Lecture 9: File Organization

Last Day: Serial Files:

- Queries, Updates
- Batch Update
- Co-Sequential Processing

Today: Direct Files.

- Hashing
- Collisions
- Examples

Folk & Zoellick §10.1, 10.2, 10.3

Direct Files

- To access a record in a serial file, all previous records must be accessed first.

  eg. To access record 100, we must first access records 1, 2, 3, ... 98, 99.

  ∴ Accessing record 100 is much slower than accessing record 1.

- With direct files, records can be accessed directly without accessing other records first

  eg. Retrieving record 100 is just as fast as retrieving record 1.

- This requires a direct access storage

Central Problem

Given the key for a record, how do we find the record (without scanning the entire file)?

A Solution: Hashing

- From a key value, compute the address of the record.
- ie, Apply a function, F, to the key to obtain the record address.
- ie, F(key) = address

Collisions

- Problem: Several different keys may hash to the same address (collide).
- ie, $f(key_1) = f(key_2) = f(key_3)$
- In this case, where should the records for key_1, key_2 & key_3 be stored?
- Solving this problem is called collision resolution, and will occupy much of our time.
- Note: As a file fills up with records, collisions become more likely.

Load Factor

$$\text{Load Factor} = \frac{\#\text{ records in file}}{\text{max. \# records the file can hold}}$$

- $0 \leq \text{Load Factor} \leq 1$
- Load Factor = 0 for an empty file
- Load factor = 1 for a full file.
- As load factor approaches 1, collisions become more likely (so we usually expand the file size).

Hash Functions

- Often, the key is alphanumeric (eg, a name).

- In such cases, hashing usually consists of two steps:

  (1) Convert the key to a number.

  (2) From the number, compute an address.

- With a good hash function, keys are distributed randomly (and uniformly) throughout the file.

Example A

(1) Take every third letter in a key.
Add up the alphabetic positions
of these letters.

eg. Mozart ⟶ 13 + 1 = 14

Tchaikovsky ⟶ 20 + 1 + 15 + 11 = 47

(2) Given a number, n, use n mod k
as the record address,
where k = file size (ie, maximum number
of records).

"Division Remainder".

Key	Numeric Equivalent	(Mod 16) Address	
Mozart	14	14	
Tchaikovsky	47	15	
Ravel	23	7	
Beethoven	44	12	Collision
Mendelssohn	44	12	
Bach	10	10	
Greig	16	0	Collision
Rachmaninoff	57	9	
Vivaldi	32	0	
Chopin	19	3	

K = 16

Load Factor = 10/16 = .625

Collisions = 2

Example B: Mid-Square Method

(1) Concatenate the alphabetic positions of the first and last letters in the key. Square the result.

eg. M O Z A R T

M → 13, T → 20

$13 \longrightarrow 1320 \longrightarrow (1320)^2$

$= 1742400$

(2) Take middle 2 digits of the squared number as the address

eg. 1742400 → 24

Note: Possible addresses are 00 - 99.

Key	Number	Square	Address
Mozart	1320	1742400	24
Tchaikovsky	2025	4100625	6
Ravel	1812	32833344	33
Beethoven	0214	45796	57
Mendelssohn	1314	1726596	65
Bach	0208	43264	32
Grieg	0707	499849	98
Rachmaninoff	1806	3261636	16
Vivaldi	2209	4879681	96
Chopin	0314	98596	85

Example C: Folding Method

① Convert key to a number.

② Partition the number into a number of equal parts, fold over each other, sum, and truncate if needed.

eg. Mozart

M → 13
t → 20

→ 1320 (number)

→ [13 | 20] (par)

→ 13 + 02 (fold over)

→ 15 (sum)

Key	Number	Address
Mozart	1320	13 + 02 = 15
Tchaikovsky	2025	20 + 52 = 72
Ravel	1812	18 + 21 = 39
Beethoven	0214	02 + 41 = 43
Mendelssohn	1314	13 + 41 = 54
Bach	0208	02 + 80 = 82
Greig	0707	07 + 70 = 77
Rachmaninoff	1806	18 + 60 = 78
Vivaldi	2209	22 + 90 = 112 → 12
Chopin	0314	03 + 41 = 44