1. (1). B

(2). C

2. (1). Node tem = p

p = x

p.next = tem

(2). p = p.next

3. (1). (rear + 1) % length

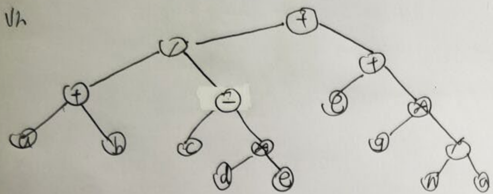
(2). length == 0;

length == m;

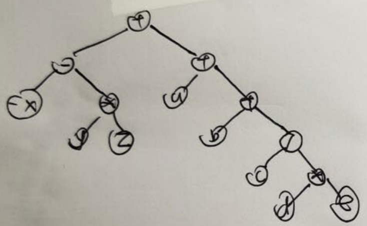
```
Enqueue(x){
 if( length <= m-1 ){
    for(int i = rear+1; i <= length; i++){
      a[i] = a[i-1]; }
    a[rear] = x;
    }
  else throws new BoundaryHitOfException();

  dlqueue(x){
    for(int i = rear+1; i<=length; i++){
       a[i] = a[i+1]; }
  }
```

南京大学
地址：南京市仙林大道163号 邮编：210046
网址：http://www.nju.edu.cn

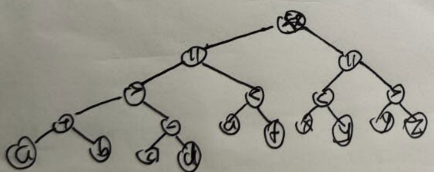
1、(1)



(2)、



(3)、



2. $n = n_0 + \sum_{i=1}^{m} n_i$

$n = B + 1$

$\sum_{i=0}^{m} n_i = \sum_{i=1}^{m} i n_i + 1$

$n_0 + n_1 + n_2 + \cdots + n_m = n_1 + 2n_2 + \cdots + m n_m + 1$

$n_0 = \sum_{i=1}^{m-1} i n_{i+1} + 1$

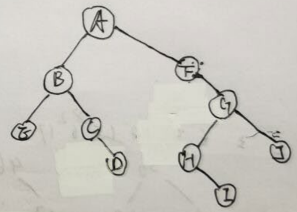
6、(1) A 、 A—B (2)、 A 、 A—B 3、 A

4、(1)

```
Stack<BinaryNode<T>*> s(10);
BinaryNode<T> *p=t;
int num;
while(true)
{ while(p!=NULL)
  { s.push(p); p=p->Left; }
  if(!s.IsEmpty())
  { p=s.pop();
    num+1
    p=p->Right;
  }        } ①
  else return num;
  }
}
```

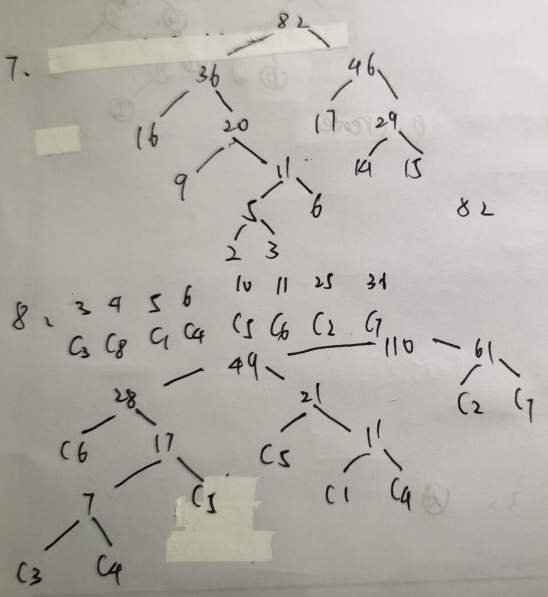
(2) 同上
在①处加 swap p.left 和 p.right

5、



6.
```
public Node createTree(String s) {
    Stack stack = new Stack();
    int nextchar = 0;
    while (s.charAt(nextchar) != null) {
        while (s.nextchar is Symbol()) nextchar++;   // 检查是否是运算符
        stack.push(new Node(s.charAt(nextchar)));
        int teamnum = s.findNotSymbol();
        if (teamnum == 1) {
            Node n = new Node(s.charAt(nextchar));
            n.left = new Node(stack.pop());
        else {
            Node n = new Node(s.charAt(nextchar));
            n.right = stak.pop();
            n.left = stak.pop();
            stack.push(n); }
    return stack.pop(); }
```

7.



110