就是判断两个 byte number 是不是互为 gray code。

我的问题是时间复杂度是多少？

是 O（n）？还是 constant ？

不要忘记加 byte

```
public static int grayCheck(byte term1, byte term2){
        byte x = (byte)(term1^term2);
        int count = 0;
        while(x!=0){
            x = (byte)(x&(x-1));
            count++;
        }
        return count == 1?1:0;
    }
```

另外还有个简单的版本

```
byte x = (byte)(a^b);
return x!=0 && ((byte)(x&(x-1)==0));
```

这个时间复杂度是 1 么？

```
byte x = (byte)(term1 ^ term2);
int total = 0;
while(x != 0){
x = (byte) (x & (x - 1));
total++;
}
if(total == 1) return 1; else return 0;
```

```
byte x = (byte)(term1 ^ term2);
    int total = 0;
    while (x != 0) {
  x = (byte)(x & (x - 1));
  total++;
    }
    return total == 1 ? 1 : 0;
```

```
public static int gray(byte term1, byte term2) {      int result = term1 ^ term2;      int
count = 0;      int ch = 0x01;      for (int i = 0 ; i < 8; i++) {          if
```

```
((result&ch) != 0) {                    count++;              }              ch = ch <<
1;        }       if (count == 1) {              return 1;       }       return 0; }
```

```
public static int greyCode(byte element1, byte element2) {
            byte res = (byte) (element1 ^ element2);
            for (int i = 0; i <= 7; i++) {
                  byte temp = (byte)(1 << i);
                  if (temp == res) {
                        return 1;
                  }
            }
            System.out.println("No");
            return 0;
      }
```

graycode 如果用

```
x = term1 & term2;
if ( (x & (x-1)) == 0)
  return 1;
else
  return 0;
```

会有一个 Case 通不过，所以只好一位位 shift 检查了。

昨天做了，确实如你所说，有一个 case 过不了。
然后把 x 的声明改成 unsigned char x = term1 ^ term2.剩下的一样。。。过了。。。现在还在想怎么回事。。估计和符号数的 x-1 有关。。。

你试试-128（10000000）和 0（00000000） ?

```
byte x = (byte)(term1 ^ term2);
int total = 0;
while(x != 0){
x = (byte) (x & (x - 1));
total++;
}
if(total == 1) return 1; else return 0;
```

看了 lz 的帖子感觉应该是 byte 数值类型的问题，也咨询了系里的大神，感觉上原因可能是这样的：
如果用"int n = term1 ^ term2 "的话， 可能会有错误，因为负数前面会添加好多 1，比如

byte 的 -1 本来有 7 个 1 在符号位后，而变成 int 之后就不止 7 个了。
楼主这里全用了 byte 所以避免了这个问题。
所以要么全转 byte 要么就用 mask。
给之后做 OA 的小孩借鉴一下~
希望大家不会犯我们犯过的错误~祝各位找工作顺利~

感谢楼主分享，小弟有一个问题没想明白。就是 Gray Code 那道题，为什么必须要做那个循环？
a ^ b 以后得到的是 8 位里只有一个 1，那为什么不能直接判断 n & (n -1) 是否为零即可呢？
谢谢指教！

如果 a 和 b 不是互为格雷码，异或之后得到的就不止一个 1。
所以要检查 1 的个数

谢谢回复！
但是如果不是互为格雷码，n & (n-1)就不为 0 对吧。
我的意思是能不能直接判断 n&(n-1) 的结果，如果为 0 则返回"是"，不为 0 则返回"否"？这样就没有必要进行循环了。

n & (n - 1) 只能判断是否 1000，100 这样 pow(2, n)这样的

谢谢回复！但是确实是这样的啊，a 和 b 如果是 gray code 关系，则做 XOR 以后不就应该是只有一个 1 吗？那 n&(n-1)就可以判断对吧？难道还有其他什么可能吗？

应该要排除如果两个数相等是不能成为 gray code 的，那样 XOR 后结果为 n=0，在 c++上测试 n&(n-1)的结果还是 0，会判断为时 gray code

我今天做 grey code 那题了，除了看 x&(x-1)==0 之外还专门事先判断了一下两个数相等的情况，但是还是有一个 test case 没有过。我猜不是这个的问题

是正负数的问题？

判断 x&(x-1)==0 的时候，前面你有没有加 byte？(byte)(x&(x-1)) == 0

gray code 这个题我知道 15 个 case 和 16 个 case 的区别了,在于 while 判断的条件。
while(c>0)和 while(c!=0)
可能是会出现负数。

去掉元音：

```
StringBuffer sb = new StringBuffer();
String v = "aeiouAEIOU";
for(int i = 0; i < string.length(); i++){
if(v.indexOf(string.charAt(i)) > -1) continue;
sb.append(string.charAt(i));
}
return sb.toStirng();
```

```
StringBuilder sb = new StringBuilder();        String v = "aeiouAEIOU";
        for (int i = 0; i < s.length(); i++) {
    char c  = s.charAt(i);
    if (v.indexOf(c) == -1) sb.append(c);
    }
        return sb.toString();
```

楼主做的是去元音和 rightrotation （挺傻逼的因为去元音的条件判断写错了，慌了以后卡了半天，其实不应该的，如果是空或者 length==0，return "" 而不是 null，不然一个 case 过不去)

remove all vowels in a string, for example input = "abcde asdf wer" output = "bcd sdf wr"
算法很简单，就是吧元音往后放。根据考场回忆，事后在 eclipse 上调试成功后，具体代码如下（本人考试时候偷偷使用了 eclipse）：

```
public class RemoveVowel {


        public static boolean isVowel(char x){
                if( x == 'a' || x == 'A' ||
                        x == 'e' || x == 'E' ||
                        x == 'i' || x == 'I' ||
                        x == 'o' || x == 'O' ||
```

```
            x == 'u' || x == 'U' ){
            return true;
        }else{
            return false;
        }
}

public static String removeVowel( String s){
        StringBuilder S = new StringBuilder(s);
        int Vstart = -1;
        int Vcounter = 0;

        for (int i = 0; i < S.length(); i++) {

            if (isVowel(S.charAt(i))) {
                if( Vstart == -1 ){
                    Vstart = i;
                }
                Vcounter++;
            }else if( Vstart != -1 ){
                char y = S.charAt(i);
                char z = S.charAt(Vstart);
                S.setCharAt(Vstart, y);
                S.setCharAt(i, z);
                Vstart++;
            }
        }
```

```
            return S.toString().substring(0,s.length() - Vcounter);
        }

        public static void main(String[] args) {
            System.out.println(removeVowel("the rain in spain is mainly from the plains."));

        }

}
```

right rotation：

```
public static int rightRotate(String word1, String word2) {
            if (word1 == null || word2 == null || word1.length() == 0 || word2.length() ==
0 || word1.length() != word2.length()) {
                return -1;
            }
            String str = word1 + word1;
            return str.indexOf(word2) != -1 ? 1 : -1;
        }
```

CTCI 第一章最后一题（ ？ ）

但是要注意的是， 首先判断两个 string 长度是否相同， 反例: word1 = "aaa" word2 = "a"，否则有小 bug， a 是 aaaaaa 的 substring

考场代码回忆如下

```
public static boolean checkRightRotation(String word1, String word2){
        if(word1.length() != word2.length()){
                return false;
        }
        word1 = word1 + word1;
        if(word1.contains(word2)){
                return true;
        }else{
                return false;
        }

}
```

```
public static int numOfArithmeticSlices(int[] arr) {. 1point 3acres 璁哄潧

    int count = 0, start = 0, end = 1, d = 0;. 鍥磋　鎴戣滑@1point 3 acres

    while (start < arr.length-2) {

        if (start+1 == end) {

            d = arr[end] - arr[start];

            end++;


        } else {.鐣欑　璁哄潧-涓€浜�-涓夊垎鏃�

            if (end == arr.length || arr[end] - arr[end-1] != d) {

                if (end - start >= 3) {

                    count += (end - start - 2);

                }

                start++;

            } else-google 1point3acres
```

```
                    end++;
                }
            }
            return count;
        }
```

```
public static void product (int[] ary){
        if(ary.length > 2){
        double sum = 1;
        for (int i = 0; i < ary.length; i++) {
            sum = sum * ary[i];
        }
        for (int j = 0; j < ary.length; j++) {
            ary[j] = sum / ary[j];
        }
    }
    }
```

http://ragingcat.info/fancybox/interviewzen_amazon_fan.pdf

考的三道题和地里说的基本一样：
1. find loop in the linked list
2. get intersection of two lists
3. give a list of students' test scores (studentId, testdate, testscore). calculate final score for each student. final score = average score of student's highest five scores.

[size=14.44444465637207px]1.check whether there is a loop in the linked list. (it is the same with leetcode. very easy)
[size=14.44444465637207px]2.Find the K closest points to the origin in 2D plane, given an array containing N points.

[size=14.44444465637207px]这一题的基本思想肯定是使用一个大小为 K 的 bomK(大顶堆). 遍历所有 N 个点,如果当前点到原点的距离小于堆顶的值,则入堆,最后输出堆里面的元素即可.
[size=14.44444465637207px]由于 N 远大于 K,而此时最费时的工作是计算 x2 + y2，所有要找到一个 herious，大概的确定范围然后在对这个范围里面的元素做复杂计算。
[size=14.44444465637207px]herious 的方法有很多，可以用欧拉 distance(x + y)，或者 max(x, y)。
[size=14.44444465637207px]想法是这样的，先使用 max(x, y) 来排序最开始所有的点，sort array。
[size=14.44444465637207px]然后 使用大顶堆来做。
[size=14.44444465637207px]当最后发现大顶堆的堆顶的元素的 x2 + y2 <= 来到的元素的 max(x, y)要小的话就停止。此时堆里面的元素即结果。

(1) Given a linked list of integers, write a function to determine whether the given list has a loop or cycle anywhere in the list. The integer values may not be relied upon to be distinct. You may use the JDK or the standard template library. Your solution will be evaluated on correctness, runtime complexity (big-O), and adherence to coding best practices. A complete answer will include the following:
Document your assumptions
Explain your approach and how you intend to solve the problem
Provide code comments where applicable
Explain the big-O run time complexity of your solution. Justify your answer.
Identify any additional data structures you used and justify why you used them.
Only provide your best answer to each part of the question.

Cycle detection,老生常谈……

(17) A museum is selling tickets to a fundraiser. As people place orders, their customer IDs are appended to a linked list. Due to an error, too many tickets have been sold and the unfortunate customers must now be notified that they cannot purchase a ticket after all. Given the list and the number of customers to notify (k), return the node in the list for the first customer that needs to be notified. An optimal solution will not use any additional data structures. You may assume k is less than the number of nodes in the list. You may use the JDK or the standard template library. Your solution will be evaluated on correctness, runtime complexity (big-O), and adherence to coding best practices. A complete answer will include the following:
Document your assumptions
Explain your approach and how you intend to solve the problem
Provide code comments where applicable
Explain the big-O run time complexity of your solution. Justify your answer.

Identify any additional data structures you used and justify why you used them.
Only provide your best answer to each part of the question.
Example:
Input: myList: 1 -> 8 -> 4 -> 2 -> 7 -> 13 -> 3 k: 2

Output: 13 -> 3

Last K elements, 老生常谈……

(4) Find the K closest points to the origin in 2D plane, given an array containing N points. You can assume K is much smaller than N and N is very large. You need only use standard math operators (addition, subtraction, multiplication, and division). You may use the JDK or the standard template library. Your solution will be evaluated on correctness, runtime complexity (big-O), and adherence to coding best practices. A complete answer will include the following:
Document your assumptions
Explain your approach and how you intend to solve the problem
Provide code comments where applicable
Explain the big-O run time complexity of your solution. Justify your answer.
Identify any additional data structures you used and justify why you used them.
Only provide your best answer to each part of the question.

三个题：
1. 单向链表找循环
2. 用 vector 算每个学生最高的五个分数的平均
3. 两个已排序链表 merge 并找交点
这段时间的 online assessment 应该都是这三道，注意第二题要用容器，第三个不光只找交点，还要 merge。
祝大家顺过！

AmazonID: 给做题的 ID 号