

# van Emde Boas Trees

光吉 健汰

北海道大学工学部 情報エレクトロニクス学科 情報理工学コース 4 年  
情報知識ネットワーク研究室

June 11, 2019

# Contents

- 1 自己紹介
- 2 van Emde Boas tree とは
- 3 binary-tree

# Contents

- 1 自己紹介
- 2 van Emde Boas tree とは
- 3 binary-tree

こんにちは

# Contents

- 1 自己紹介
- 2 van Emde Boas tree とは
- 3 binary-tree

# van Emde Boas tree とは (1/2)

van Emde Boas tree (以下 vEB 木) は動的集合を扱うデータ構造

## 動的集合

動的集合とは, 集合に対して後述の各種操作が行えるようなデータ構造

- 今回扱う要素は非負整数とする
- vEB 木が保持しうる要素の集合を全体集合  $U$  とし, その大きさを  $u$  とする
- vEB 木が現在保持している集合を  $V$  とし, その大きさを  $n$  とする

## van Emde Boas tree とは (2/2)

van Emde Boas tree (以下 vEB 木) は動的集合を扱うデータ構造

### 操作

**MEMBER**( $V, x$ )  $V$  に  $x$  が存在するかを返す

**MIN**( $V$ )  $V$  の要素の最小値を返す

**MAX**( $V$ )  $V$  の要素の最大値を返す

**SUCCESSOR**( $V, x$ )  $V$  の  $x$  より大きい最小の要素を返す

**PREDECESSOR**( $V, x$ )  $V$  の  $x$  より小さい最大の要素を返す

**INSERT**( $V, x$ )  $V$  に  $x$  を挿入する

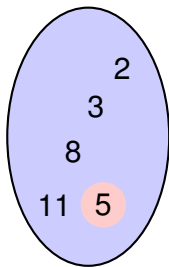
**DELETE**( $V, x$ )  $V$  から  $x$  を削除する

これらの操作が最悪時間計算量  $O(\log \log u)$  で実行可能

## 操作 MEMBER( $V, x$ )

- MEMBER( $V, x$ ) は,  $x$  が集合に存在するかを真偽値で返す.

Figure:  $V = \{2, 3, 5, 8, 11\}$



- MEMBER( $V, 5$ ) = TRUE
- MEMBER( $V, 6$ ) = FALSE

### 各関数の引数

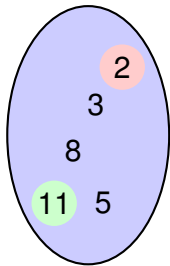
関数に渡す引数は要素として取りうる値のみとする



## 操作 $\text{MIN}(V)$ , $\text{MAX}(V)$

- $\text{MIN}(V)$  は, 集合の要素の最小値を返す.
- $\text{MAX}(V)$  は, 集合の要素の最大値を返す.

Figure:  $V = \{2, 3, 5, 8, 11\}$

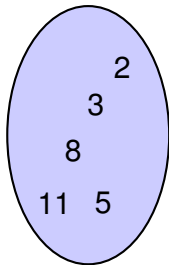


- $\text{MIN}(V) = 2$
- $\text{MAX}(V) = 11$

## 操作 $\text{SUCCESSOR}(V, x)$ , $\text{PREDECESSOR}(V, x)$

- $\text{SUCCESSOR}(V, x)$  は, 集合の要素から  $x$  より大きい最小の値を返す.
- $\text{PREDECESSOR}(V, x)$  は, 集合の要素から  $x$  より小さい最大の値を返す.

Figure:  $V = \{2, 3, 5, 8, 11\}$



- $\text{SUCCESSOR}(V, 2) = 3$
- $\text{PREDECESSOR}(V, 7) = 5$

# 操作 INSERT( $V, x$ )

- INSERT( $V, x$ ) は, 集合に  $x$  を挿入する.

Figure:  $V = \{2, 3, 5, 8, 11\}$

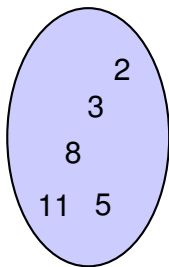
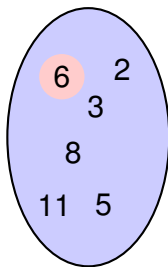


Figure:  $V = \{2, 3, 5, \mathbf{6}, 8, 11\}$



# 操作 DELETE( $V, x$ )

- DELETE( $V, x$ ) は, 集合  $V$  から  $x$  を削除する.

Figure:  $V = \{2, 3, 5, \mathbf{6}, 8, 11\}$

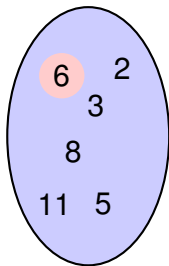
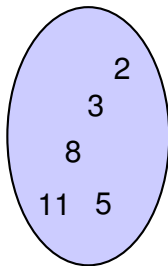


Figure:  $V = \{2, 3, 5, 8, 11\}$



# Contents

- 1 自己紹介
- 2 van Emde Boas tree とは
- 3 **binary-tree**

# 直接アドレス法 (1/2)

空間計算量  $O(u)$  で動的集合を保持する手段として、直接アドレス法を考える。

## 直接アドレス法

要素の値を配列の添字として利用し、データを保持するテクニック

今回考えているデータ構造では付属データは持たないので、各配列の要素には要素を保持しているかを bit で格納する。

Figure:  $V = \{2, 3, 5, 8, 11\}$

0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

## 直接アドレス法 (2/2)

空間計算量  $O(u)$  で動的集合を保持する手段として、直接アドレス法を考える。

$\text{member}(e)$ ,  $\text{insert}(e)$ ,  $\text{delete}(e)$  は,

Figure:  $V = \{2, 3, 5, 8, 11\}$

0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15