Tutorial 4

Agenda

- Ex 4.3-1
- P 4-1 (a, c, e)
- Ex 5.3-4
- Ex 6-5.2
- Ex 6-5.6
- P 6-1 (a)

4.3-1

We guess that $T(n) \le cn^2$ for some constant $c > 0$. We have

$$\begin{aligned} T(n) &= T(n-1) + n \\ &\le c(n-1)^2 + n \\ &= cn^2 - 2cn + c + n \\ &= cn^2 + c(1-2n) + n . \end{aligned}$$

This last quantity is less than or equal to cn^2 if $c(1-2n) + n \le 0$ or, equivalently, $c \ge n/(2n-1)$. This last condition holds for all $n \ge 1$ and $c \ge 1$.

For the boundary condition, we set $T(1) = 1$, and so $T(1) = 1 \le c \cdot 1^2$. Thus, we can choose $n_0 = 1$ and $c = 1$.

P4-1 (a)

$$T(n) = 2\,T(n/2) + n^3$$

a = 2, b = 2, f(n) = n^3

Using the master theorm,

$$n^{(\log_a b)} = n^{(\log_2 2)} = n$$

In order to make f(n) = n^3, we need to choose ϵ=2

Using case 3 of master theorm, we get

$$T(n) = \theta(n^3)$$

P4-1 (c)

$$T(n) = 16\,T(n/4) + n^2$$

a = 16, b = 4, f(n) = n^2

Using the master theorm, $n^{(\log_a b)} = n^{(\log_4 16)} = n^2$

Since f(n) also equals n^2, ϵ=0. We can apply case 2 of master theorm:

$T(n) = \theta(n^2 \lg(n))$

P4-1 (e)

$$T(n) = 7\,T(n/2) + n^2$$

$a = 7$, $b = 2$, $f(n) = n^2$

Using master theorm $n^{(\log_a b)} = n^{(\log_2 7)}$

Since $f(n) = n^2$, and $n^{(\log_2 7)}$ is between n^2 and n^3, thus we can use case 1 of master theorm:

$T(n) = \theta(n^{\lg 7})$

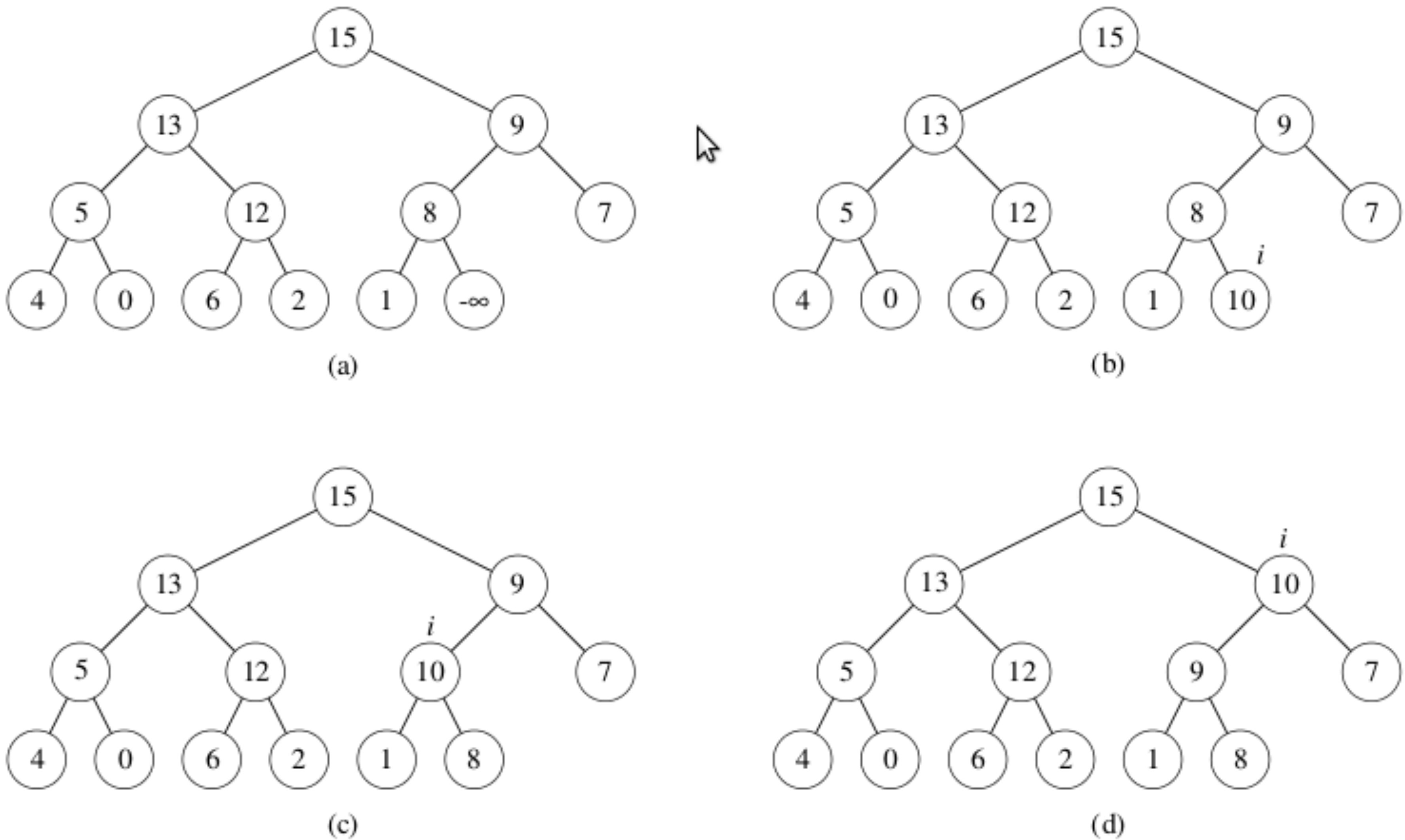
Ex 5.3-4

offset is random number between 1 and n. The algorithm shifts the value at element i by the offset. If the offset is greater than n, then a cyclic rotation is performed.

Since the value of offset is between 1 and n, each value has a probability of 1/n.

The procedure does not produce a uniform random permutation, since it can produce n different permutations.

Ex 6.5-2


15
13
9
5
12
8
7
4
0
6
2
1
-∞
(a)
15
13
9
5
12
8
7
i
4
0
6
2
1
10
(b)
15
13
9
5
12
i
10
7
4
0
6
2
1
8
(c)
15
i
13
10
5
12
9
7
4
0
6
2
1
8
(d)

Ex 6.5-6

HEAP-INCREASE-KEY(A, i, key)

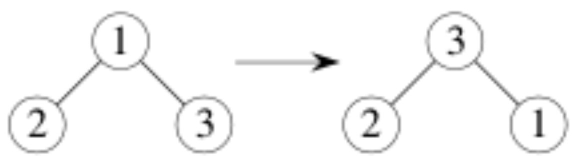
if $key < A[i]$
    error "new key is smaller than current key"
$A[i] = key$
while $i > 1$ and $A[\text{PARENT}(i)] < A[i]$
    $A[i] = A[\text{PARENT}(i)]$
    $i = \text{PARENT}(i)$
$A[i] = key$

P6-1 (a)

Input array A:

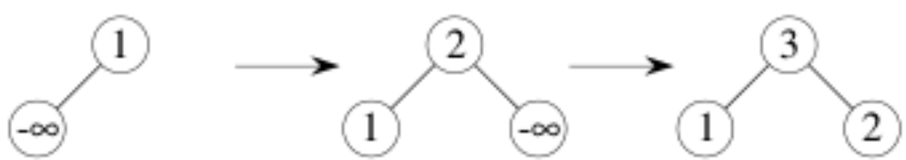
A			
	1	2	3

BUILD-MAX-HEAP(A):



A			
	3	2	1

BUILD-MAX-HEAP′(A):



A			
	3	1	2