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

Contents

こんにちは

Contents

van Emde Boas tree とは

van Emde Boas tree (以下 vEB 木) は動的集合に対して,
空間計算量 $O(u)$ で後述する k 操作が可能なデータ構造

操作

member(e)	e が存在するかを返す
min()	要素の最小値を返す
max()	要素の最大値を返す
successor(e)	e より大きい最小の要素を返す
predecessor(e)	e より小さい最大の要素を返す
insert(e)	e を挿入する
delete(e)	e を削除する

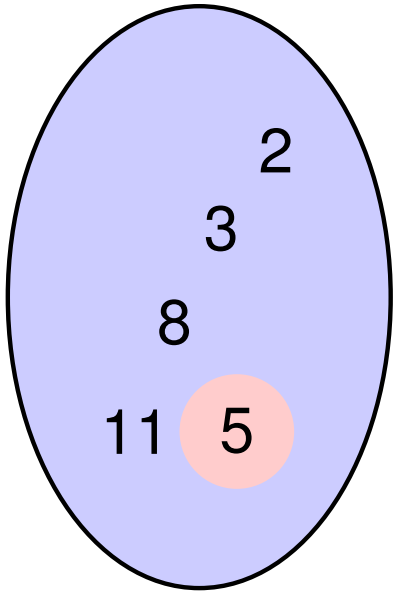
これらの操作が最悪時間計算量 $O(\log \log u)$ で実行可能

u := 保持しうる要素の集合 (全体集合) の大きさ

操作 member(e)

- member(e) は, e が集合に存在するかを真偽値で返す.

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

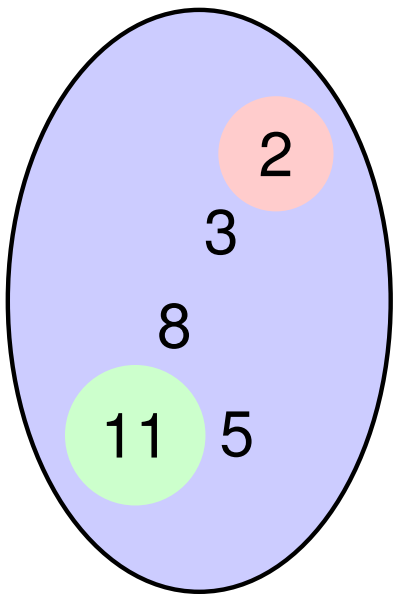


- member(5) = TRUE
- member(6) = FALSE

操作 min(), max()

- min(e) は, 集合の要素の最小値を返す.
- max(e) は, 集合の要素の最大値を返す.

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

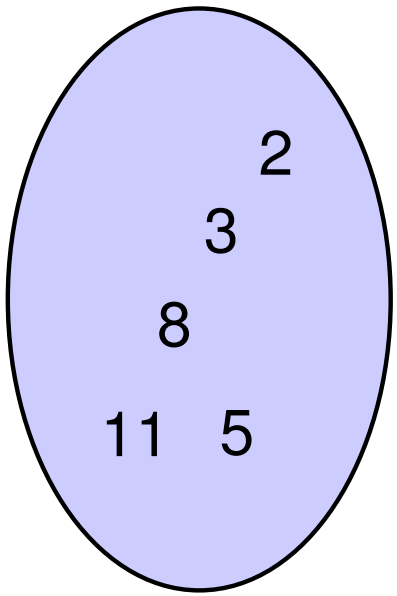


- min() = 2
- max() = 11

操作 successor(e), predecessor(e)

- successor(e) は, 集合の要素から e より大きい最小の値を返す.
- predecessor(e) は, 集合の要素から e より小さい最大の値を返す.

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



- successor(2) = 3
- predecessor(7) = 5

操作 insert(e)

- insert(e) は, 集合に e を挿入する.

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

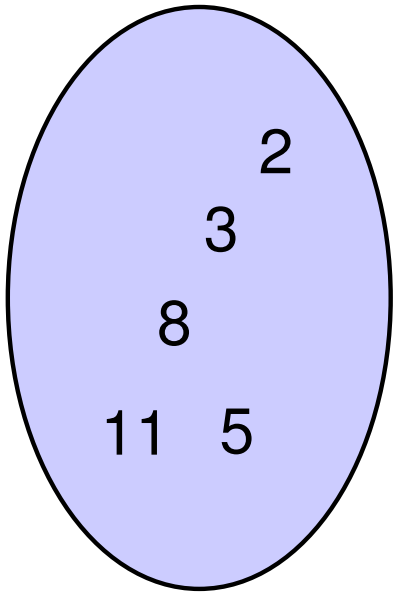
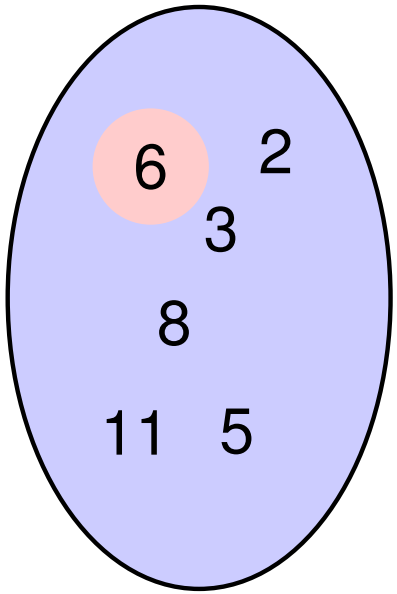


Figure: $S = \{2, 3, 5, 6, 8, 11\}$



操作 delete(e)

- ($delete$) は, 集合から e を削除する.

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

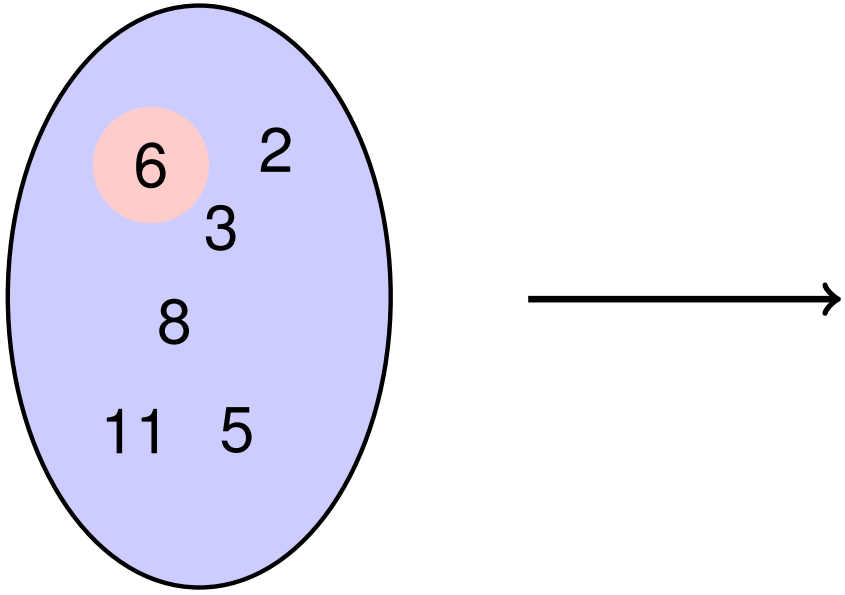
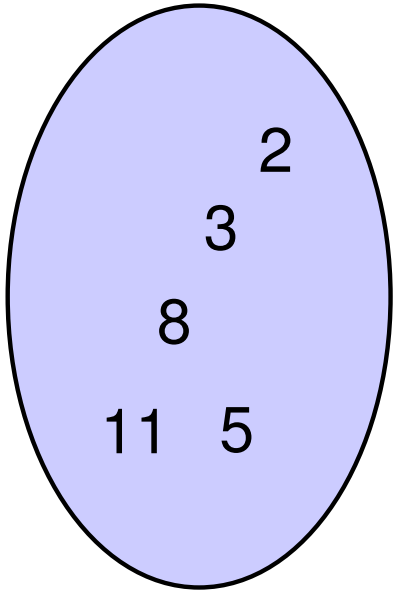


Figure: $S = \{2, 3, 5, 8, 11\}$



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

Contents

直接アドレス法 (1 / 2)

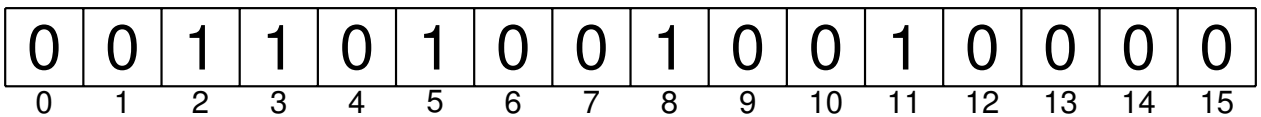
空間計算量 $O(u)$ で動的集合を保持する手段として, 直接アドレス法を考える.

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



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.

member(e)	ランダムアクセスが可能なので, $O(1)$
min()	配列の最小の要素から探索するので, $O(u)$
max()	min() と同様に, $O(u)$
successor(e)	配列の隣を順に探索するので, $O(u)$
predecessor(e)	successor(e) と同様に $O(u)$
insert(e)	e の配列の要素の bit を立てるだけなので $O(1)$
delete(e)	delete(e) と同様に $O(1)$

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