

# Heap

## Supplemental Lecture

2010/4/14

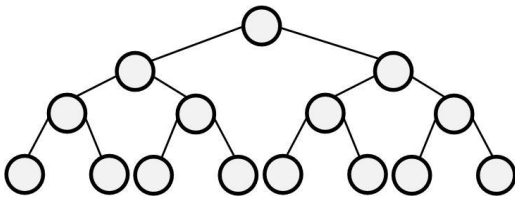
1

# Heapとは？

2010/4/14

2

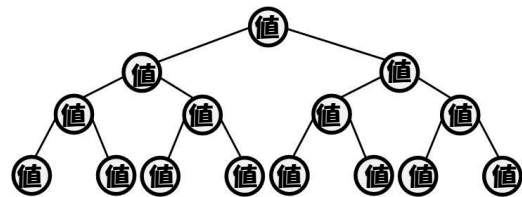
## ・2分木である



2010/4/14

3

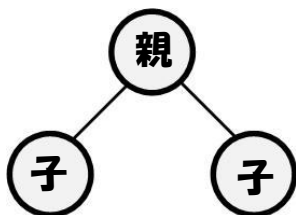
## ・各ノードは値を持つ



2010/4/14

4

## ・親の値は子より大きい（小さい）

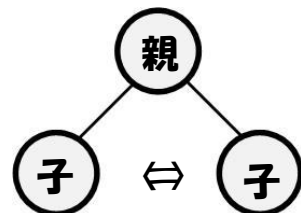


親  $\geq$  子  
(親  $\leq$  子)

2010/4/14

5

## ・兄弟の大きさは関係ない

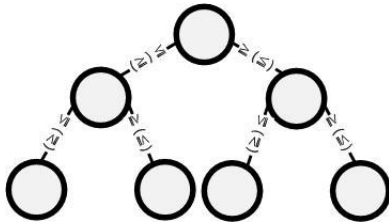


彼らの値の大小は関係ない

2010/4/14

6

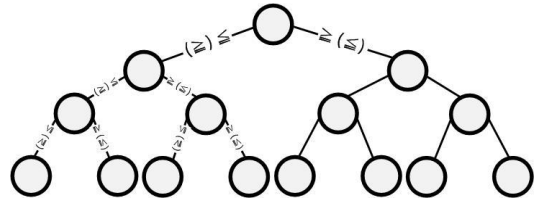
・部分木のペアでも同じ



2010/4/14

7

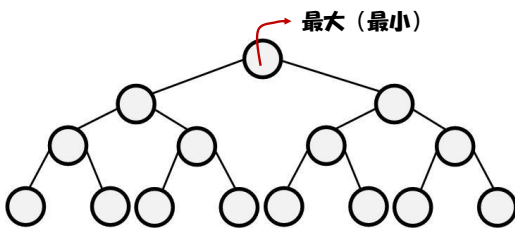
・全ての部分木に対しても同じ



2010/4/14

8

・すなわち、先頭は最も大きい（小さい）値になる



2010/4/14

9

- ・親  $\geq$  子  
–Maximum Heap
- ・親  $\leq$  子  
–Minimum Heap

2010/4/14

10

## Heapで出来る事

2010/4/14

11

- ・最大（最小）が先頭に来るので優先順位キューに使える
- ・これを利用してソートもできる

2010/4/14

12

## Heapを使うために

2010/4/14

13

- キューやソートのために根のデータをとったり、
- ヒープにデータを加える操作が必要
- そもそもどうやってヒープを作るの？

2010/4/14

14

- Up Heap と Down Heapを使うえばそれらを実現できる。

2010/4/14

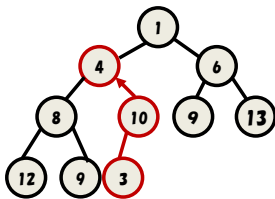
15

## Up Heap

2010/4/14

16

- Up Heapはある節または葉から、その値の正しい位置を、上へ探していくやり方。



2010/4/14

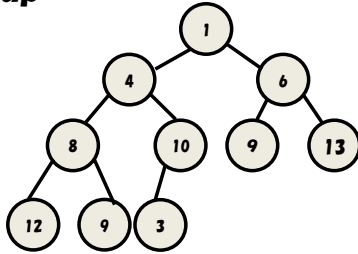
17

1. 現在地の値を  $v$  に保存
2.  $v$  と現在地の親を比べる、
  - もし、 $v \geq (\leq)$  現在地の親の値なら3へ
  - 違うなら、4へ。
3. 現在地の親の値を現在地の値にする。現在地を親の所に移動させる。→2へ戻る
4. 現在地に  $v$  を入れる。終了

2010/4/14

18

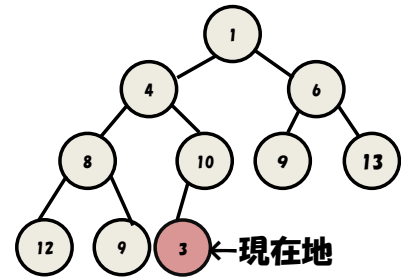
• (例) Minimum HeapでUp Heap



2010/4/14

19

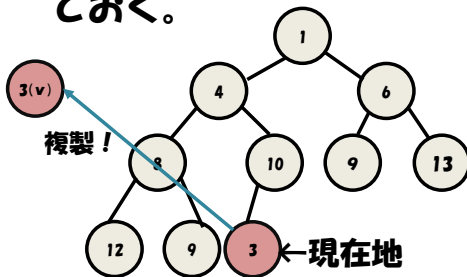
• 3から始める。



2010/4/14

20

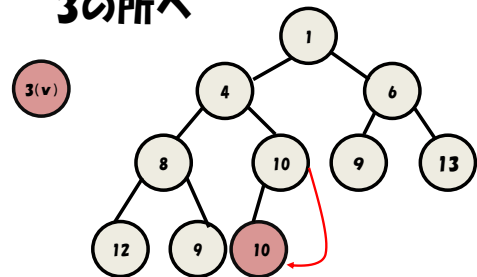
1. まず3をどこかに保存しておく。



2010/4/14

21

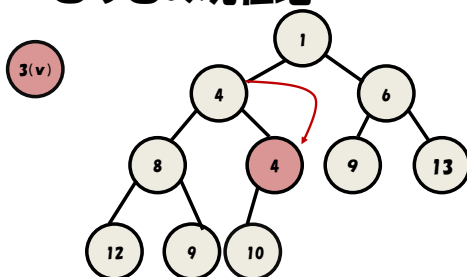
2. 3. 10は3より大きいので3の所へ



2010/4/14

22

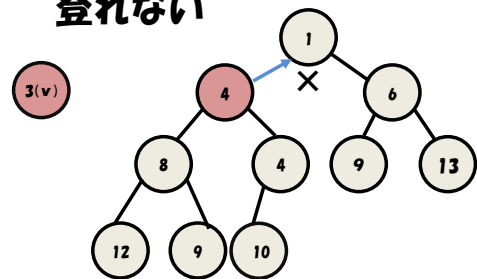
2. 3. 4は3より大きいのでさっきの現在地へ



2010/4/14

23

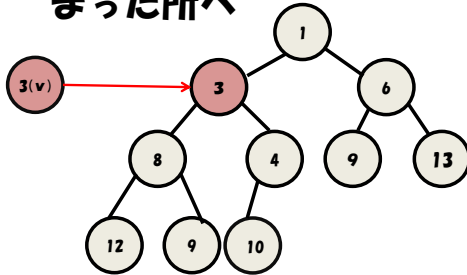
2. 1は3より小さいのでもう登れない



2010/4/14

24

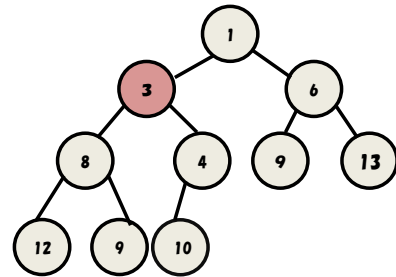
#### 4. ここでストップ。3は止まった所へ



2010/4/14

25

#### ・お終い



2010/4/14

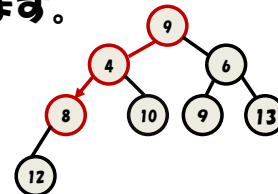
26

### Down Heap

2010/4/14

27

#### ・Down Heapはある節または葉から、その値の正しい位置をヒープを下りながら探します。



2010/4/14

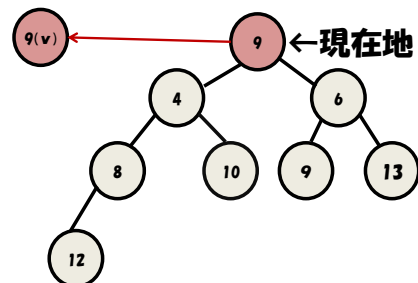
28

1. 現在地の値を  $v$  に保存
2.  $v$  と現在地の子を比べる、
  - もし、 $v \geq (\leq)$  現在地の子の値なら3へ
  - 違うなら、4へ。
3. 現在地の子の値を現在地の値にする。現在地を子の所に移動させる。→2へ戻る
4. 現在地に  $v$  を入れる。終了

2010/4/14

29

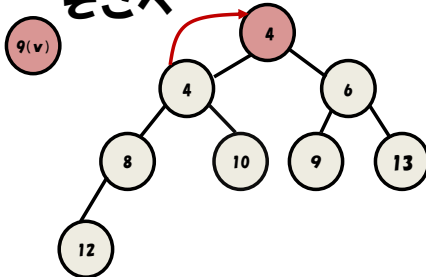
#### 1. (例)9から始める/9を保存



2010/4/14

30

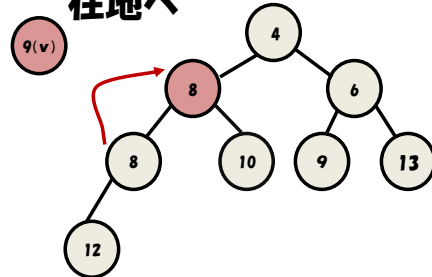
2. 3. 4は9より小さいから  
そこへ



2010/4/14

31

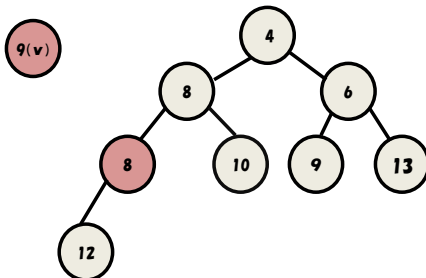
2. 3. 8の方が小さいから現  
在地へ



2010/4/14

32

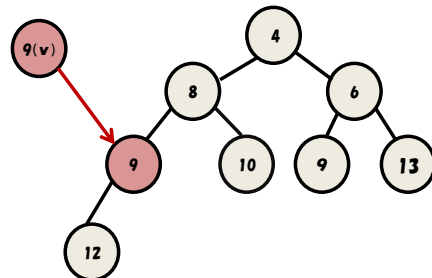
2. 9の方が大きいから…



2010/4/14

33

4. vを現在地に入れて終了



2010/4/14

34

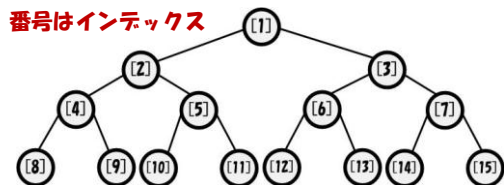
# Heapの実装ヒント

2010/4/14

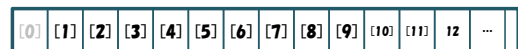
35

・ヒープは1次元配列である

番号はインデックス



配列



2010/4/14

36

- 子から親に行くには
  - [子のインデックス] / 2
- 親から子に行くには
  - 左  $2 * [\text{親のインデックス}]$
  - 右  $\text{左} + 1$