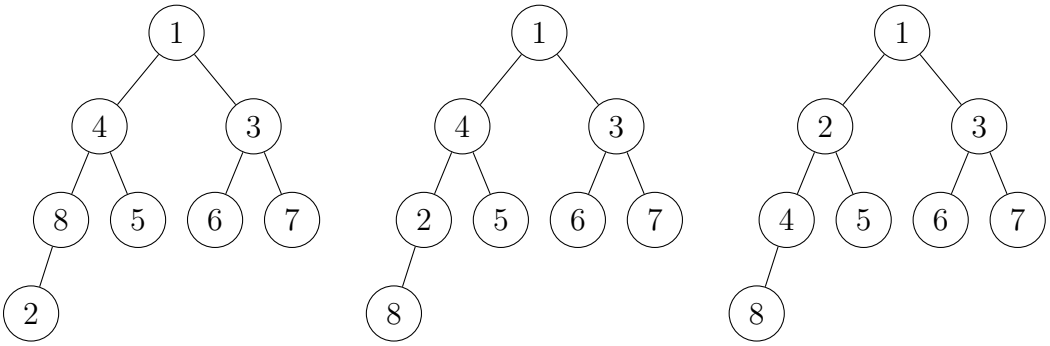
CS2468 Mid-Term Solution

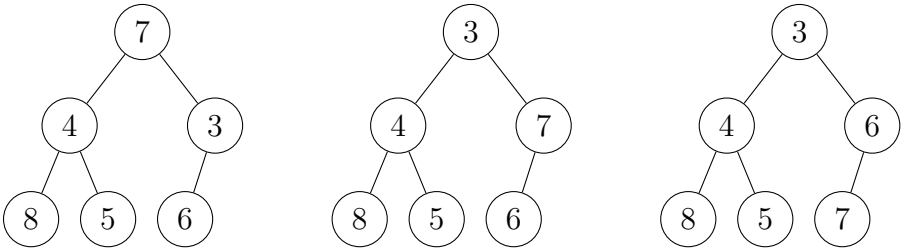
April 22, 2015

Question 1

(a)



(b)



(c)

A heap is a binary tree storing keys at its nodes and satisfying the following properties:

- Heap-Order: for every node v other than the root, key(v) < key(parent(v)).
- Complete Binary Tree: let h be the height of the heap for $i = 0, \ldots, h-1$, there are 2^i nodes of depth i at depth $h1$, the internal nodes are to the left of the external nodes.

(d)

Let h be the height of a heap storing n keys. Since there are 2^i keys at depth $i = 0, \ldots, h-1$ and at least one key at depth h. We have

$$n \geq 1 + 2 + 4 + \cdots + 2^{h-1} + 1 = 2^h.$$

Thus, $n \geq 2^h$, i.e., $h \leq \log n$.

Question 2

The parent of $a[i]$ if $a[i/2]$. The left child is $a[2i]$ and right child is $a[2i+1]$.

-	1	4	3	8	5	6	7
-	7	4	3	8	5	6	
-	3	4	7	8	5	6	
-	3	4	6	8	5	7	

Question 3

(a)

Inorder: 8 4 5 1 6 3 7.
Postorder: 8 5 4 6 7 3 1.
Preorder: 1 4 8 5 3 6 7.

(b)

```
function inOrder(v)
    if hasLeft(v) then
        inOrder(left(v))
    visit(v)
    if hasRight(v) then
        inOrder(right(v))
end function

function postOrder(v)
    if hasLeft(v) then
        inOrder(left(v))
    if hasRight(v) then
        inOrder(right(v))
    visit(v)
end function
```

Question 4

(a)

function countInternal(v)
  if v is null **then**
    return 0
  if not hasLeft(v) and not hasRight(v) **then**
    return 0
  else
    return 1 + countInternal(left(v)) + countInternal(right(v))
end function

(b)

The left child of v is $v.left$. The right child of v is $v.right$. The number associated with v is $v.number$.

function countEven(v)
  if v is null **then**
    return 0
  if $v.number$ is even **then**
    return 1 + countEven($v.left$) + countEven($v.right$)
  else
    return countEven($v.left$) + countEven($v.right$)
end function

Question 5

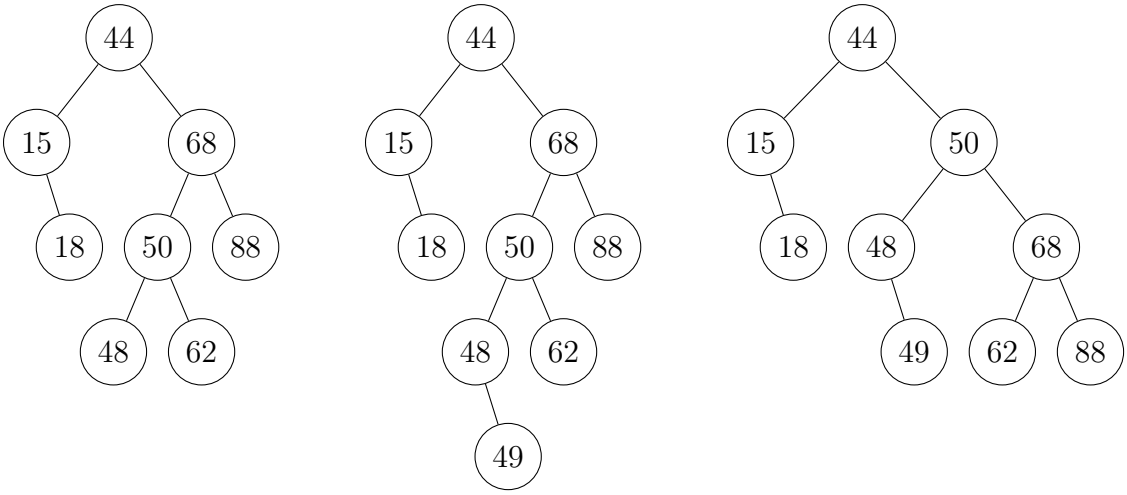
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(a)				20		73		41	8	25	44	59	42	12		32	
(b)				20		73		41	$8 \to 35 \to 59 \to 42$		44		12			32	

Question 6

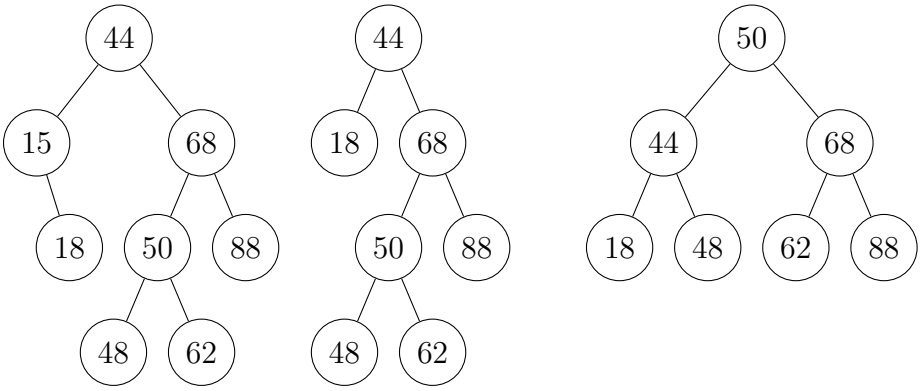
(a)

An AVL tree is a binary search tree such that for every internal node v, the heights of the children of v can differ by at most 1. The height of AVL tree consisting n elements is $O(\log n)$ and the running time of inserting is $O(\log n)$.

(b)



(c)



(c)

$O(\log n)$.

Question 7

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	1	1	0
3	0	0	0	0	0	0
4	0	0	0	0	1	1
5	0	0	0	0	0	0
6	0	0	1	0	0	0

1	$\rightarrow 2$
2	$\rightarrow 3 \rightarrow 4 \rightarrow 5$
3	
4	$\rightarrow 5 \rightarrow 6$
5	
6	$\rightarrow 3$

Question 8

a	b	c	d	e	f	g	h
0	∞	∞	∞	∞	∞	∞	∞
	∞	10/a	∞	∞	∞	1/a	6/a
	∞	10/a	∞	∞	6/g		3/g
	∞	10/a	5/h	∞	6/g		
	∞	8/d		7/d	6/g		
	∞	8/d		7/d			
	∞	8/d					
	11/c						

Path: $a \to g \to h \to d \to c \to b$.