一、13
1、顺序存储即在内存中以连续的方式存储数据，如数组；链式存储则以节点接节点的方式储存，内存地址可跳跃；
8 顺序存储优点在于快、占用空间小，缺点在于若删除数据会在内存中留空，难以扩容也无法插入数据；
链式存储优点在于易于增删改查，不必担心内存留空；缺点在于内存空间占用大，索引指定第几个数据需多次访问致慢。

0 2. ✗

3、

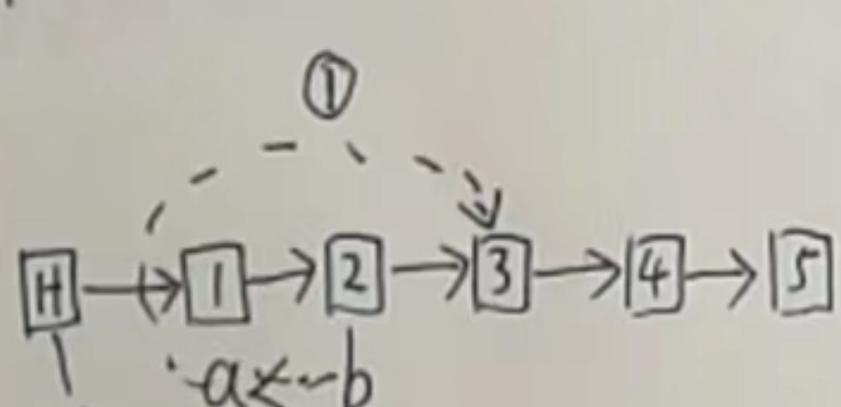
	Q	A✓					
B	∞	4 A→B	3 A→D→B	3 A→D→B ✓			
C	∞	∞	3 A→D→C				
D	∞	1 A→D ✓					
E	∞	∞	9 A→D→E	6 A→D→C→E	6 A→D→C→E ✓		

5

∴最短路径：
A→D→B
A→D→C
A→D
A→D→C→E

需要按照吉大的来画表，
最短路径需要指明路径长度

二、1、
```
class Node{
    int value;
    Node next;
}
void handle(Node head){ //姑且认为不会传null进来(头结点一般永远在)
    while(head.next != null){ //如果后面还有节点
        Node a = head.next; //下面第一个
        Node b = a.next; //下面第二个
        if(b == null){ //不足两个
            break;
        }
        a.next = b.next; //①
        b.next = a; //②
        head.next = b; //③
        head = a; //下一组
    }
}
```
15



8月11日 14:30 — 16:00.

2.
```
class Node{
    int value;
    Node left;
    Node right;
}
int target; //目标.X
List<Node> results = new ArrayList(); //结果我到的
void find(Node node, Node father){ // DFS
    if (node == null) return; //无节点.,略过
    if(node.value == target){ //符合条件
        results.add(father); //加结果
    }
    find(node.left, node); //找孩子
    find(node.right, node);
}
void FindFather(Node rod, int x){
    target = x;  //设目标
    results.clear(); //清空能可复用
    find(root, null);
    return results;
}
```

找所有父节点，而不是只是一个

3.
```
class Node{
    List<Node> childs;
}
List<Node> visited = new ArrayList(); //已访问
boolean check(Node node){
    visited.add(node);
    List remains = new ArrayList(node.childs);
    remains.removeAll(visited); //去除已访问
    if(remains.size() > 1){ //若有多余的则不唯一
        return false);
    }
for(Node c : node.childs){ //检查子节点
    if(!check(c)){
        return false;
    }
}
    return true;
}
```

整体是通过最小生成树算法变形得出，
你的算法并没有写清楚算法思路，
而且图的存储也有问题

三、

1、
```
void sort(int a[],int n){
    int r[n]; //结果
    int left=0; //左指针
    int right = n-1; //右指针
    for(int i=0; i<n; i++){
        if(a[i]%2==0){ //若偶放右
            r[right--]=a[i];
        }else{
            r[left++]=a[i]; //若奇放左
        }
    }
    for(int i=0;i<n;i++){ //结果写入原数组
        a[i]=r[i];
    }
}
```

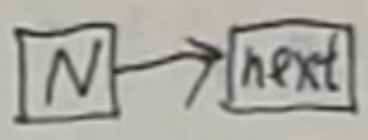
20

2、
```
bool s; //是否非质数
void di(int n){
    for(int i=2;i<n;i++){ //试图找除数
        if(n%i==0){
            if(!s){ //分解成功
                s=true;
                print("%d=%d",n,i); //开头
            }else{
                print("*%d",i); //已联,写后尾
            }
            di(n/i); //续分
            break;
        } //分不下去了
    }
    if(!s){ //如果没成功分过.
        if(n<2){
            print("0"); //0
        }else{
            print("%d",n); //n
        }
    }else{print("%d",n);} //是已(成功分过的)情况)
}
```

18

可能存在多个重复的值没有考虑进去
比如16=2*2*2*2,
按你的逻辑只会输出一次2

```
3、struct Node{ //节点
15    int value; //值
      Node* next; //子
   };
```



c语言申请新节点
需要使用malloc函数申请新的内存空间

```
void inserts(Node* node){
   while(node->next !=NULL){
      Node * one = {node->value+node->next->value, node->next}; // ①
      node->next = one; // ②
      node = one->next // 下一个
   }
}
```