

Lecture 9: File Organization

Last Day: Serial Files:

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

Today: Direct Files:

- Hashing
- Collisions
- Examples

Walk & Zellck § 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.
- i.e., Apply a function, F , to the key to obtain the record address.
- i.e., $F(\text{key}) = \text{address}$

Collisions

- Problem: Several different keys may hash to the same address (collide).
- i.e., $f(\text{key}_1) = f(\text{key}_2) = f(\text{key}_3)$
- 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 \rightarrow 13 + 1 = 14

Ichaikovsky \rightarrow 20 + 1 + 15 + 11 = 47

- (2) Given a number, n , use $n \bmod k$
as the record address,
where k = file size (i.e., maximum number
of records).

"Division Remainder"

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

Collision

Collision

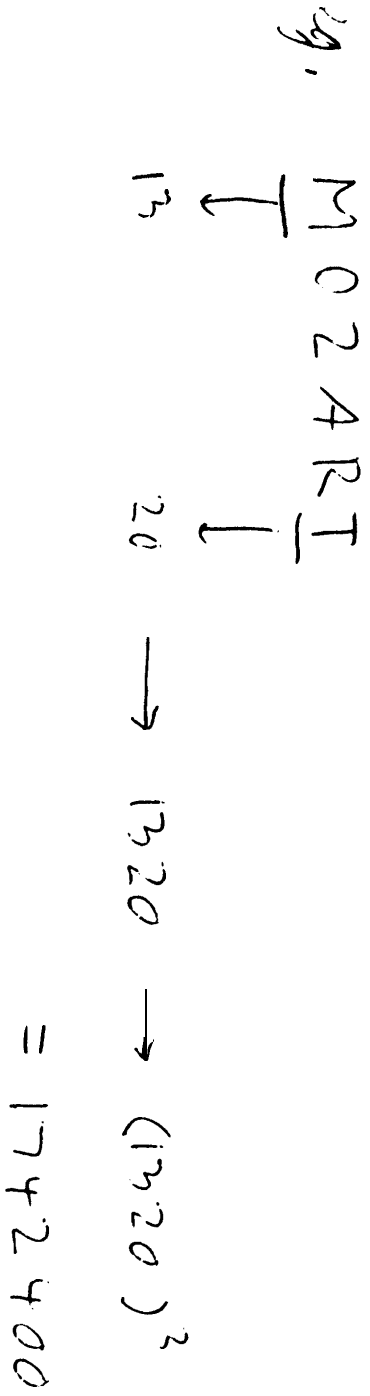
$K = 16$

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.



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

eg. 1742400 → 24

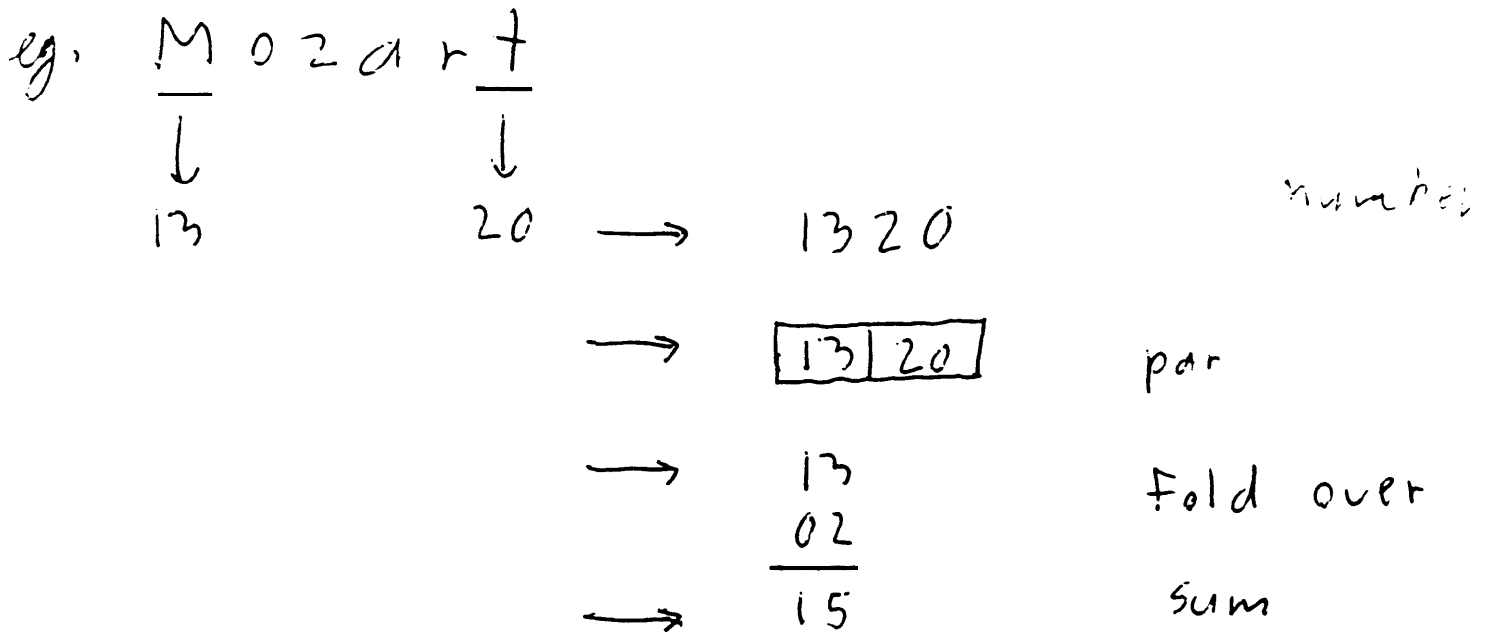
Note: Possible addresses are 00-99.

<u>Key</u>	<u>Number</u>	<u>Square</u>	<u>Address</u>
<u>Mozart</u>	1320	174 <u>2400</u>	24
<u>Tchaikovsky</u>	2025	410 <u>0625</u>	6
<u>Ravel</u>	1812	328 <u>3344</u>	33
<u>Beethoven</u>	0214	4 <u>5796</u>	57
<u>Mendelssohn</u>	1314	172 <u>6596</u>	65
<u>Bach</u>	0208	4 <u>3264</u>	32
<u>Greig</u>	0707	499 <u>849</u>	98
<u>Rachmaninoff</u>	1806	326 <u>1636</u>	16
<u>Vivaldi</u>	2209	487 <u>9681</u>	96
<u>Chopin</u>	0314	98 <u>596</u>	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.



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