



# Static Hashing

---

*NDBI007: Assignment 2*





# Task 1: Cormack

---

- ❖ Use *Cormack* perfect static hashing method and *insert assigned keys* in a given order into directory of size  $s = 7$ 
  - ❖ The assignment of keys is *attached in a separate file*
- ❖ Use the following hashing functions:
  - ❖ *Primary hashing function* is given as  $h(k, s) = k \bmod s$
  - ❖ *Secondary hashing function* is  $h_i(k, r) = (k \gg i) \bmod r$
- ❖ *Compute* all the parameters *and illustrate* the directory and primary file changes with each step
  - ❖ After each addition you may only note changes in the structure of the directory and the primary file
  - ❖ Illustrate the structure after addition of all keys
- ❖ Or, you may implement the Cormack 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:** 2.5

# Task 2: Larson & Kalja

---

- ❖ Use *Larson & Kalja* static hashing method and *insert assigned keys* in a given order into a structure that consists of  $n$  pages
  - ❖ The assignment of keys is *attached in a separate file*
  - ❖ Select  $n \in \langle 3, 5 \rangle$  as a page number
  - ❖ Select page *capacity* within range  $\langle 3, 4 \rangle$
  - ❖ *Initial separator* values are set to  $111_2$
- ❖ To get the *page number* in which the data should be inserted, use *hash function*  $h_i(k) = (k + i) \bmod n$
- ❖ To get the *signatures*, apply *function*  $s_i(k) = (k \gg i) \bmod 7$ 
  - ❖  $i$  stands for the number of *previously unsuccessful inserts*
- ❖ *Note* all the computations and *illustrate* the result
- ❖ Or, you may implement the Larson & Kalja 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: 2.5**

# Bonus Task 3

---

- ❖ Can we use *arbitrary hashing functions* for Cormack or are some more or less suitable?
  - ❖ In particular, we are interested in appropriate selection of primary and secondary hashing functions
  - ❖ Provide an *informal reason* for your answer *and example*, i.e., no need to attach a mathematical proof.
- ❖ Help:
  - ❖ G. V. Cormack, R. N. S. Horspool, and M. Kaiserwerth, Practical perfect hashing, The Computer Journal, 28(1), 54-58, Oxford University Press (1985)
  - ❖ D. Knuth, The Art of Computer Programming, vol III, Addison-Wesley, Reading (1973)
- ❖ **Points: 1**