

# 思维题分享

## Car

题目传送门: [\[2018牛客多校\] Car](#)

### 题意:

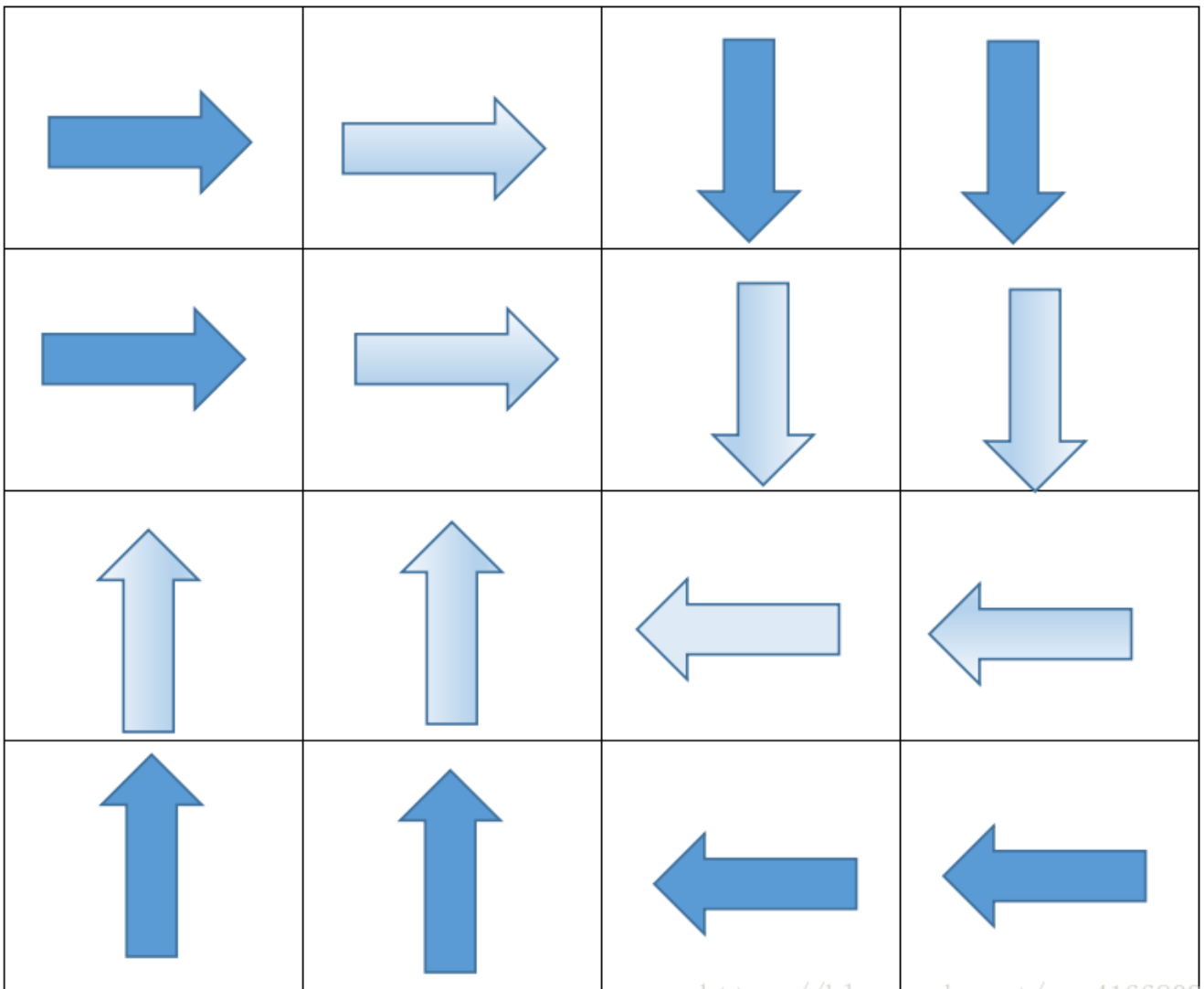
给一个  $n \times n$  的网格, 只能在边缘放小车且小车要向对岸直行  $n$  个格子, 每个小车速度相等; 在网格中会有  $m$  个陷阱, 在小车不会相撞或者走到陷阱的情况下最多能放多少辆?

$$1 \leq n, m \leq 10^5$$








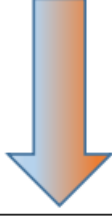


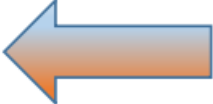









### 题解:

首先考虑没有陷阱的情况:

偶数:



奇数:

				
				
		中间只能二选一		
		$ans = 2 * n - 1$		
				

[https://blog.csdn.net/qq\\_41668093](https://blog.csdn.net/qq_41668093)

有陷阱的情况：

当n为偶数时 我们只需要分别标记 x y 这条路上是否已经有了陷阱 如果没有 ans则--

奇数时 中间那一辆车有两条路可以走，所以当中间 x y 那两条路都死了 其实才损失了一辆车，但前面因为那两条路 ans已经-2了 所以现在要ans+1

## jack & rose

### 题意：

Jack 和 Rose在泰坦尼克号上邂逅后（1912年），一起玩起了博弈，当时还没博弈论（博弈论，1913年才开始有人研究），但是2人都很聪明，都能够选择最优策略。Jack 和 Rose相对而坐，面前有n个石块，两个人轮流从中取石块，规定每次至少取一个，最多取m个，最后取光者得胜。因为2人关系不一般，游戏规则有点改变，每次开局，Rose先取，而Jack每次最多可以取m+1个，Rose还是最多取m个。

## 题解：

理论上来说：

- R如果想赢，就必须给J留 $M+2$ 个石头；
- J如果想赢，就必须给R留 $M+1$ 个石头；

比如每人最多拿10个石头，目前余下来11个石头的时候，先拿的必输。

但是J可以比R多拿一个，面对R给J留的 $M+2$ 的难题时，可以取一个，反而转化为J给R留 $M+1$ 个石头，最后让J胜利。

除非R一下子全部拿完，不然就是J赢。

换句话说，J可以抵挡住R的杀手锏，而R不能抵挡住J的杀手锏，导致J是赢家。

## String Problem

题目传送门：[2021 ICPC沈阳 M.String Problem](#)

## 题意：

给一个只包含小写字母的字符串  $S$  ( $1 \leq |S| \leq 10^6$ )，求出每一个前缀的最大子串是什么

## 题解：

首先我们可以发现：对于一个字符串，其字典序最大的子串，这个子串的结尾一定是整个字符串的结尾。

证明（反证法）：假设字符串范围是  $[1, n]$ ，其字典序最大的子串范围是  $[l, r]$  ( $1 < l \leq r < n$ )，那么存在一个子串  $[l, n]$ ，而这个子串的字典序是大于  $[l, r]$  的子串的（如果两串前缀完全一样则比较长度）

因此我们只需要找出前缀字符串最大子串的开头即可。

我们可以从前往后枚举每一位，并记录下当前字典序最大的前缀，以这个前缀的开头到枚举的整个前缀的结尾来作为字典序最大的子串。

## AC代码：

这道题的代码比较巧妙，在这里贴出来：

```
1 #include <iostream>
2 #include <cstdio>
3 #include <cmath>
4 #include <string>
5 #include <cstring>
6 #include <set>
7 #include <map>
8 #include <queue>
9 #include <vector>
10 #include <algorithm>
11 #include <iomanip>
12 #define LL long long
13 #define ULL unsigned long long
```

```

14 #define PII pair<int,int>
15 #define PLL pair<LL,LL>
16 #define PDD pair<double,double>
17 #define x first
18 #define y second
19 using namespace std;
20 const int N=1e6+5,mod=1e9+7;
21 int l[N];
22 char s[N];
23 int main()
24 {
25     scanf("%s",s+1);
26     int n=strlen(s+1);
27     for(int i=1;i≤n;) // 枚举字符串的每一位
28     {
29         if(!l[i]) l[i]=i; // 如果这一位之前没有记录, 那么其最大子串为[i,i]
30         int j=i,k=i+1;
31         while(k≤n&&s[j]≥s[k]) // 更新所有以i为开头, 以k为结尾的最大子串
32         {
33             if(!l[k]) l[k]=i; // 如果当前位没被记录过, 则以i为开头的子串为其最大子串
34             if(s[j]==s[k]) j++; // 如果当前位j与位k相等, 则j+1, 继续向后比较 (直到决定出谁才
是字典序最大的前缀)
35             else j=i; // 如果不相等还在while里, 说明以i开头的前缀还是最大的前缀
(比较结束, j从新置i)
36             k++;
37         }
38         while(i≤j) i+=k-j; // 跳出了上一个while, 说明[i,j]这个前缀字典序小于后面某个子
串, 则跳到j之后
39     }
40     for(int i=1;i≤n;i++) printf("%d %d\n",l[i],i);
41     return 0;
42 }

```

## Sum on Subarrays

题目传送门: [Problem - 1809C - Codeforces](#)

### 题意:

用  $-1000 \sim 1000$  的数, 构造一个长度为  $n$ , 有  $k$  个子序列的元素和为正数, 其余  $\frac{n \times (n+1)}{2} - k$  个子序列元素之和为负数的整数序列

## 题解：

$k$  个正数，我们已知长度为  $n$  的序列有  $\frac{(n+1)}{2}$  个子序列

对于  $k$  个整数，我们可以通过构造全正数序列来满足正数子序列需求

构造形如：2, 2, 2.....2, x, -1000, -1000

例如：

n=4, k=5时

2 2 -1 -1000

设  $k$  为当前状态仍需要的正数个数

前两个2可提供3个正数，还差2个整数，则该x需要保证自身是负数的情况下，与前面的数组组合提供2个正数

n=7, k=4时

2 2 -3 -1000 -1000 -1000

前两个2可以提供3个正数，还差1个正数，要保证自己是负数的情况下，与前面的数组组合只提供1个正数