

Data Structures

Maarten Dhondt

Realdolmen

June 23, 2017



Who am I?

- ▶ Master of Engineering: Computer Science (KUL)
 - ▶ Computational informatics
- ▶ Realdolmen: acADDemICT in 09/2015
- ▶ Current project: Planning infrastructure @ Infrabel



Outline

- 1 Introductory Data Structures
 - Array
 - Linked List
 - Hash Table
 - Tree
- 2 Java Collection API & Map API
 - Java Collection API
 - Java Map API
- 3 Advanced Data Structures
 - Stuff...



Outline

1 Introductory Data Structures

- Array
- Linked List
- Hash Table
- Tree

2 Java Collection API & Map API

- Java Collection API
- Java Map API

3 Advanced Data Structures

- Stuff...



What are Data Structures?

Data Structure¹

A way in which data are stored for efficient search and retrieval.
Different data structures are suited for different problems.

- ▶ Data type \neq data structure
- ▶ `java.util.HashSet` vs. hash table
- ▶ array vs. array

¹Encyclopædia Britannica

Outline

1 Introductory Data Structures

Array
Linked List
Hash Table
Tree

2 Java Collection API & Map API

Java Collection API
Java Map API

3 Advanced Data Structures

Stuff...



Array

Definition

- ▶ An indexed set of related elements.²
 - ▶ An assemblage of items that are randomly accessible by integers, the index.³
-
- ▶ Example: linear array



² Oxford Dictionary

³ National Institute of Standards & Technology

Array

Operations

- ▶ `get`
- ▶ `set`
- ▶ `indexOf`



Array

Operations

- ▶ `get`
- ▶ `set`
- ▶ `indexOf`



`get(1)`

Array

Operations

- ▶ `get`
- ▶ `set`
- ▶ `indexOf`



`get(1)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set`
- ▶ `indexOf`



`get(1)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set`
- ▶ `indexOf`



`set(2)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set`
- ▶ `indexOf`



`set(2)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`set(2)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`indexOf(object)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`indexOf(object)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`indexOf(object)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`indexOf(object)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`indexOf(object)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf`



`indexOf(object)`

Array

Operations

- ▶ `get` $O(1)$
- ▶ `set` $O(1)$
- ▶ `indexOf` $O(n)$



Outline

1 Introductory Data Structures

Array
Linked List
Hash Table
Tree

2 Java Collection API & Map API

Java Collection API
Java Map API

3 Advanced Data Structures

Stuff...



Linked List

Definition

A linked list is a data structure in which the objects are arranged in a linear order. Unlike arrays in which the linear order is determined by indices, the order is determined by a pointer in each object.⁴

- ▶ Different types: singly, doubly, multiply, circular, ...
- ▶ Example: doubly linked list



⁴ Introduction to Algorithms By Cormen, Leieron, Rivest & Stein

Linked List

Operations

- ▶ add/remove first/last
- ▶ get/insertAt
- ▶ indexOf



Linked List

Operations

- ▶ add/remove first/last
- ▶ get/insertAt
- ▶ indexOf

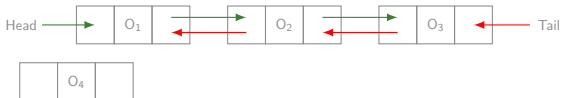


`addFirst(O_4)`

Linked List

Operations

- ▶ add/remove first/last
- ▶ get/insertAt
- ▶ indexOf

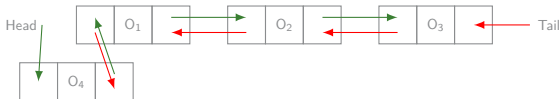


`addFirst(O4)`

Linked List

Operations

- ▶ add/remove first/last
- ▶ get/insertAt
- ▶ indexOf

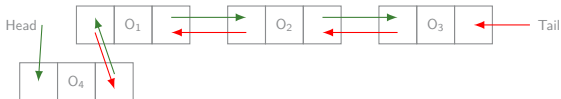


`addFirst(O4)`

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt
- ▶ indexOf



`addFirst(O4)`

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt
- ▶ indexOf

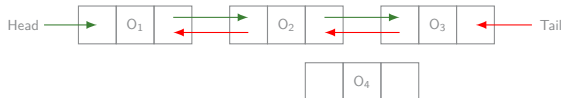


`insertAt(2)`

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt
- ▶ indexOf

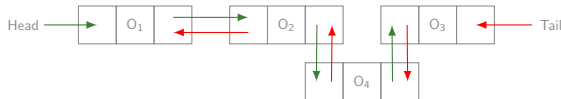


`insertAt(2)`

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt
- ▶ indexOf

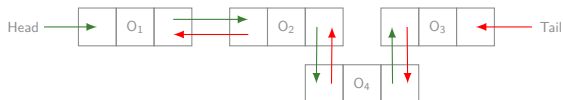


insertAt(2)

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt $O(n)$
- ▶ indexOf



insertAt(2)

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt $O(n)$
- ▶ indexOf



indexOf(O_2)



Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt $O(n)$
- ▶ indexOf

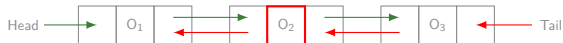


`indexOf(O_2)`

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt $O(n)$
- ▶ indexOf



`indexOf(O2)`

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt $O(n)$
- ▶ indexOf $O(n)$



indexOf(O_2)

Linked List

Operations

- ▶ add/remove first/last $O(1)$
- ▶ get/insertAt $O(n)$
- ▶ indexOf $O(n)$



Outline

1 Introductory Data Structures

Array
Linked List
Hash Table
Tree

2 Java Collection API & Map API

Java Collection API
Java Map API

3 Advanced Data Structures

Stuff...

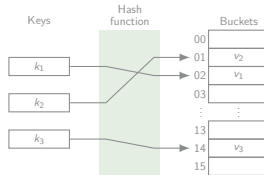


Hash Table

Definition

A dictionary in which keys are mapped to array positions by hash functions.⁵

- ▶ Hash functions: determinism, uniformity, defined range, data normalisation, non-invertible, perfect, . . .
- ▶ Collisions resolution: chaining, open addressing, . . .
- ▶ Example:

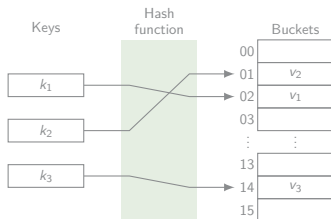


⁵ National Institute of Standards & Technology

Hash Table

Operations

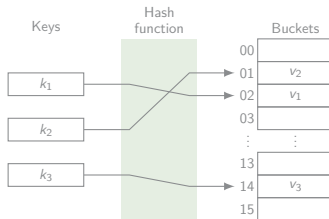
- put
- remove
- get



Hash Table

Operations

- put
- remove
- get

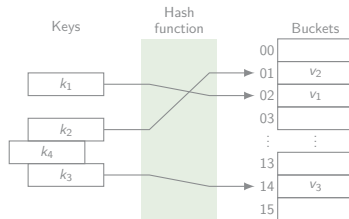


put(0₄)

Hash Table

Operations

- put
- remove
- get

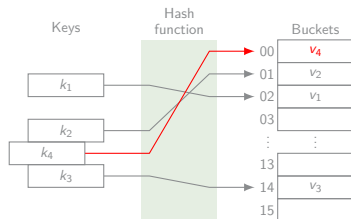


put(04)

Hash Table

Operations

- put
- remove
- get



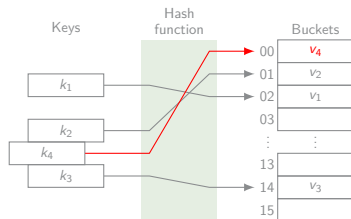
put(04)

Hash Table

Operations

- put
- remove
- get

$O(1)$ / $O(n)$



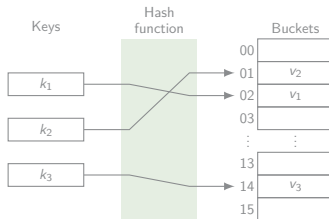
put(0_4)

Hash Table

Operations

- put
- remove
- get

$O(1)$ / $O(n)$



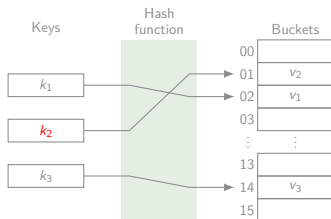
`remove(02)`

Hash Table

Operations

- put
- remove
- get

$O(1)$ / $O(n)$



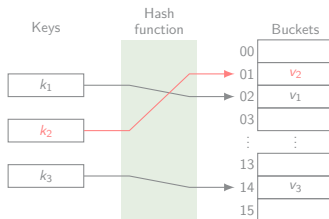
`remove(02)`

Hash Table

Operations

- put
- remove
- get

$O(1)$ / $O(n)$

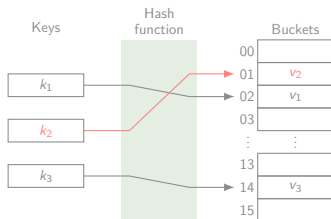


`remove(02)`

Hash Table

Operations

- put $O(1) / O(n)$
- remove $O(1) / O(n)$
- get

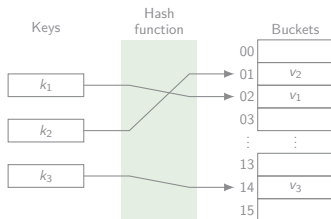


`remove(02)`

Hash Table

Operations

- put $O(1) / O(n)$
- remove $O(1) / O(n)$
- get

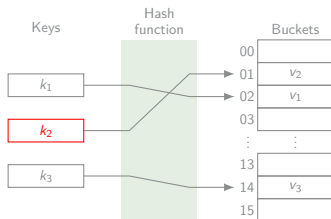


get(0_2)

Hash Table

Operations

- put $O(1) / O(n)$
- remove $O(1) / O(n)$
- get

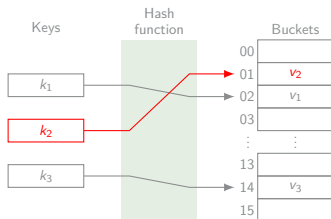


get(0₂)

Hash Table

Operations

- put $O(1) / O(n)$
- remove $O(1) / O(n)$
- get

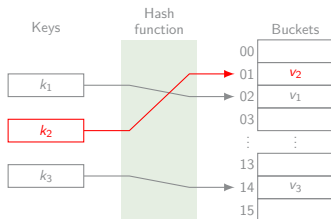


get(0_2)

Hash Table

Operations

- put $O(1) / O(n)$
- remove $O(1) / O(n)$
- get $O(1) / O(n)$



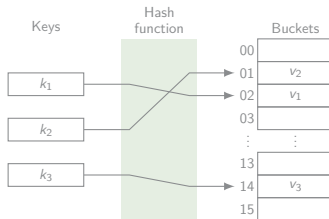
get(0_2)



Hash Table

Operations

- put $O(1) / O(n)$
- remove $O(1) / O(n)$
- get $O(1) / O(n)$



Outline

1 Introductory Data Structures

- Array
- Linked List
- Hash Table
- Tree
 - Heap
 - Red-Black Tree

2 Java Collection API & Map API

- Java Collection API
- Java Map API

3 Advanced Data Structures

- Stuff...



Tree

...



Binary Heap

...



Binary Min-Heap

...



Red-Black Tree

...



Red-Black Tree

...



Outline

- 1 Introductory Data Structures
 - Array
 - Linked List
 - Hash Table
 - Tree
- 2 Java Collection API & Map API
 - Java Collection API
 - Java Map API
- 3 Advanced Data Structures
 - Stuff...



Outline

- 1 Introductory Data Structures
 - Array
 - Linked List
 - Hash Table
 - Tree
- 2 Java Collection API & Map API
 - Java Collection API
 - Java Map API
- 3 Advanced Data Structures
 - Stuff...



Java Collection API

...



Java Collection API

...



Outline

- 1 Introductory Data Structures
 - Array
 - Linked List
 - Hash Table
 - Tree
- 2 Java Collection API & Map API
 - Java Collection API
 - Java Map API
- 3 Advanced Data Structures
 - Stuff...



Java Map API

...



Java Map API

...



Outline

- 1 Introductory Data Structures
 - Array
 - Linked List
 - Hash Table
 - Tree
- 2 Java Collection API & Map API
 - Java Collection API
 - Java Map API
- 3 Advanced Data Structures
 - Stuff...



Outline

- 1 Introductory Data Structures
 - Array
 - Linked List
 - Hash Table
 - Tree
- 2 Java Collection API & Map API
 - Java Collection API
 - Java Map API
- 3 Advanced Data Structures
 - Stuff...



Stuff...

...

