

## 3. スタック, キュー, 木

- スタックとは、後に入れたデータを先に取り出すデータ構造である。この性質を LIFO (last-in first-out) と呼ぶ。次の操作を行うことができる。
  - push( $x$ ): スタックの一番上に  $x$  を挿入する。
  - pop: スタックの一番上の要素を返し、それをスタックから取り除く。
- キュー (待ち行列) とは、先に入れたデータを先に取り出すデータ構造である。この性質を FIFO (first-in first-out) と呼ぶ。次の操作を行うことができる。
  - enqueue( $x$ ): キューの最後尾に  $x$  を挿入する。
  - dequeue: キューの先頭の要素を返し、それをキューから取り除く。
- 閉路を持たない連結な無向グラフを木という。以下では、木の各頂点 (節点) に値が定まっているものとする。
- 特別な頂点が 1 つ定まっている木を根付き木といい、この特別な頂点を根という。隣接する 2 頂点  $P, Q$  に対し、 $P$  の方が根に近いとき、 $P$  を  $Q$  の親、 $Q$  を  $P$  の子という。また、頂点  $Q$  から根までの道の間に頂点  $P$  があるとき、 $P$  を  $Q$  の先祖、 $Q$  を  $P$  の子孫という。子を持たない頂点を葉という。頂点から根までの道の長さをその頂点の深さという。頂点の深さの最大値を根付き木の高さ<sup>\*1</sup>という。
- 各頂点の子の数が  $n$  以下であるような根付き木を  $n$  分木という。
- 2 分木においては、子が左側か右側かを常に指定する。
- 2 分木  $T$  において、どの 2 頂点の値も比較可能であり、簡単のため、どの 2 頂点も同じ値を持たないものとする。 $T$  が 2 分探索木であるとは、各頂点の値が、それより左にある子孫の値より大きく、それより右にある子孫の値より小さいことである。次の操作を行うことができる。
  - min:  $T$  に含まれる値の最小値を求める。
  - search( $x$ ):  $x$  が  $T$  に含まれるかどうか判定する。
  - insert( $x$ ):  $x$  を  $T$  に追加する。
  - delete( $x$ ):  $x$  を  $T$  から取り除く。

最小値を求めるには最も左にある葉を探せばよい。

$x$  が  $T$  に含まれるか判定するには、まず  $x$  を根と比較する。一致すれば  $x$  は  $T$  に含まれる。 $x$  の方が小さいとき左の子について、 $x$  の方が大きいとき右の子について同じ操作を繰り返す。葉に到達しても一致しないならば、 $x$  は  $T$  に含まれない。

---

\*1 頂点の深さの最大値に 1 加えたものを高さと呼ぶこともある。

データを追加する際には正しい位置を同様に探し、葉を付け加えればよい。

データの削除は次のように行われる。まず、削除する頂点が葉である場合は、それを取り除けばよい。削除する頂点が1個の子を持つ場合、上下の頂点をつなげばよい。削除する頂点が2個の子を持つ場合、左側の子孫のうち値が最大のもの（または右側の子孫のうち値が最小のもの）を削除する頂点の位置に移し、空いた頂点の上下の頂点をつなげばよい。

一般に、2分探索木はデータの探索が高速に可能である（平均で  $O(\log n)$  回の比較でよい）。しかし、データの追加・削除の順序によっては木の高さが大きくなり、効率が非常に悪くなる。

### 問題

- 3-1. 空のスタックに対し、次の操作を行ったときの過程と最終的に得られるスタックを図示せよ。

push( $c$ ) → push( $b$ ) → push( $e$ ) → push( $f$ ) → pop →  
push( $d$ ) → pop → push( $a$ ) → pop → pop

- 3-2. 空のキューに対し、次の操作を行ったときの過程と最終的に得られるキューを図示せよ。

enqueue( $c$ ) → enqueue( $b$ ) → enqueue( $e$ ) → enqueue( $f$ ) → dequeue →  
enqueue( $d$ ) → dequeue → enqueue( $a$ ) → dequeue → dequeue

- 3-3. 2分木  $T$  が頂点を  $m$  個持つとする。  $T$  の高さの最大値と最小値を求めよ。

- 3-4. 要素 17, 32, 5, 8, 7, 56, 88 を持つ 2分探索木の中で、高さが最も小さくなるものと、高さが最も大きくなるものをそれぞれ1つずつ図示せよ。

- 3-5. 空の2分探索木に対し、次の操作を行ったときの過程と最終的に得られる木を図示せよ。

insert(57) → insert(16) → insert(62) → insert(96) →  
insert(80) → delete(57) → delete(62)

- 3-6. 空の2分探索木に対し、次の操作を行ったときの過程と最終的に得られる木を図示せよ。

insert(86) → insert(46) → insert(37) → insert(98) →  
insert(65) → delete(86) → delete(65)