# Lecture 12: Direct Files/Collision Resolution

Last Few Days: Open Addressing
(Progressive Overflow)

Today: Separate Overflow

- Organization of overflow file:
  - Serial
  - Open Addressing
  - Chaining

Folk & Zoellick § 10.8

# Collision Resolution

There are two basic approaches:

(1) Open Addressing (Progressive Overflow).

ie, Store the record at some other address in the same file.

(2) Separate Overflow.

ie, Store the record in another file, called the overflow area.

So far, we have looked at Open Addressing

Now, we will look at Separate Overflow.

# Separate Overflow

- Store colliding (overflowing) records in a separate file, called the overflow file (possibly on a separate disk).
- This way, more records are likely to be stored at their home address (since overflow records are not stored at some other record's home address).
- This reduces average access time.
- The overflow file itself may be a serial file or a hash file.

# Example

| key | hash(key) |
|---|---|
| Mozart | 1 |
| Tchaikovsky | 8 |
| Ravel | 10 |
| Beethoven | 5 |
| Mendelssohn | 5 |
| Bach | 10 |
| Grieg | 3 |
| Rachmaninoff | 5 |
| Vivaldi | 6 |
| Chopin | 6 |

# Example (Cont)

hash File

| | |
|---|---|
| 0 | |
| 1 | Mozart |
| 2 | |
| 3 | Greig |
| 4 | |
| 5 | Beethoven |
| 6 | Vivaldi |
| 7 | |
| 8 | Tchaikovsky |
| 9 | |
| 10 | Ravel |
| 11 | |
| 12 | |

Serial Overflow File

| |
|---|
| Mendelssohn |
| Bach |
| Rachmaninoff |
| Chopin |

Note: Average access time to overflow records can be large (because the file is serial).

Final File Contents

# Sample Retrievals

## Retrieve Mozart:

Get record 1 in hash file.

Cost = 1 file access.

## Retrieve Bach:

Get record 10 in hash file. (Not Bach)

Get first record in overflow file. (Not Bach)

Get next record in overflow file. (Bach!)

Cost = 3 file accesses.

## Retrieve Chopin:

Cost = 5 file accesses

# Hashed Overflow Files

- Need two hash functions (hash1 & hash2), and two collision-resolution methods (One for the main file, and one for the overflow File).

## Example

- Main file uses separate overflow.
  Record address = hash1(key).

- Overflow file uses open addressing with linear probing, where

$$PA_i = [hash2(key) + i] \bmod 9$$

step = 1

file size = 9

# Example (Cont)

| key | hash1(key) | hash2(key) |
|---|---|---|
| Mozart | 1 | |
| Tchaikovsky | 8 | |
| Ravel | 10 | |
| Beethoven | 5 | |
| Mendelssohn | 5 | 2 |
| Bach | 10 | 5 |
| Grieg | 3 | |
| Rachmaninoff | 5 | 7 |
| Vivaldi | 6 | |
| Chopin | 6 | 2 |

# Example (Cont.)

| | |
|---|---|
| 0 | |
| 1 | Mozart |
| 2 | |
| 3 | Greig |
| 4 | |
| 5 | Beethoven |
| 6 | Vivaldi |
| 7 | |
| 8 | Tchaikovsky |
| 9 | |
| 10 | Ravel |
| 11 | |
| 12 | |

Main File (hashed)

| | |
|---|---|
| 0 | |
| 1 | |
| 2 | Mendelssohn |
| 3 | Chopin |
| 4 | |
| 5 | Bach |
| 6 | |
| 7 | Rachmaninoff |
| 8 | |

Overflow File (hashed)

## Final File Contents

# Sample Updates and Retrievals

## Retrieve Bach:

Get record 10 in main file. (Not Bach)

Get record 5 in overflow file. (Bach!)

Cost = 2 file accesses.

## Modify Mozart:

Get record 1 in main file. (1 access)

Modify record in MM.

Write modified record back to position 1 of main file. (1 access)

Cost = 2 file accesses.
(1 read + 1 write)

## Modify Chopin:

Get record 6 from main file. (Not Chopin)

Get record 2 from overflow file. (Not Chopin)

Get record 2+1 = 3 from overflow file. (Chopin!

Modify record in MM.

Write modified record back to position 3 of overflow file.

Cost = 3 file reads + 1 file write

= 4 file accesses

# Separate Overflow with Chaining

- Each overflow record is appended to the end of the overflow file (like a serial overflow file).

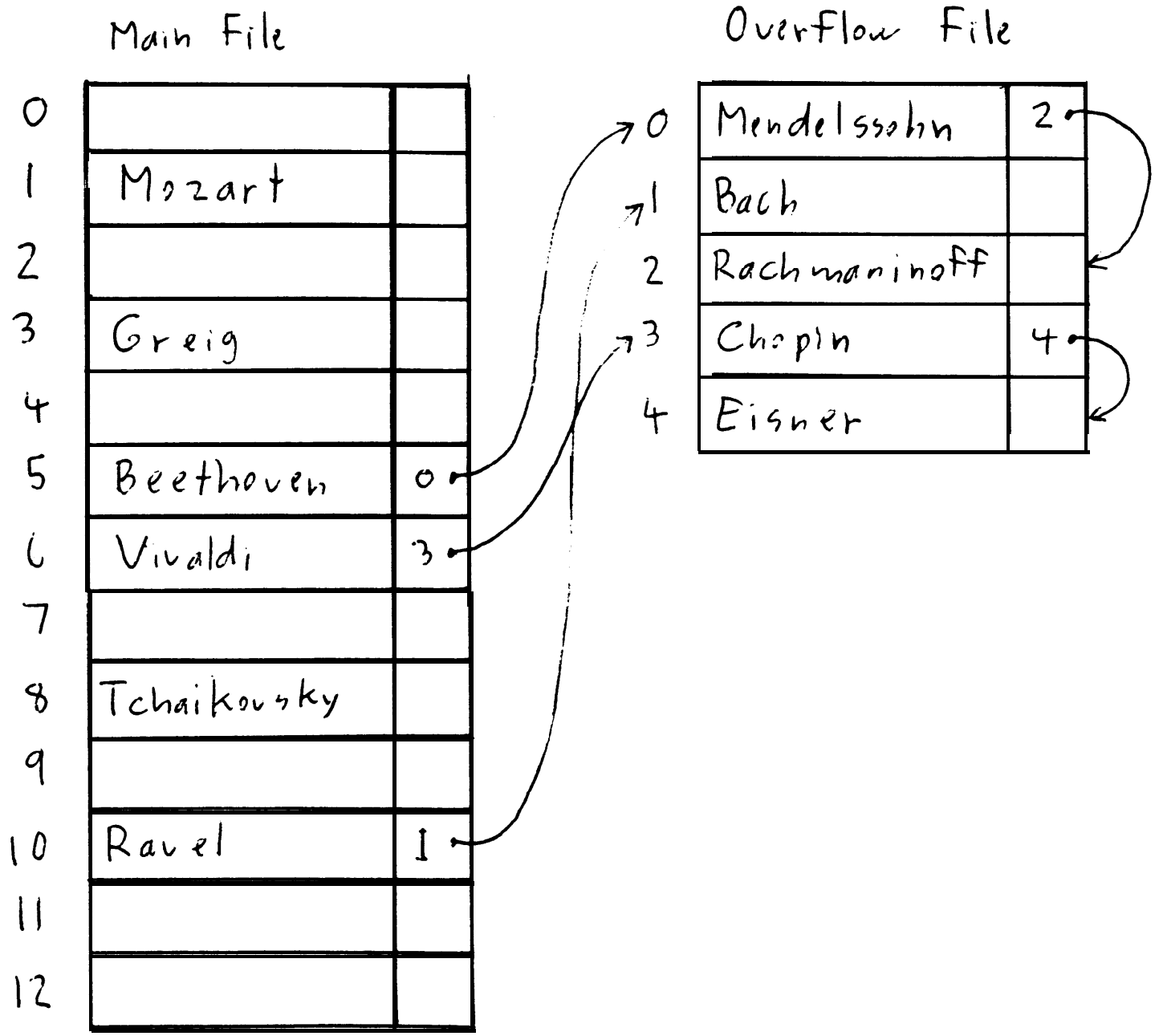
- In addition, all records with the same home address are "chained" together in a linked list (called an overflow chain).

- Each record contains an extra field, which points to the next record in the overflow chain.

# Example

| key | hash(key) |
|---|---|
| Mozart | 1 |
| Tchaikovsky | 8 |
| Ravel | 10 |
| Beethoven | 5 |
| Mendelssohn | 5 |
| Bach | 10 |
| Greig | 3 |
| Rachmaninoff | 5 |
| Vivaldi | 6 |
| Chopin | 6 |
| Eisner | 6 |

# Example (Cont.)




## Final File Contents

# Sample Retrievals

## Retrieve Rachmaninoff:

Get record 5 of main file. (Not Rachmaninoff)

Get record 0 of overflow file. (Not Rachmaninoff)

Get record 2 of overflow file. (Rachmaninoff!)

Cost = 3 file accesses (all reads).

## Retrieve Dvorcak (home address = 5)

Get record 5 of main file. (Not Dvorcak)

Get record 0 of overflow file. (Not Dvorcak)

Get record 2 of overflow file. (Not Dvorcak)

End of overflow chain. (empty pointer)

Fail. (ie, No record is retrieved.)

Cost = 3 file accesses (all reads).

# Summary

- Hashing with a main File of fixed size.

- Two main methods of collision resolution:
  - Separate overFlow
  - Open Addressing (Progressive OverFlow).
    * Special Case: Table Assisted Hashing.