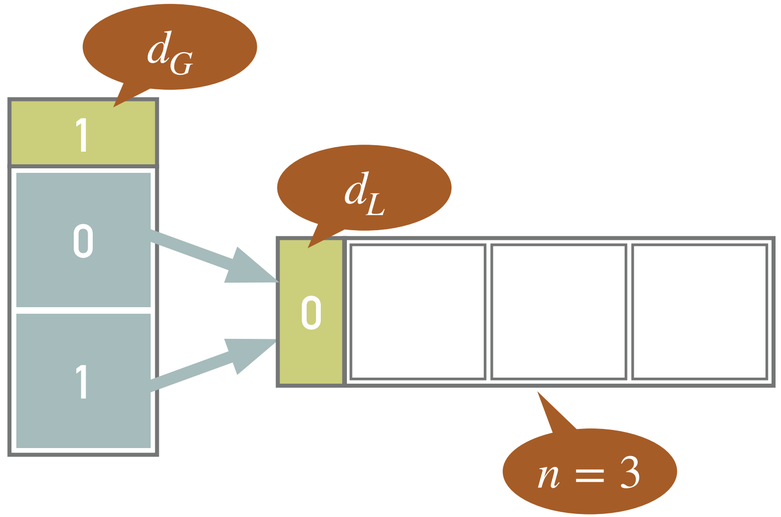
NDBI007: Assignment 3



Task 1: Fagin

- Use *Fagin* hashing method and *insert assigned keys in a given order* into primary file

- Use the following initial settings:
  - *Global depth* $d_G = 1$
  - For each page, *local depth* $d_L = 0$ and *capacity* $n \in \langle 3,4 \rangle$
    - I.e., choose the capacity and explain the reason for the choice
  - Use d_L least significant bits of the *hash* $h(k_{10}) = k_2$ to store a key k into the particular page



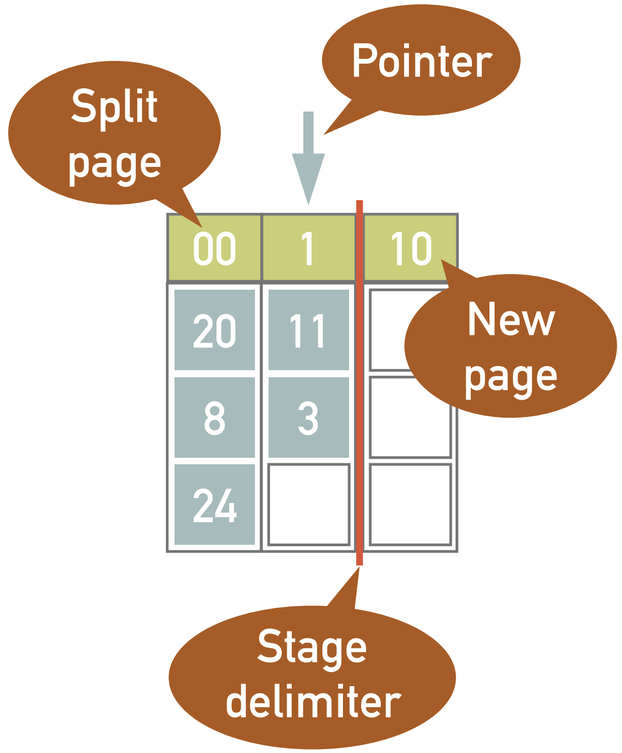
- *Compute* all the parameters *and illustrate* the directory and primary file changes
  - After each addition you may only note changes
  - After each page split, illustrate the structure
- *Or*, you may *implement* the Fagin method *and log* all events (i.e., submit the source code and, e.g., makefile)
  - The permitted languages are Java, Python, C, C++, and Swift

- **Points:** 1

Task 2: Litwin

- Use *Litwin* hashing method and *insert assigned keys in a given order* into primary file

- Use the following initial settings:
  - *Stage* $d = 0$
  - At the beginning of stage d, the pointer points to page 0
  - Page capacity $n \in \langle 3,5 \rangle$
  - *Pre-defined condition*: splitting occurs after $\langle 2,3 \rangle$ inserts
    - I.e., choose the condition and explain the reason for the choice and an the expected consequence
  - Use *hash function* $h_d(k_{10}) = k_2$ for pages not yet split, i.e., the least significant d bits of the hashed value $h_d(k_{10})$
  - Use *hash function* $h_d(k_{10}) = k_2$ for the already split pages



- *Compute* all the parameters *and illustrate* the primary file changes
  - After each addition you may only note changes
  - After each page split, illustrate the structure
- *Or*, you may *implement* the Fagin method *and log* all events (i.e., submit the source code and, e.g., makefile)
  - The permitted languages are Java, Python, C, C++, and Swift

- **Points:** 1

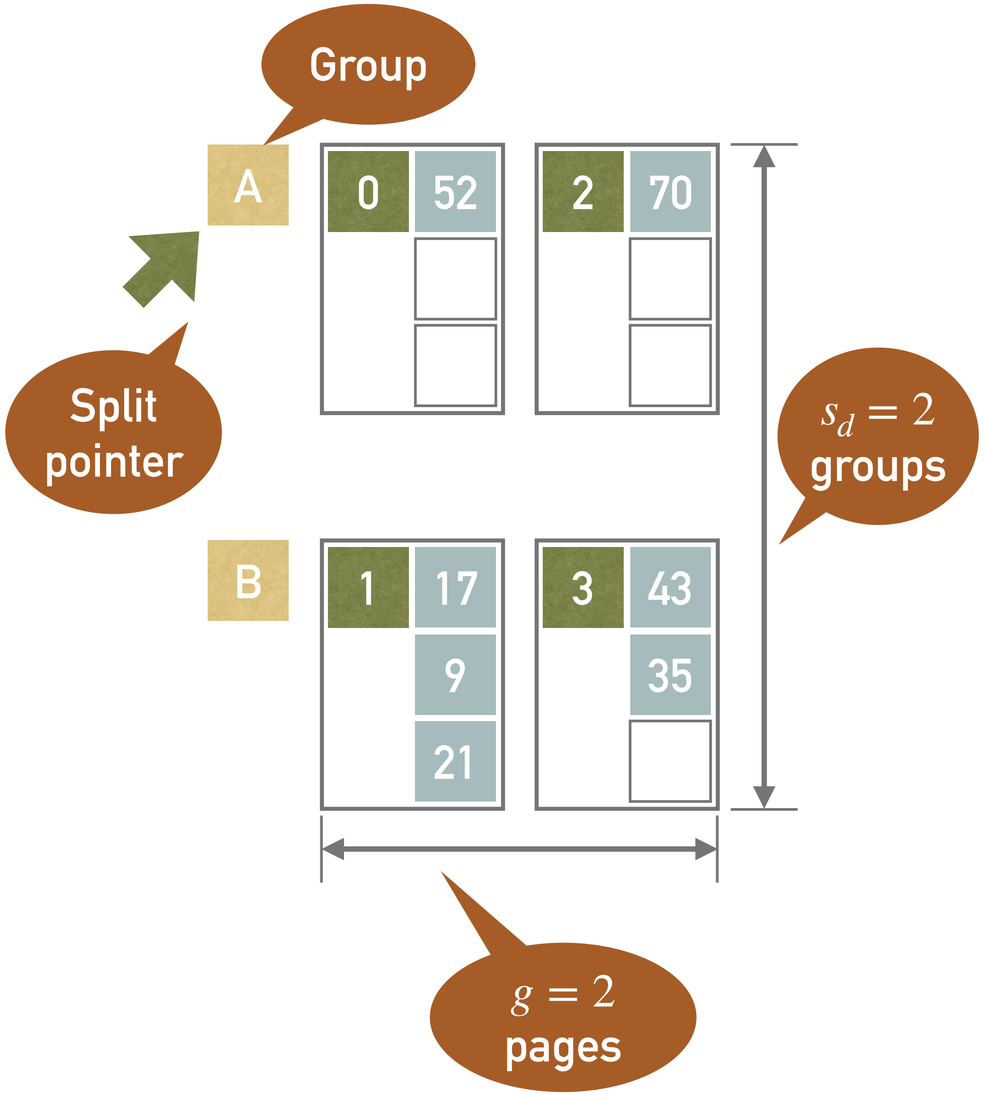
Task 3: LHPE-RL

- Use *LHPE-RL* hashing method and *insert assigned keys in a given order* into primary file

- Use the following initial settings:
  - *Stage* $d = 0$, the primary file consists of $p_0 \in \{4,6\}$ *pages*, each page has *capacity* $b = 3$
  - *Pages are grouped* into $s_d = p_d \div g$ groups, each group has $g = 2$ pages (at the beginning of a stage d)
    - I.e., choose the p_0 and explain the reason for the choice and an the expected consequence
  - *Pre-defined condition*: $L = 2$
    - The *first split* occurs after insertion of $p_0 \bullet L$ records
    - Each *additional split* occurs regularly after L additional inserts
  - Use function $h_0(k) = k \mod p_0$ to determine into which of p_0 initial pages a record in inserted at the beginning
  - Use function $h_1(k) = k \mod 3$ to determine where the records are inserted when a group splits for the first time
  - Use function $h_2(k) = (k \div 3) \mod g_2$, where g_2 is a number of pages in a group after the second split
  - Propose and use an appropriate function $h_3(k)$ if necessary

- *Compute* all the parameters *and illustrate* the primary file changes
  - Note changes after each addition
  - Illustrate the structure only after each page insertion or virtual reorganization

- **Points:** 1



Bonus Task 4

- Explain the *difference* between *LHPE* and *LHPE-RL* methods
  - You may illustrate the difference with an example

- **Points:** 1