1. 一堆平面上的二维点， 找离origin最近的k个点

2. Find loops in LinkedList

3. Start position of the loop

4. Given four integers, make F(S) = abs(S[0] - S[1]) + abs(S[1] - S[2]) + abs(S[2] - S[3]) to be largest.

5.

(10) Given a list of integers and a window size, return a new list of integers where each integer is the sum of all integers in the kth window of the input list. The kth window of the input list is the integers from index k to index k + window size - 1 (inclusive).
For example, [4, 2, 73, 11, -5] and window size 2 should return [6, 75, 84, 6]. For another example, [4, 2, 73, 11, -5] and window size 3 should return [79, 86, 79].
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.

6. Given an array, return the number of possible arithmetic sequence.

A sequence of numbers is called *arithmetic* if it consists of at least three elements and if the difference between any two consecutive elements is the same. For example, these are arithmetic sequences:

```
1,  3,  5,  7, 9
7,  7,  7,  7
3, -1, -5, -9
```

The sequence [1, 1, 2, 5, 7] is not arithmetic.
A zero-indexed array A consisting of N numbers is given. A *slice* of that array is any pair of integers (P, Q) such that $0 \le P < Q < N$.
A slice (P, Q) of array A is called *arithmetic* if the sequence:

A[P], A[P+1], ..., A[Q−1], A[Q]

is arithmetic. In particular, this means that $P + 1 < Q$.
Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given array A consisting of N numbers, returns the number of arithmetic slices in A. The function should return −1 if the result exceeds 1,000,000,000.
For example, given array A such that:

```
A[0] = -1
A[1] =  1
A[2] =  3
A[3] =  3
A[4] =  3
A[5] =  2
A[6] =  1
A[7] =  0
```

the function should return 5 because there are five arithmetic slices of that array, namely:

```
(0, 2)  (2, 4)  (4, 6)  (4, 7)  (5, 7)
```

Assume that:

- N is an integer within the range [0..100,000];
- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

7. Given a tree of N nodes, return the amplitude of the tree.

8. Round Robin Scheduling
大概是这样，给一个int[] arrival time, int[] Execution time, int q. 例子： 【0，1，4】 【5，2，3】 q=3. 输出的是average wait time 2.3333333
q是指quantum，round robin的算法每隔quantum个时间都会强制切换到下一个任务，如果当前任务没有执行完的话。

return float

9. Shortest job first
example1：
requestTime: [0, 2, 4, 5]
duration: [7, 4, 1, 4]
题目要求是short task first。也就是说先处理p1，处理之后的时间是7，那么之后就处理p3,因为它的处理时间为1，最短。p3之后是p2，最后是p5。这个例子的average waiting time 是4，test case里给的答案.

example2:. 1point 3acres 璁哄潵
requestTime: [0, 1, 3, 9]
duration: [2, 1, 7, 5]
average waiting time 是0.5

10. Cache Miss, given the max size of a LRU cache and a input array, calculate the miss times
输入 maxSize， input int array
输出是miss cout

example： size = 4， input array 【1，2，3，4，5，4，1】
1 miss
2 miss
3 miss
4 miss
5 miss 替换 1
4 hit 把4提前到第一位
1 miss 替换 2
大概就是这样。。。

实现思路：

```
vector<int> store;
int miss;
for(x in input){
   it = store.find(store.begin, store.end, x);

if(it!= store.end()){
erate(it);
push_back;
}
if not find—>
{
if(size>=capacity)
erate first element
push_back(x);}
else
```

push_back();

miss++;
}

11. Days. 给一个array和一个天数，问你第N天数组是啥。具体变化规则：一个数的两边一样，那么此数字变0，否则为1.对于头尾，默认头左为0，尾右为0
给里一串数字cells(比如[1, 0, 0, 0, 0, 1, 0, 0])以及一个天数days， 问几天以后这一个数组里的数最后变成什么样?
变化的具体要求是：一个cell, 如果它左右两边的数一样，那么就将这个数设置为0， 不一样则为1(题目中用inactive和active来描述，后来给出coding的时候用0和1来代替)。例如cells[2]，因为cells[1] == cells[3]，所以第二天cells[2] == 0; cells[4]的话，因为cells[3] != cells[5]，所以第二天cells[4] = 1。因为第一个数和最后一个数只有一个相邻的数，所以默认这个cells[0]的左边及cells[len-1]的右边都为0。

example:
cells: [1, 0, 0, 0, 0, 1, 0, 0]. 涓€浜-涓夋垯-鍦帮紝鐙??鍙戝竷
days: 1
那么一天以后这个数组就变成[0, 1, 0, 0, 1, 0, 1, 0]
做的时候数组可以看成(0)[1, 0, 0, 0, 0, 1, 0, 0](0)来做，即左右两边的树默认为0。

12. Rotate m*n Matrix 90,给一个flag参数用来决定是往左还是往右

13. Given a BST and its root, 找从root到leaves的权值最小的路径。不要求返回路径，只要求返回路径上节点值的sum。要求这个sum最小。用divide and conquer就行。注意这题有null node，以及它的null left和null right

14. 给出一个已经排序的循环链表，往里面插入一个节点。http://www.geeksforgeeks.org/sorted-insert-for-circular-linked-list/ 实际的题有一点不一样的是它每次给你的HEAD不一定是链表最小那个。题目定义了一个CNode，其实跟ListNode一样，有val，有指针，list首尾相连而已，list里的node升序排列，题目也不难，不过有两点注意一下：一个是给你的CNode start不一定是最小值的CNode，所以要先找到最小的点

15. Greates Common Divisor；输入是一个int数组，输出是所有int类型的最大公约数。
我的思路是，一个数组的最大公约数一定是某两个数最大公约数和其他数的最大公约数
所以求出来前两个的最大公约数，再拿到这个数字和之后所有的数字求最大公约数就是结果

16. find path in 2D matrix，输入一个2D int array,可能是三种情况，0，1，9。 0表示可以走， 1表示障碍不能通过，9表示最终结果。出生点在（0，0），返回true/false 表示能不能从（0，0）走到有9的位置上。并没有明说9是否唯一。
(0, 0)可能是终点