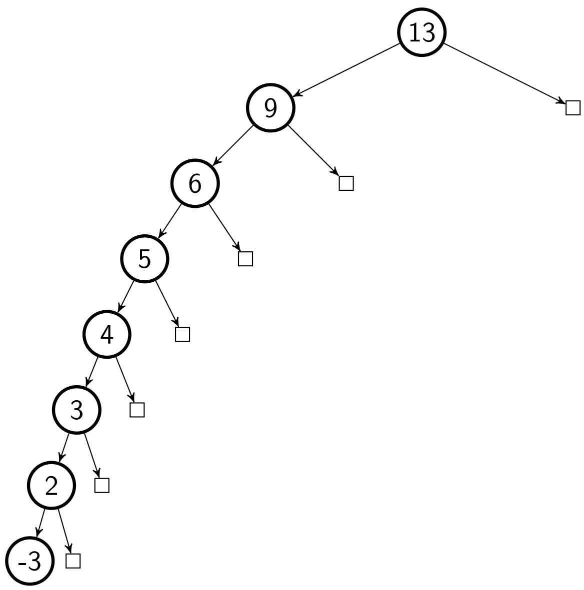
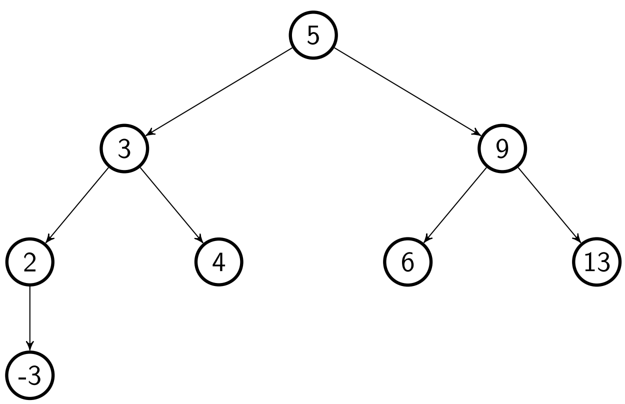
Q1.（10+10=20 分）

1. 对于关键字集合 $\{3, 6, 9, 2, 4, 5, 13, -3\}$，其对应的二叉搜索树（定义见 Topic_5-1.pdf，第 6 页）的最大树高 H_{max} 为多少？最小树高 H_{min} 为多少？并分别画出一个树高为 H_{max} 和一个树高为 H_{min} 的二叉搜索树；

   解：$H_{max} = 7$（2'），$H_{min} = 3$（2'）



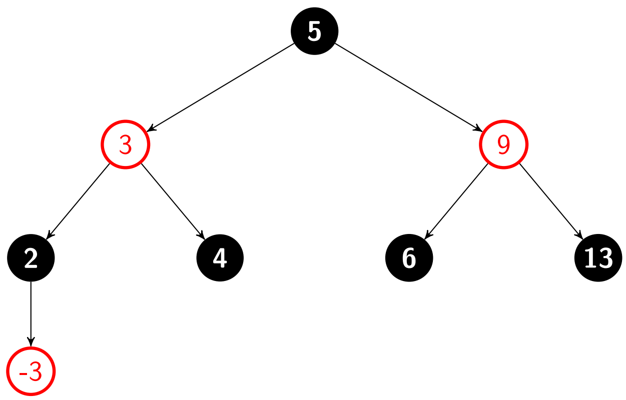
（3'，答案不唯一）



（3'，答案不唯一）

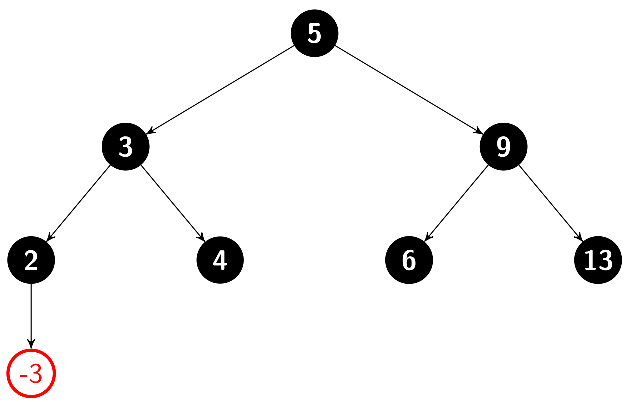
2. 对于关键字集合 $\{3, 6, 9, 2, 4, 5, 13, -3\}$，分别画出两个黑高不同的红黑树（定义见 Topic_5-1.pdf，第 27 页），并标明对应的黑高 H_{black}。

   解：$H_{black} = 2$（2'）



（3’，答案不唯一）

$H_{black} = 3$（2’）



（3’，答案不唯一）

Q2.（5+10+5+（3+3+4）=30 分）

1. 参考算法 InorderTreeWalk（见 Topic_5-1.pdf，第 7 页），并在其基础上修改，得到算法 InorderTreeWalk-Leaf（只输出叶子节点）；

Algorithm 1 InorderTreeWalk-Leaf(x)

if $x \neq$ NIL **then**
  InorderTreeWalk($left\,[x]$)
  if $left\,[x]$ == NIL *and* $right\,[x]$ == NIL **then**
    print x
  end if
  InorderTreeWalk($right\,[x]$)
end if

2. 考虑某个关键字 k，已知关键字 k 在算法 InorderTreeWalk-Leaf 的输出中出现且仅出现了 1 次。现在请设计算法 TreeSearch-Set(x, k)（给出伪代码），输入为二叉搜索树的根节点 $root$ 以及关键字 k，输出三个集合 S_{path}，S_{left}，S_{right}，其中 S_{path} 为算法 TreeSearch(x, k)（见 Topic_5-1.pdf，第 12 页）查找路径上的关键字集合，S_{left} 为查找路径左边的关键字集合，S_{right} 为查找路径右边的关键字集合（例如，对于下图 TreeSearch-Set$(root, 9)$ 的输出为 $S_{path} = \{8, 9, 10\}$，$S_{left} = \{0, 1, 6\}$,$S_{right} = \{12, 20, 52\}$）；

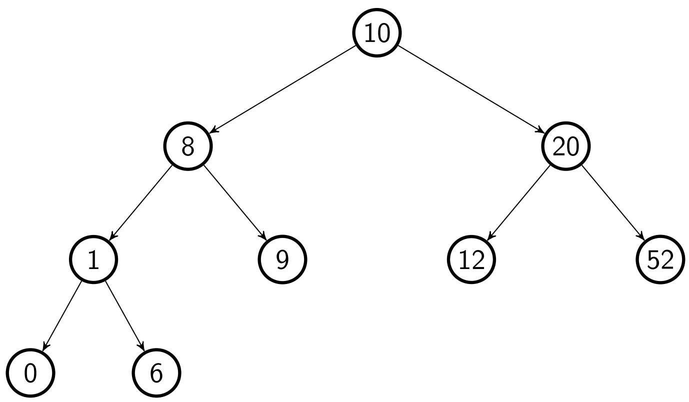


图 1: TreeSearch-Set 示例

Algorithm 2 Tree-Set(x)

$S = \emptyset$
if $x \neq$ NIL **then**
  $S = S \cup \{x\}$
  $S = S \cup$ Tree-Set$(left[x])$
  $S = S \cup$ Tree-Set$(right[x])$
end if
return S

Algorithm 3 TreeSearch-Set(x, k)

$S_{path} = \emptyset$, $S_{left} = \emptyset$, $S_{right} = \emptyset$
$tmp = x$
while $k \neq key\,[tmp]$ **do**
  $S_{path} = S_{path} \cup \{key\,[tmp]\}$
  if $k < key\,[tmp]$ **then**
    $S_{right} = S_{right} \cup$ Tree-Set$(right\,[tmp])$
    $tmp = left\,[tmp]$
  else
    $S_{left} = S_{left} \cup$ Tree-Set$(left\,[tmp])$
    $tmp = right\,[tmp]$
  end if
end while
$S_{path} = S_{path} \cup \{k\}$
print S_{path}, S_{left}, S_{right}

3. 对你设计的算法 TreeSearch(x, k) 的时间复杂度进行分析；

   解：算法 TreeSearch(x, k) 遍历二叉搜索树上的所有节点，复杂度为 $O(n)$。

4. 已知 S_{path}，S_{left}，S_{right} 为算法 TreeSearch-Set$(root, k)$ 的输出，请问以下命题是否正确：

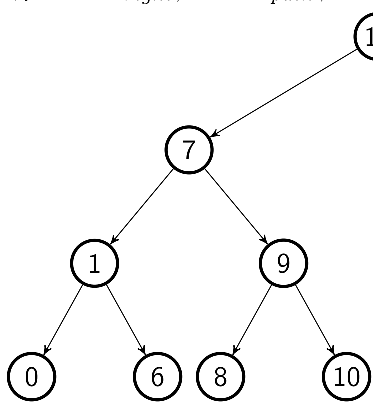
   - $\forall a \in S_{left}$，$\forall b \in S_{right}$，$a \leq b$.
   - $\forall a \in S_{left}$，$\forall b \in S_{right}$，$a \leq k \leq b$.
   - $\forall a \in S_{left}$，$\forall b \in S_{right}$，$\forall c \in S_{path}$，$a \leq c \leq b$.

   若正确，请简要证明；若不正确，请给出反例。

   解：

   - 正确。对于 $\forall a \in S_{left}$，$\forall b \in S_{right}$，$\exists c \in S_{path}$，在关键字 c 对应的结点为根的二叉搜索树中，a 在左子树上，b 在右子树上，所以有 $a \leq c$ 且 $c \leq b$，故有 $a \leq b$。

- 正确。$\forall a \in S_{left}$，$\exists c \in S_{path}$，在关键字 c 对应的结点为根的二叉搜索树中，a 在左子树上，k 在右子树上，所以有 $a \leq k$。同理，有 $k \leq b$。故有 $a \leq k \leq b$。
- 错误。对于下图所示的二叉搜索树,算法 TreeSearch-Set$(root, 8)$ 的输出为 $S_{path} = \{7, 8, 9, 11\}$,$S_{left} = \{0, 1, 6\}$,$S_{right} = \{10, 12, 20, 52\}$。有 $10 \in S_{right}$, $11 \in S_{path}$，10<11。



Q3.（15 分）

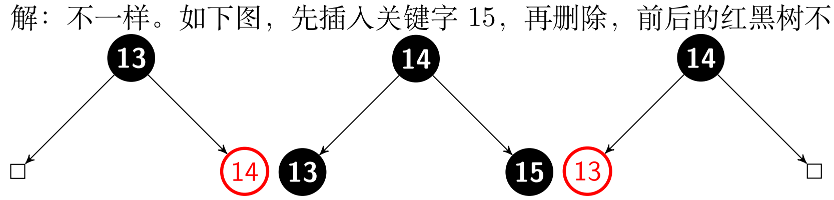
证明：任何一棵含 n 个结点的二叉搜索树可以通过 $O(n)$ 次旋转，转变为其他任何一棵含 n 个结点的二叉搜索树。

证：对于任意给定含 n 个结点的二叉搜索树 T_1 可以通过至多 $n-1$ 次左旋，得到一个所有右孩子都为空的树。又因得到的是有序链表。故可通过至多 $n-1$ 次右旋（即 T_2 转化为有序链表的逆操作），得到另一棵树 T_2。

Q4.（15 分）

假设用 RB-Insert 将一个结点 z 插入一棵红黑树，紧接着又用 RB-Delete 将它从树中删除。结果的红黑树与初始的红黑树是否一样？若一样，请证明；若不一样，请给出反例。

解：不一样。如下图，先插入关键字 15，再删除，前后的红黑树不同。



Q5.（20 分）

给出下图中的斐波那契堆调用 Extract-Min 后得到的斐波那契堆。

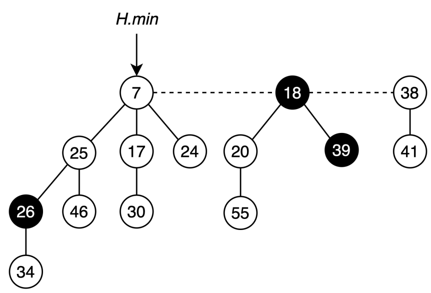
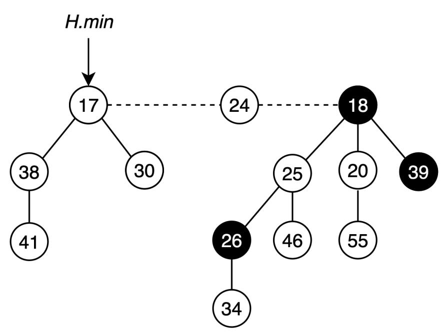

H.min
7
18
38
25
17
24
20
39
41
26
46
30
55
34

图 2: 斐波那契堆

解:


H.min
17
24
18
38
30
25
20
39
41
26
46
55
34