**1）当一个无符号整数x是非2的幂次方时候怎么求大于x且最接近x的2的幂次的数**

**static** **public** **int** nextPowerOfTwo (**int** value) {

**if** (value == 0) **return** 1;

value--;

value |= value >> 1;

value |= value >> 2;

value |= value >> 4;

value |= value >> 8;

value |= value >> 16;

**return** value + 1;

}

原理：将该数的最高位为1的二进制后面的数都变为1，最后再加上1即可

value >> 1保证最高位为1相邻的位变为1

value >> 2保证最高位为1的相邻3为变为1，依次类推

**2）将链表中的所有元素为奇数的节点移到元素为偶数的前面，并且使奇数之间顺序反转，偶数之间顺序反转**

struct Node\* swapList(struct Node \*list)

{

if (list == NULL || list->next == NULL)

return list;

struct Node \*oddTail = NULL, \*oddHead = NULL, \*evenHead = NULL, \*curNode = list;

while (curNode)

{

struct Node \*tmpNode = curNode;

curNode = curNode->next;

if (isOdd(tmpNode->value)) // 当前为奇数

{

if (oddTail == NULL) // 第一个奇数

{

oddHead = tmpNode;

oddTail = tmpNode;

tmpNode->next = NULL; // 这里非常重要，要置空

}

else

{

tmpNode->next = oddHead;

oddHead = tmpNode;

}

}

else

{

if (evenHead == NULL) // 第一个偶数

{

evenHead = tmpNode;

tmpNode->next = NULL; // 这里非常重要，要置空

}

else

{

tmpNode->next = evenHead;

evenHead = tmpNode;

}

}

}

if (oddTail != NULL) // 找到了奇数

{

oddTail->next = evenHead;

return oddHead;

}

else // 没有找到奇数

{

return evenHead;

}

}

对于数组呢？如果要保证顺序呢？

**3）给一个正整数n，找到若干个完全平方数（1,4,9,...）使得它们的和等于n，你需要让平方数的个数最少**

**给出n=12，返回3，因为12=4+4+4**

**给出n=13，返回2，因为13=4+9**

**4）一个N×N方阵，现从左上角到右下角找一条路径，只能向下或者向右走，要求路径中各数字和最大**

**5）一个整数分解为连续正整数之和**

http://blog.csdn.net/wang664626482/article/category/6124634/1

**全排列**

**如何判断一个单向链表有环？**

**Kmp算法**

**红黑树**

**非递归遍历二叉树**

**排序（基数排序、shell排序、桶排序、其他排序及非递归方式）**

**八数码问题**

**Kd-tree**

**R-tree**

**A\*算法及优化**

**并查集**