

Chapter10

石田研 M2 松山祐輔

2017/7/14

1 10章 ヒープ

- Yusuke Matsuyama
- 7/14@book-seminar2017
- via jupyter RISE

2 おきもち

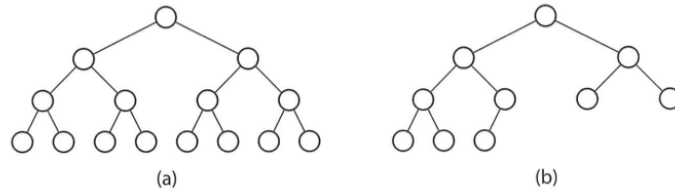
1. 優先度付きキューを実装したい
2. 二分探索木を使えば優先度付きキューは作れる
 - しかし非常に難しい
3. 二分ヒープを使えば比較的簡単
 - つくろう!

3 アウトライン

1. 二分ヒープの説明
 1. 完全二分木
 2. 二分ヒープ
 1. max-ヒープ
2. max-ヒープの実装
3. 優先度付きキューの実装

4 完全二分木

- すべての葉が同じ深さを持ち、全ての内部接点の子の数が2であるような二分木 (a)
- 深さの差が max1 でも、ちゃんと左側から埋まってるような木も (おおよそ (?)) 二分木 (b)



5 二分ヒープ

完全二分木の各ノードに割り当てられたキーが、配列の各要素に対応したデータ構造

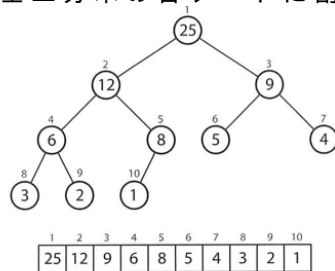


図10.2: 二分ヒープ

6 index の計算

添字 i が与えられた時:

```
In [5]: import math

        L = 10 # length of heap

        # origin 1

        parent      = lambda i : math.floor(i/2)
        left_child   = lambda i : 2 * i
        right_child  = lambda i : 2 * i + 1
        parent(5), left_child(5), right_child(5)
```

Out[5]: (2, 10, 11)

7 max-ヒープ条件

- 節点のキーがその親のキー以下であること (逆: min ヒープ条件)
- 制約があるのは親子間のみ (兄弟間にはない)

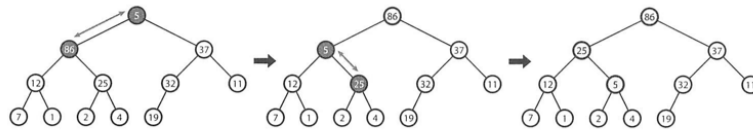


図 10.3: ヒープ化

maxheapfi.png

7.1 max-ヒープ

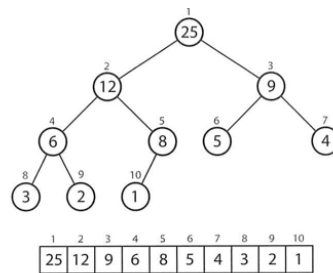


図 10.2: 二分ヒープ

- max ヒープ条件を満たす二分ヒープ

8 ヒープへの要素追加

8.1 maxheapfi(A,i)

$A[i]$ を max ヒープ条件を満たすまで葉に向かって下降

1. if $\max(\text{左子のキー}, \text{右子のキー}) > \text{自分のキー}$ then そいつと自分を入れ替える
2. 1. を繰り返す

8.2 max-ヒープの構築

子を持つ節点の中で添字が最大の節点 s から逆順に $\text{maxheapfi}(A,s)$ をすればいい

9 ヒープ構築の計算量

要素数を H とする。

9.1 maxheapfi :

$O(\log \text{ 木の高さ})$

9.2 ヒープの構築:

- 高さ 1 の部分木 $H/2$ 個に対して maxheapfi
- 高さ 2 の部分木 $H/4$ 個に対して maxheapfi
- ... 高さ $\log H$ の部分木 1 つ に対して maxheapfi

$$H * \sum_{k=1}^{\log H} \frac{k}{2^k} = O(H)$$

10 問題 1(15 分) 最大ヒープの実装 (ALDS1_9_B)

与えられた配列から、max-ヒープを実装してください

10.1 入力

- 最初の行:配列サイズ H
- H 個の配列要素

10

4 1 3 2 16 9 10 14 8 7

10.2 出力例

max-ヒープの節点の値を $1 \dots H$ に向かって順番に空白区切で出力

16 14 10 8 7 9 3 2 4 1

10.3 制約

$$1 \leq H \leq 500000$$

$$-2,000,000,000 \leq \text{節点の値} \leq 2,000,000,000$$

11 優先度付きキュー

キーを持つ要素 の集合 ## insert(S, k): 集合 S に要素 k を挿入

11.1 extractMax(S):

最大のキーを持つ S の要素を S から削除し、その値を返す

12 要素の追加

1. ヒープの末尾に追加
2. ヒープ条件の確保

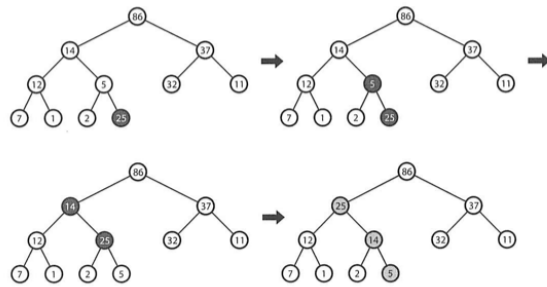


図 10.5: 要素の追加とヒープ化

- while(親より強い): 親と入れかわる

13 要素の取り出し

1. 根を記録
2. ヒープの一番末尾の要素を根に移動
3. ヒープサイズ $H-1$
4. 根から maxheapfi を実行

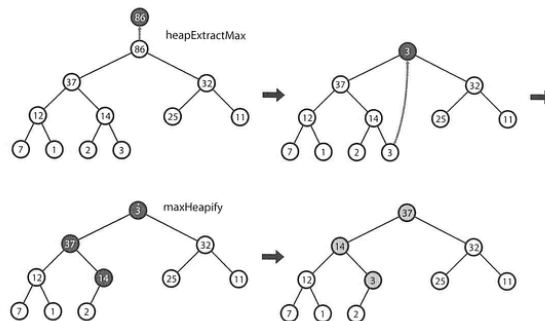


図 10.6: 最大要素の取得・削除

5. 記憶していた旧根の値を return

14 問題 2: 優先度付きキューの実装 (ALDS1_9_C)

14.1 入力:

優先度付きキューへの複数の命令 - insert k - extract - end

14.2 出力:

extract 命令ごとに、優先度付きキューから取り出される値を一行ごとに出力

14.3 制約

命令の数 $\leq 2,000,000$

$0 \leq k \leq 2,000,000,000$

- 入力

```
insert 8
```

```
insert 2
```

```
extract
insert 10
extract
insert 11
extract
extract
```

- 出力

```
8
10
11
2
```