

# WEEK1

段雷

# 目录

## CONTENTS



01

渐进时间复杂度



02

位运算



03

记录路径



04

栈、队列



05

题目讲解

# Rules

1. 每周完成作业 (3 + 2)
  - (1) 3道课堂相关题目
  - (2) 2道板刷题
  - (3) 讲解完后一周内要把讲过的内容完成
2. 课件不外传，涉及题目来源问题
3. 教室容量有限，不扩大规模

4. **Practice > Think**

# 一些说明

1. 数据结构 + 算法（**不是**机器学习、深度学习的算法）
2. PPT只会放主干的要点，详细的解释说明主要靠嘴和黑板
3. 会讲一些最新的网测题
4. 每道题我会轮流请大家说一下自己的想法，错了没关系，重要的是参与进来
5. 如果大家有别的想法、有意思的题目可以带过来一起讨论
6. 有疑问或者我没讲清楚的地方欢迎随时提问

---

# 01 渐进时间复杂度

# 渐进（近）时间复杂度（asymptotically）

## $\Theta$ 记号 渐进确界

$\Theta(g(n)) = \{f(n) : \text{存在正常数 } c_1, c_2, \text{ 和 } n_0, \text{ 使得对所有 } n \geq n_0, \text{ 有 } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$

## $o$ 记号 渐进上界

$o(g(n)) = \{f(n) : \text{存在正常数 } c \text{ 和 } n_0, \text{ 使得对所有 } n \geq n_0, \text{ 有 } 0 \leq f(n) \leq c g(n)\}$

## $O$ 记号 非渐进紧确的上界

$O$ 记号与 $o$ 记号的定义是类似的，主要区别在于对  $f(n) = O(g(n))$ ，界  $0 \leq f(n) \leq c g(n)$  对某个常数  $c > 0$  成立，但对  $f(n) = o(g(n))$ ，界  $0 \leq f(n) \leq c g(n)$  对所有常数  $c > 0$  成立

## $\Omega$ 记号 渐进下界

## $\omega$ 记号 非渐进紧确的下界

# 渐进（近）时间复杂度

$$T(n) = 2n^2 + 3n + 4$$

$$T(n) = 2T(n/2) + n$$

求 1 ~ n 的和

```
1 tot = 0
2 for i in range(n + 1):
3     tot += i
```

线性阶

$O(n)$

```
1 tot = (n + 1) * n / 2
2
```

常数阶

$O(1)$

# 渐进（近）时间复杂度

```
1 x = 1
2 while x <= n:
3     x *= 2
4
```

对数阶

$O(\log n)$

```
1 tot = 0
2 # a: n * n matrix
3 for i in range(n):
4     for j in range(n):
5         tot += a[i][j]
6
```

平方阶

$O(n^2)$

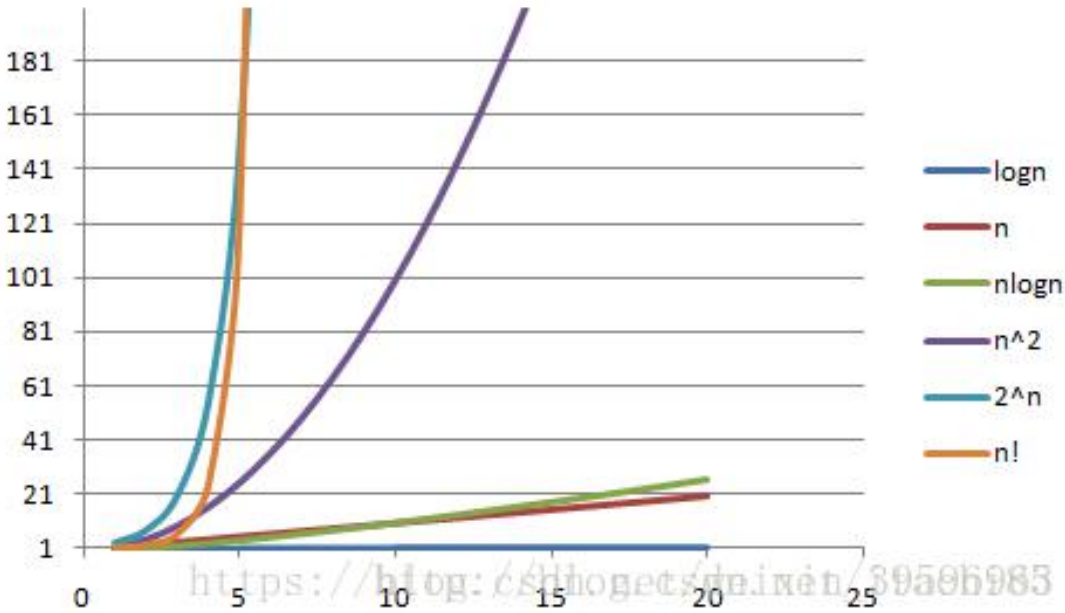
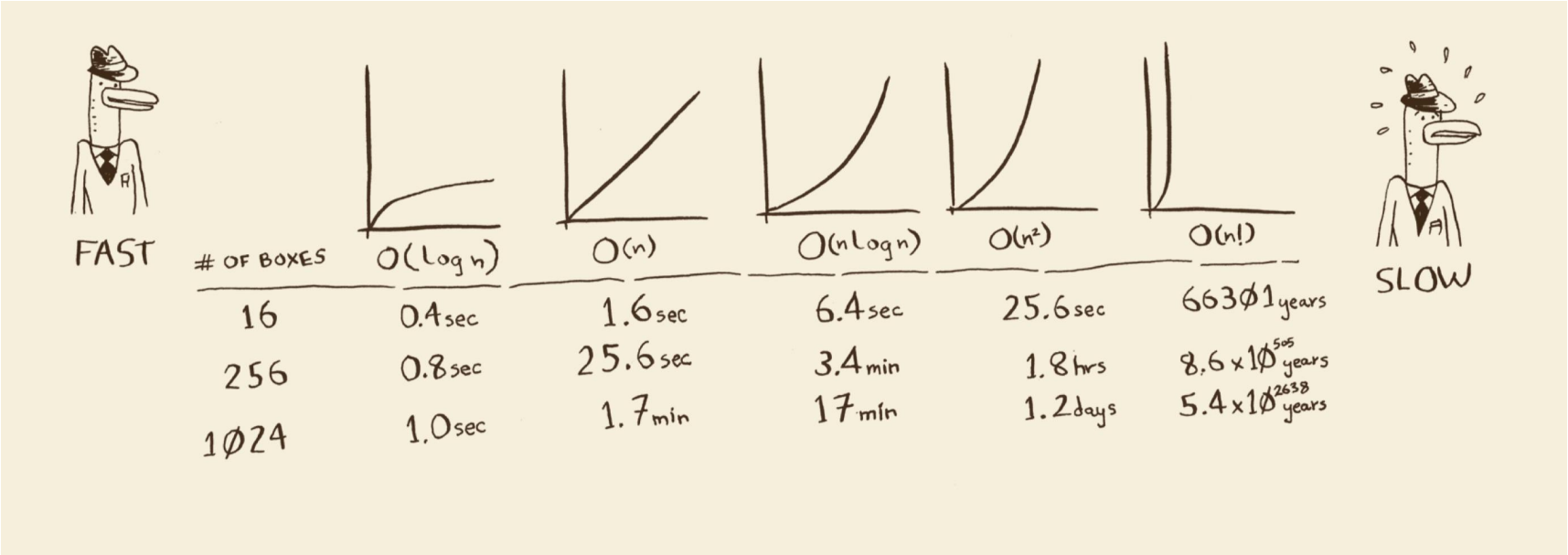
```
1 def fact(n):
2     if n == 1:
3         return 1
4     return n * fact(n - 1)
5
```

阶乘阶

$O(n!)$



# 渐进（近）时间复杂度



---

## 02位运算

# 位运算

- 与运算	&	Q1: 求 x 二进制下第y位 (y从0开始) 是什么
- 或运算		A1: $x \gg y \& 1$
- 取反运算	~	
- 非运算	!	Q2: 将x第y位取反
- 异或运算	^	A2: $x \wedge (1 \ll y)$
- 位移运算	>>, <<	

# 位运算

原码

	正数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

	负数
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

反码

	正数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

	负数
-0	1111
-1	1110
-2	1101
-3	1100
-4	1011
-5	1010
-6	1001
-7	1000

反码

原码

	负数
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111



# 位运算

## 补码

	正数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

补码

	负数
0	<del>0000</del>
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

反码

	负数
-0	1111
-1	1110
-2	1101
-3	1100
-4	1011
-5	1010
-6	1001
-7	1000

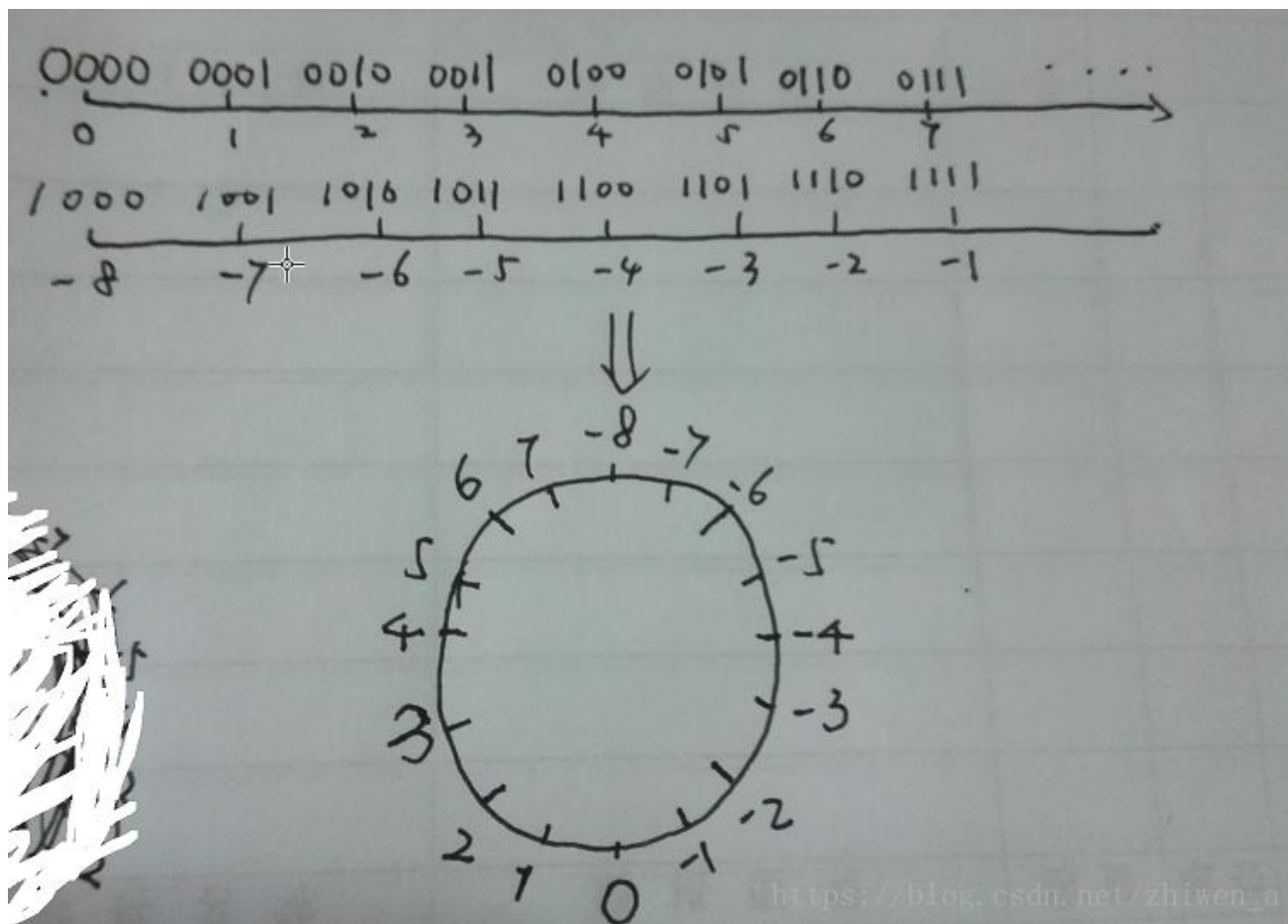
原码

	负数
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111

[https://blog.csdn.net/zhiwen\\_a](https://blog.csdn.net/zhiwen_a)

# 位运算

## 补码



口诀：负数取反加1

---

## 03记录路径

# 记录路径

P1:

有一个数字金字塔，从金字塔的顶端走到最下面，每次只能往下一层相邻的2个数字走，将经过的数字求和，问和最大为多少？

同时将经过的数字按顺序输出来





# 记录路径

P2 (某公司OT) :

有一个奇数长度的方阵，从四个角出发，到达正中心，每次只能横着或者竖着走，要求走的格子数最少。目标是将经过的数字求和，问和最大为多少，且要求输出路径

如右图为答案为 d,d,r,r

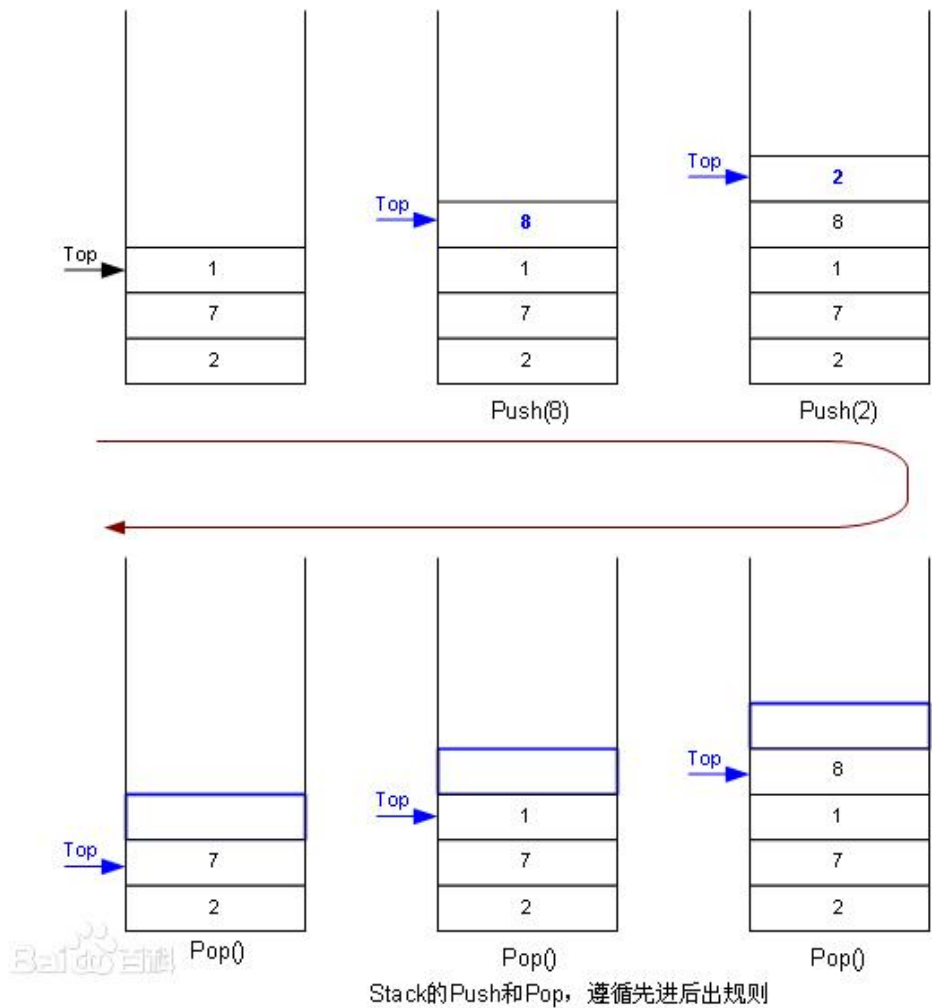
5	-1	3	4	9
2	-1	-2	-3	5
2	1	5	-10	3
-2	-3	-5	-6	-7
1	2	3	4	5

---

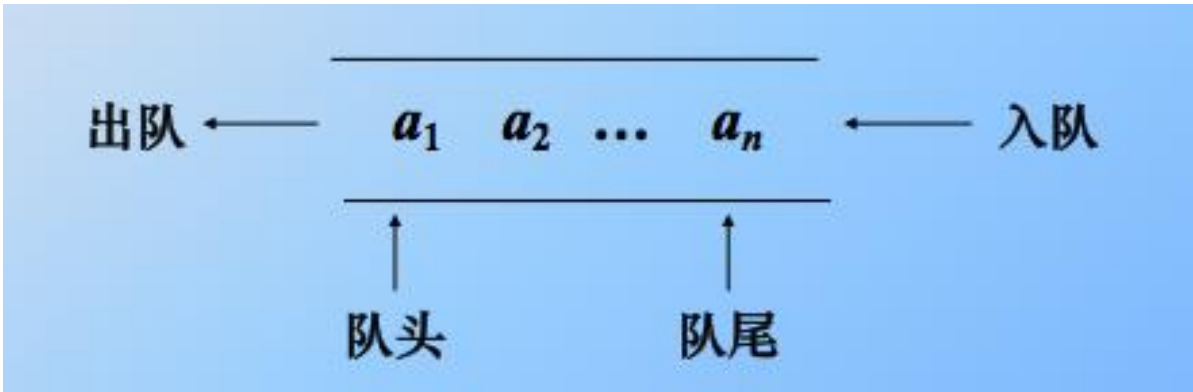
## 04栈、队列

# 栈、队列

## 栈



## 队列



# 单调队列

## Poj 2823

An array of size  $n \leq 10^6$  is given to you. There is a sliding window of size  $k$  which is moving from the very left of the array to the very right. You can only see the  $k$  numbers in the window. Each time the sliding window moves rightwards by one position. Your task is to determine the maximum and minimum values in the sliding window at each position.

Window position								Minimum value	Maximum value
[1	3	-1]	-3	5	3	6	7	-1	3
1	[3	-1	-3]	5	3	6	7	-3	3
1	3	[-1	-3	5]	3	6	7	-3	5
1	3	-1	[-3	5	3]	6	7	-3	5
1	3	-1	-3	[5	3	6]	7	3	6
1	3	-1	-3	5	[3	6	7]	3	7

### Sample Input

```
8 3
1 3 -1 -3 5 3 6 7
```

### Sample Output

```
-1 -3 -3 -3 3 3
3 3 5 5 6 7
```

## 单调栈

Poj 3250

Some of Farmer John's  $N$  cows ( $1 \leq N \leq 80,000$ ). FJ wants to count the number of other cows that can see the top of other cows' heads.

Each cow  $i$  has a specified height  $h_i$  ( $1 \leq h_i \leq 1,000,000,000$ ) and is standing in a line of cows all facing east (to the right in our diagrams). Therefore, cow  $i$  can see the tops of the heads of cows in front of her (namely cows  $i+1$ ,  $i+2$ , and so on), for as long as these cows are strictly shorter than cow  $i$ .

# 单调栈

## Poj 3250

### Sample Input

6  
10  
3  
7  
4  
12  
2

### Sample Output

5

```
      =  
=      =  
=  -  =      Cows facing right -->  
=  =  =  
= - = = =  
= = = = = =  
1 2 3 4 5 6
```

Cow#1 can see the hairstyle of cows #2, 3, 4

Cow#2 can see no cow's hairstyle

Cow#3 can see the hairstyle of cow #4

Cow#4 can see no cow's hairstyle

Cow#5 can see the hairstyle of cow 6

Cow#6 can see no cows at all!

---

## 05题目讲解

# HW 1

## 131. Palindrome Partitioning

Medium

👍 1102

💬 45

♡ Favorite

🔗 Share

Given a string *s*, partition *s* such that every substring of the partition is a palindrome.

Return all possible palindrome partitioning of *s*.

**Example:**

**Input:** "aab"

**Output:**

```
[  
  ["aa","b"],  
  ["a","a","b"]  
]
```



# HW 2

## 134. Gas Station

Medium

👍 874

💬 305

♡ Favorite

🔗 Share

There are  $N$  gas stations along a circular route, where the amount of gas at station  $i$  is `gas[i]`.

You have a car with an unlimited gas tank and it costs `cost[i]` of gas to travel from station  $i$  to its next station  $(i+1)$ . You begin the journey with an empty tank at one of the gas stations.

Return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1.

### Note:

- If there exists a solution, it is guaranteed to be unique.
- Both input arrays are non-empty and have the same length.
- Each element in the input arrays is a non-negative integer.

### Example 1:

#### Input:

`gas = [1,2,3,4,5]`

`cost = [3,4,5,1,2]`

**Output:** 3

# HW 3

## 138. Copy List with Random Pointer

Medium

👍 1830

💬 482

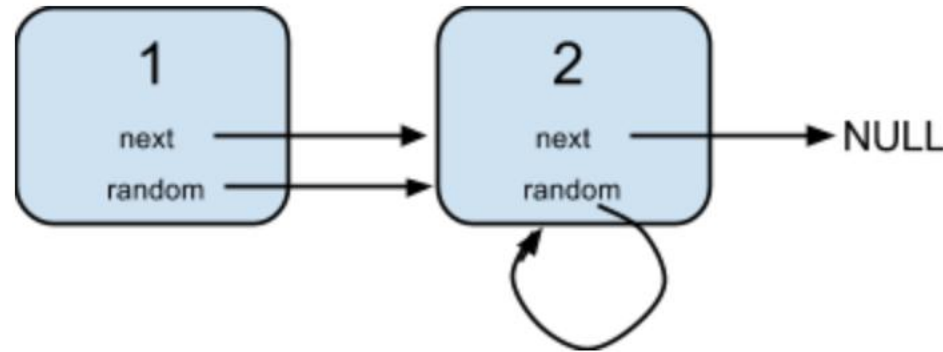
♡ Favorite

🔗 Share

A linked list is given such that each node contains an additional random pointer which could point to any node in the list or null.

Return a **deep copy** of the list.

**Example 1:**



**Input:**

```
{"$id": "1", "next": {"$id": "2", "next": null, "random": {"$ref": "2"}}, "val": 1}
```

**Explanation:**

Node 1's value is 1, both of its next and random pointer points to Node 2. Node 2's value is 2, its next pointer points to null and its random pointer points to itself.

# GS 20190904 OT P1

有一个  $n*m$  的数字棋盘，两个人轮流从里面挑选数字，每个人的得分是挑选过的数字之和。两个人的目的都是让自己的得分尽可能高。如果有一个人挑选了  $(i,j)$  这个位置的数字，那么第  $j$  列其他数字都不能再被挑选。问两人都采取最优策略下，第一个人的得分减第二个的得分为多少？

Sample Input:

```
3 6
3 7 5 3 4 5
4 5 2 6 5 4
7 4 9 7 8 3
```

Sample Output:

```
3
```

# GS 20190904 OT P2

0~9每个数字都会被映射到0~9中的数字，且构成双射。现在给出映射后的数字列表，要求将这个数字列表按原数字大小从小到大排序后输出，如果原数字大小相同，那么按照映射后数字出现次序排列。

Sample Input:

```
10
3 5 4 6 2 7 9 8 0 1
3
990      (-> 669)
332      (-> 004)
32       (->  04)
```

Sample Output:

```
332
32
990
```