1. (a) Let $T(n)$ be the worst-case running time.

$$T(n) = T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + cn = 2T(\lceil \frac{n}{2} \rceil) + cn.$$

Suppose we have n elements, then after the sequence is halved k times, $n \le 2^k$. that is, the quick sort-tree has height $O(\log n)$. Therefore, if the size is n, then running time is $O(n\log n)$.

3. (a) Algorithm preorderNext (v):
Input: the current node v
Output: the next node in the preorder traversal of T

```
if T.isInternal(v) then
    return v.leftChild
else
    Node p = v.parent
    if v== v.leftChild then
        return p.rightChild
    else
        while v != v.leftChild   do
            v = p
            p = p.parent
        return p.rightChild
```

(b) inorderNext(v)
Input: the current node v
Output: the next node in the inorder traversal of T

```
if V.isInternal() then
      V = right child of V.
      while v is not external do
      V= leftchild of v
      return V.
else
    Node P = parent of V
    if V is left child of P then
        return P.
    else
        while V is not left child of p do
              V= P
              P = P.parent.
        return P.

(c). postorderNext(V)
    if V isInternal() then
        P= parent of V
        if V = right child of P then
              return P.
        else
              V = right child of P.
              While V is not external do.
                    V= left child of V
              return V.
    else
        P= parent of V.
        if V is left child of P then
              return right child of P.
        else
            return P.
```

The worst-case running time for all these algorithms are all $O(\log n)$ and n is the height of the tree T.

4. Base Case: if $n=0$,

$E(T)=I(T)=0$. $\therefore E(T)=I(T)+2n$. ✓.

Induction Hypothesis: $E(T)=I(T)+2n$ for all $n \geq 0$.

Let $n=k+1$ nodes.

Induction Steps: Let T be the full binary tree and T' be the tree after 2 external nodes are removed from T. Let P be an internal node of T.

Let d be the depth of a node of T.

$\therefore E(T)=E(T')-d(P)+d(P.left)+d(P.right)$.

$I(T)=I(T')+d(P)$

$E(T')=I(T')+2k$

$d(P.left)=d(P)+1$.

$\therefore E(T)=I(T')+2k-d(P)+2d(P.Left)$

$=I(T)-d(P)+2k-d(P)+2d(P.left)$

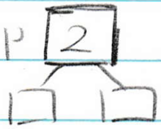
$=I(T)+2k\cancel{-2d(P)}\cancel{+2d(P)}+2$

$=I(T)+2(k+1)$

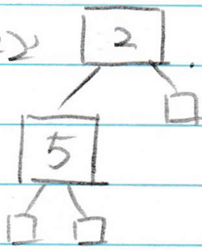
Therefore, for all $n \geq 0$, $E(T)=I(T)+2n$.

5. an alternate way of representing the elements is to take remainders of (every element) mod (n), then using bucketsort to sort the sequence of new-form element.

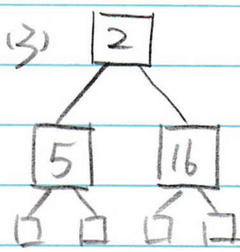
2. (1)



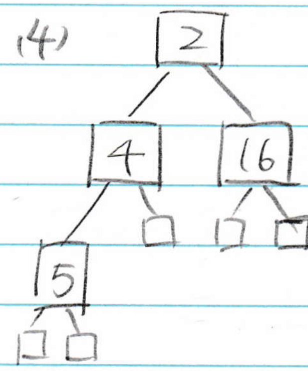
(2)



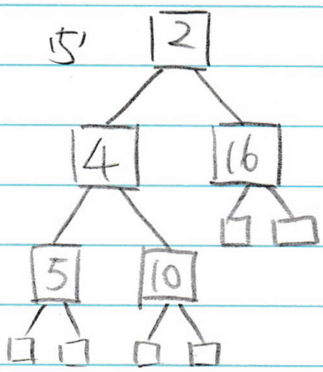
(3)



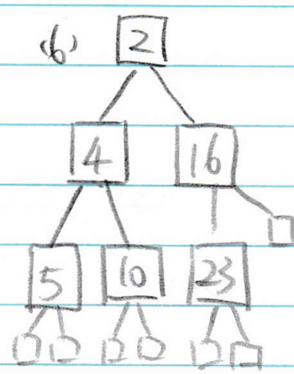
(4)



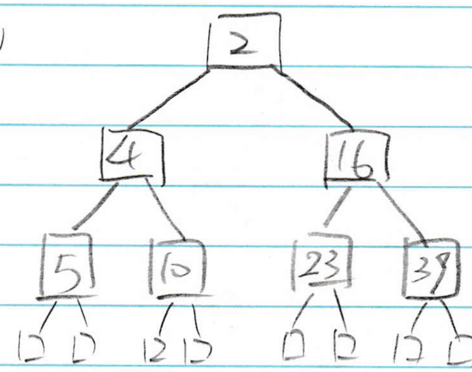
(5)



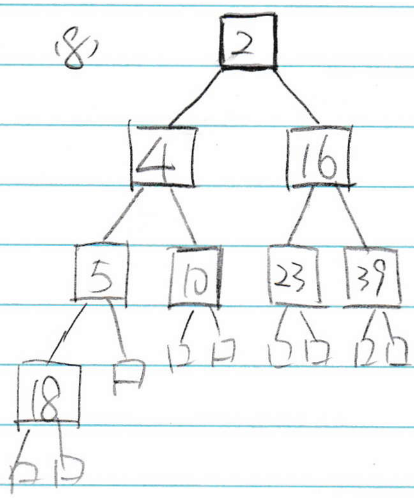
(6)



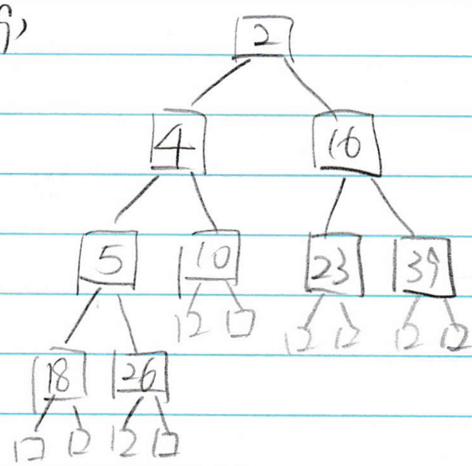
(7)



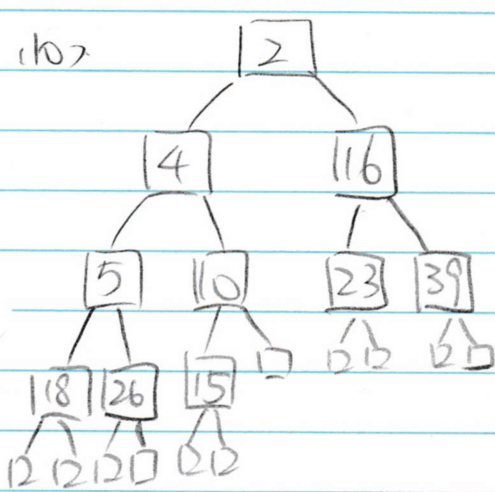
(8)



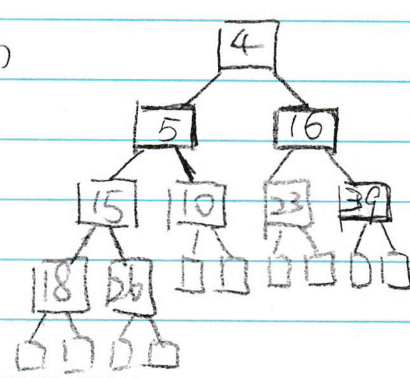
(9)



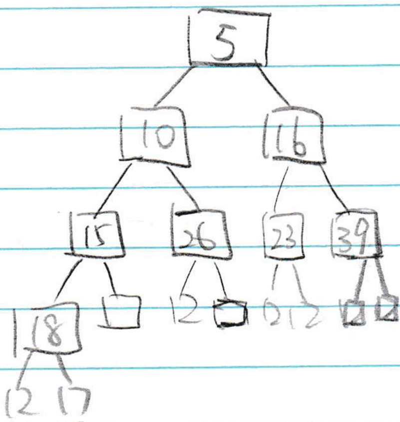
(10)



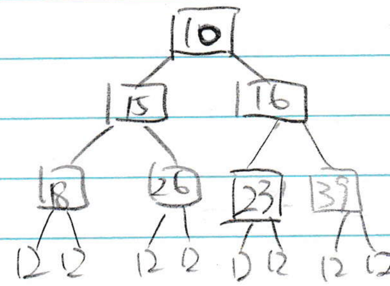
(11)



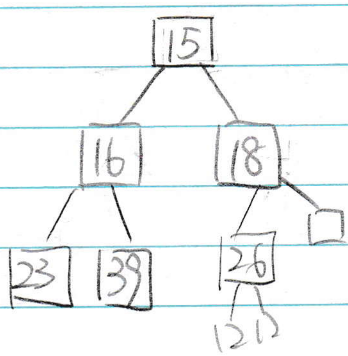
(12).



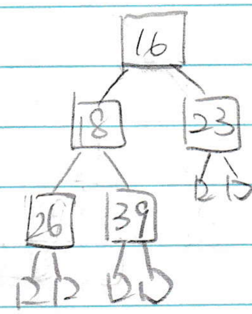
(13)



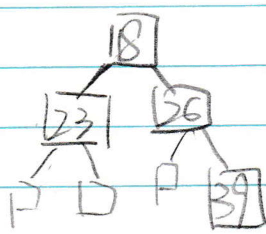
(14)



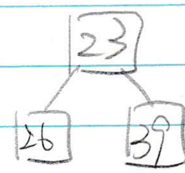
(15)



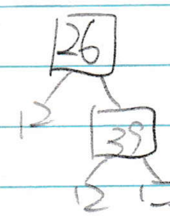
(16)



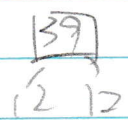
(17)



(18)



(19)



(20)