算法基础——枚举和贪心

邓丝雨

课程安排

- 1月11：枚举(前缀和/差分、尺取、状压)、贪心
- 1月12：递归与分治、二分、三分，01分数规划
- 1月15：堆栈队列(单调队列)、堆、并查集
- 1月16：深搜、广搜、搜索剪枝
- 1月18：线性dp、背包问题、区间dp
- 1月19：树形dp、状压dp
- 1月21：图论相关概念、最小生成树、最短路1月22:基本数论、组合数学
- 1月27：线段树、树状数组
- 1月28：字符串hash、trie树、kmp、manacher
- 如果出现没讲完的情况，我会酌情拖堂or加课2333

关于交作业：

- https://docs.qq.com/form/page/DSUNxTkVNR21URWV4?_w_tencentdocx_form=1

课程说明与要求

- 希望大家能够养成写题解的习惯。
- 请大家独立完成编码。
- 知其然也要知其所以然，当个十万个为什么没有什么不好。

一、什么是算法?

- 来考虑一个题目：《明明的随机数》 NC16669
- 明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了N个1到1000之间的随机整数，对于其中重复的数字，只保留一个，把其余相同的数去掉。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作。

- 几种解法：
- 1.对数列进行去重
  ——没有标记的元素和他后面的元素两两比较，相同的则把后一个标记为不要，对去重之后的数组再排序（以冒泡为例）。
- 2.对数列进行排序（以冒泡为例），然后从小到大遍历，若当前a[i]和a[i-1]相等就不输出，否则就可以输出。
- 3.用一个数组b[i]表示i有没有出现过，每读入一个x ，就将b[x]赋值为1，表示x出现过了，最后从0到1000遍历b数组，如果b[i] =1 就输出i。

一、什么是算法?

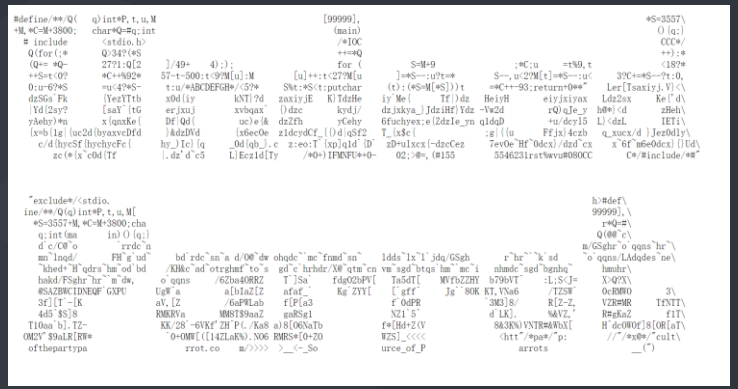
- 算法是解题过程的准确而完整的描述。它是一个有限规则的集合，这些规则确定了求解某一类问题的一个运算序列，对于某一类问题的任何初始输入，它能机械地一步一步地计算，并且通过有限步骤之后，计算终止并产生输出。

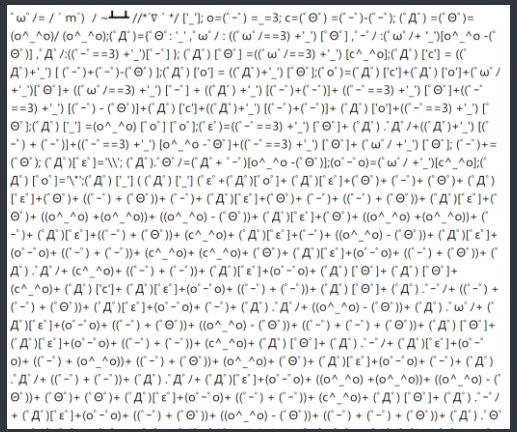
一、什么是算法?

- 算法的特征:
- 1.有穷性：一个算法必须总是在执行有限步之后结束。
- 2.确定性：算法的每一个步骤必须是确切地定义的。
- 3.输入：一个算法 有0个或多个输入。
- 4.输出：一个算法有1个或多个输出。
- 5.可行性：算法中要执行的每一个计算步骤都是可以在有限时间内完成的.

二、怎样评价一个算法?

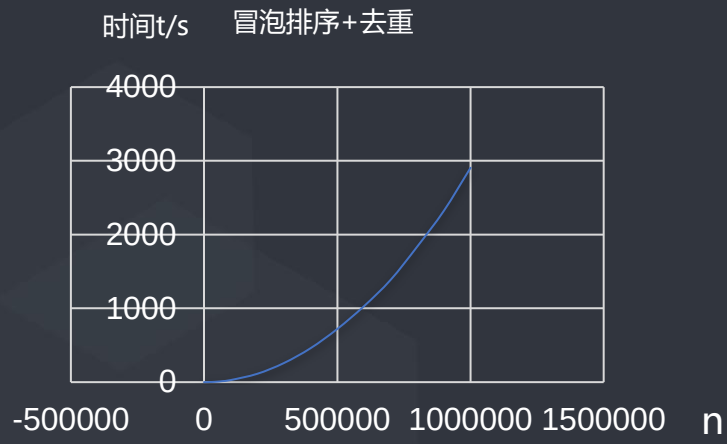
- 1.正确性
- 2.可读性
- 3.健壮性（容错性）——对不规范数据的处理能力（竞赛中一般不考虑）
- 4.时间复杂度
- 5.空间复杂度

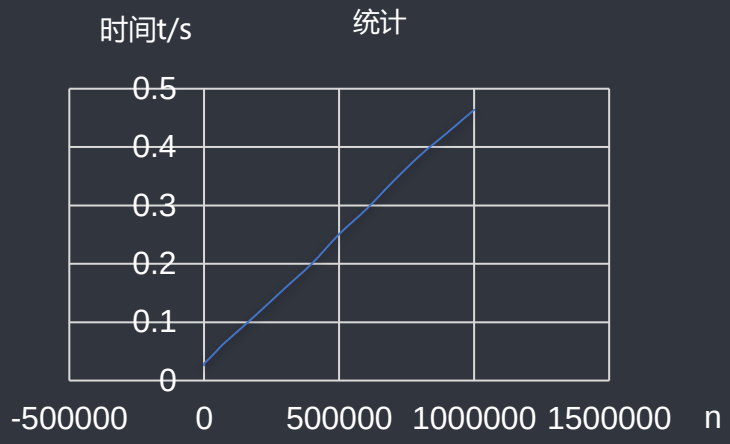




三、时间复杂度

- 我们先来看“明明的随机数”一题两个算法的运行时间的比较
- 随着n的增大，算法的运行时间是怎么变化的?





三、时间复杂度

- 时间复杂度是衡量程序的运行速度的量度，它在衡量的时候忽略了硬件的差异。
- 它是一个定性描述程序运行时间和数据规模n的关系的函数，这个函数即程序执行基本操作（加减寻址赋值简单的数学函数等）的次数，记做 T(n)

```
for(int i = 1; i <= k; i++)
{
    ans = (ans + f[n][i]) % mod;
}
```

三、时间复杂度

- 大O表示法：考察n趋近于正无穷时的情况
- 当且仅当存在常数c>0和N>=1，
- 对一切n>N均有|g(n)|<=c|f(n)|成立
- 称函数g(n)以f(n)为渐进上界或者称g(n)受限于f(n)。记作g(n)=O(f(n))。

三、时间复杂度

- 实际计算时，往往直接算，忽略低次项和系数即可
- Eg:冒泡排序
- g(n)=(n-1)C+(n-2)C+(n-3)C+……+1C

  =1/2（n^2-n）C

  =O(n^2)

```
for (i=0; i<len-1; i++)
    for (j=0; j<len-1-i; j++)
        if (a[j] > a[j+1]) swap(a[j], a[j+1]);
```

三、时间复杂度

- 常见的时间复杂的
- O(1) 和输入数据规模无关
- O(logn) 一般我们默认底数是2，不是2也没关系，用换底公式之后就是常数了
- O($\sqrt{n}$)
- O(n) 线性时间复杂度
- O(n^2)
- O(n^3).....
- O(C^n) C是一个常数，指数级
- O(n!) 阶乘级

```cpp
#include<cstdio>
using namespace std;
int n,t;
int a[21];
int main(){
        scanf("%d",&n);
        a[0]=1;
        a[1]=3;
        for(int i=2;i<=n;i++)
        {
                t=0;
                for(int j=0;j<=i-2;j++)
                t=t+a[j];
                a[i]=a[i-1]+2*t+2;
        }
        printf("%d",a[n]);
        return 0;
}
```

```cpp
#include<cstdio>
using namespace std;
int n,m[21][21];
int main(){
        scanf("%d",&n);
        for(int i=1;i<=n;i++){
                for(int j=i;j<=n;j++){
                        m[i][j]=i;
                        m[j][i]=i;
                }
        }
        for(int i=1;i<=n;i++){
                for(int j=1;j<=n;j++){
                        printf("%3d",m[i][j]);
                }
                printf("\n");
        }
}
```

```cpp
#include<cstdio>
using namespace std;
int n,m;
int main(){
        scanf("%d",&n);
        for(int i=2;i*i<=n;i++){
                if(n%i==0){
                        m=n/i;
                }
        }
        printf("%d",m);
        return 0;
}
```

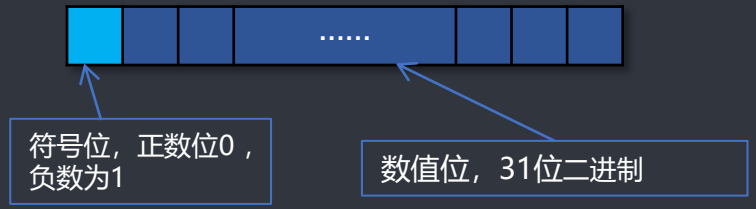
```cpp
#include<cstdio>
#include<cmath>
using namespace std;
int n;
double x,h;
int main(){
        scanf("%lf",&h);
        for(n=1;n<=9;n++){
                x+=h;
                h=0.5*h;
                x+=h;
        }
        x+=h;
        h=0.5*h;
        printf("%g\n%g",x,h);
        return 0;
}
```

四、空间复杂度

- 计算机内各种变量的存储
- sizeof 判断数据类型长度符的关键字 返回值类型为unsigned int
- 用法:
- sizeof (类型说明符)
- sizeof 表达式

四、空间复杂度

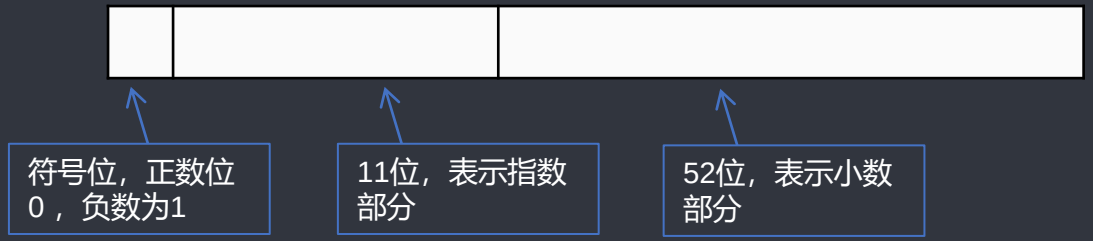
- int 32位二进制



- Long long 64位二进制
- Unsigned 用于修饰 int 或者long long， 将符号位变为数值位

四、空间复杂度

- Double 64位二进制 科学计数法存储 eg：2.33*10^2333



四、空间复杂度

- Char 8位二进制
- Bool 只有0和1 但还是8位二进制

四、空间复杂度

- 计算变量所占的空间
- 8位二进制 = 1个字节
- 1024字节 = lKB
- 1024kB = 1MB
- 1024MB = 1GB……
- 比赛时题目的空间限制位512M，在没有递归等其他消耗下，且只需要开一个int类型的数组，那么这个数组最大可以开到什么数量级？
- 一个长度为10^6的double类型数组占多少空间（说出数量级即可）？

枚举

- 什么是枚举

- 一一列举
- 要点：不重复，不遗漏

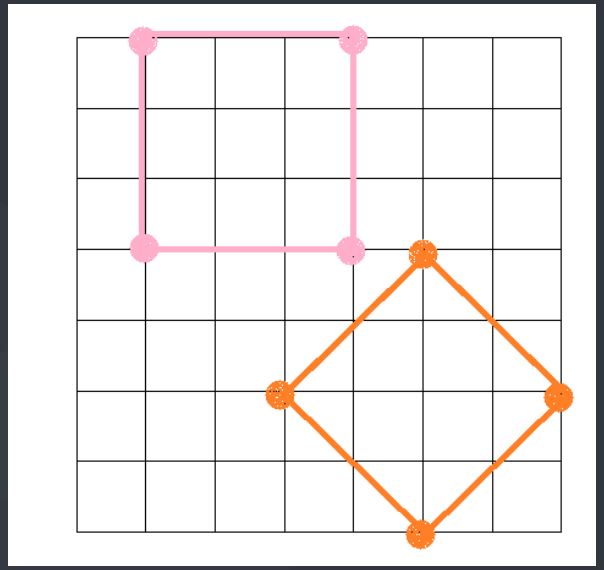
- 优化枚举的基本思路：——减少枚举次数
- 1.选择合适的枚举对象
- 2.选择合适的枚举方向——方便排除非法和不是最优的情况
- 3.选择合适的数据维护方法——转化问题

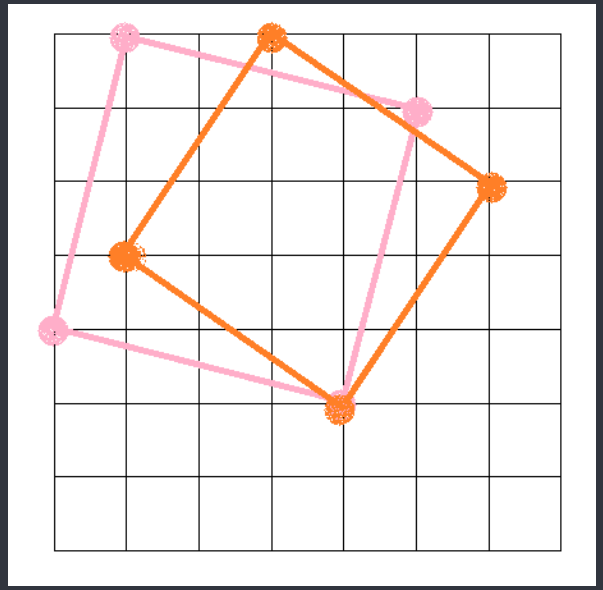
	A	B	C	D	E	F
1	学生信息表					
2	学号	班级	姓名	性别	出生年/月/日	家庭住址
3	001	111	xxx			

例1：最大正方形

- 在一个N*N(N<=100)矩阵中求一个最大的正方形使得该正方形的四个顶点都是有字符“#”构成。
- #*#***
- ******
- #*#*#*
- ******
- #*****
- ***#**

- 几个点能确定一个正方形?





例2：NC16438 回文日期

- 在日常生活中，通过年、月、日这三个要素可以表示出一个唯一确定的日期。
- 牛牛习惯用8位数字表示一个日期，其中，前4位代表年份，接下来2位代表月份，最后2位代表日期。显然：一个日期只有一种表示方法，而两个不同的日期的表示方法不会相同。
- 牛牛认为，一个日期是回文的，当且仅当表示这个日期的8位数字是回文的。现 在，牛牛想知道：在他指定的两个日期之间包含这两个日期本身），有多少个真实存在的日期是回文的。
- 一个8位数字是回文的，当且仅当对于所有的i （ 1 <=i<= 8)从左向右数的第i个 数字和第9-i个数字（即从右向左数的第i个数字）是相同的。

- 例如：
- •对于2016年11月19日，用8位数字20161119表示，它不是回文的。
- •对于2010年1月2日，用8位数字20100102表示，它是回文的。
- •对于2010年10月2日，用8位数字20101002表示，它不是回文的。
- 每一年中都有12个月份：
- 其中，1、3、5、7、8、10、12月每个月有31天；4、6、9、11月每个月有30天；而对于2月，闰年时有29天，平年时有28天。

- 一个年份是闰年当且仅当它满足下列两种情况其中的一种：
- 1.这个年份是4的整数倍，但不是100的整数倍；
- 2.这个年份是400的整数倍。
- 例如：
- •以下几个年份都是闰年：2000、2012、2016。
- •以下几个年份是平年：1900、2011、2014。

- 从0000-00-00到9999-99-99枚举不太合适
- 但是因为日期要满足要求
- 月只会有00-12
- 日只会有 00-31
- 这样其实只枚举后四位看前四位对应之后在不在范围里即可

例3：

- 给定长度为n的整数数列{Ai}，找出两个整数ai和aj（i<j），使得ai-aj尽量大

A_i - A_j,A_j必须是最小的

可以首先做一下得到Min[],使得Min[x]为后面所有数的最小的。

或者看M[]是前面所有的最大的。

这样的两种都是一重for即可

例4：数列求和问题

- 给你一个数列{an}（1≤n≤100000），有q（1≤q≤100000）次询问，每次询问数列的第li个元素到第ri个元素的和。

前缀和

例4：数列求和问题

- 主要的时间复杂度瓶颈在哪里?
- 对区间的查询需要将整个区间扫一遍

- 考虑如何进行转化
- ——将对区间的查询变为对区间端点的查询

例5：数列求和问题

- 我们可以用sum[i]存储前i个数的和，那么sum[i] = sum[i-1] + a[i]，当我们要查询第li个元素到第ri个元素的和时，用sum[ri] - sum[li-1]即可。
- 这样单次查询的复杂度是O(1)的，总复杂度是O(n+q)的。
- 这里的sum[i]是前缀和，前缀和也是算法竞赛当中非常常用的小技巧。

例5：数列修改问题

- 给你一个数列{an}（1≤n≤100000），有q（1≤q≤100000）次修改，每次把数列中的第li到第ri的每个元素都加上一个值ki，求所有的修改之后每个数的值。

例5：数列修改问题

- 主要的时间复杂度瓶颈在哪里?
- 对区间的修改需要将整个区间扫一遍
- 考虑如何进行转化
- ——将对区间的修改变为对区间端点的修改
- 考虑在区间加的过程中有什么值是在区间端点处发生了变化而区间内是没有变化的
- 是每个数与其前一个数的差值!

例5：数列修改问题

- 当我们对第li个到第ri个数加上ki时，第li个数与第li-1个数的差值增加了ki，第ri+1个数与第ri个数的差值减少了ki，而区间内部的相邻两个数的差值是不变的！
- 所以我们可以用数组delta[i]来维护第i个数和其前一个数的差值，（可以默认第一个数前面有一个0），然后当需要将[li,ri]区间的每一个数+ki时，只需要修改delta[li]和delta[ri+1]即可。
- 在所有的修改操作进行完之后，我们再对delta[i]求一次前缀和，就可以得到数列的每个元素的值了。
- 用数组delta[i]来维护第i个数和其前一个数的差值的办法叫做差分。

- 差分和前缀和是一对对称的操作（即对差分数组求前缀和就是原数组，对前缀和求差分也会得到原数组）

扩展与应用： NC16649校门外的树

- 某校大门外长度为L的马路上有一排树，每两棵相邻的树之间的间隔都是1米。我们可以把马路看成一个数轴，马路的一端在数轴0的位置，另一端在L的位置；数轴上的每个整数点，即0,1,2,...,L都种有一棵树。
- 由于马路上有一些区域要用来建地铁。这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。

扩展：校门外的树

- 原题数据范围：$1 \le L \le 10000$和$1 \le M \le 100$

扩展：校门外的树

- 扩大数据范围：$1 \le L \le 100000$和$1 \le M \le 100000$

看数轴上有多少个点统计最后为1的个数

扩展：校门外的树

- 继续增大数据范围：1≤L≤10^9和1≤M≤100000

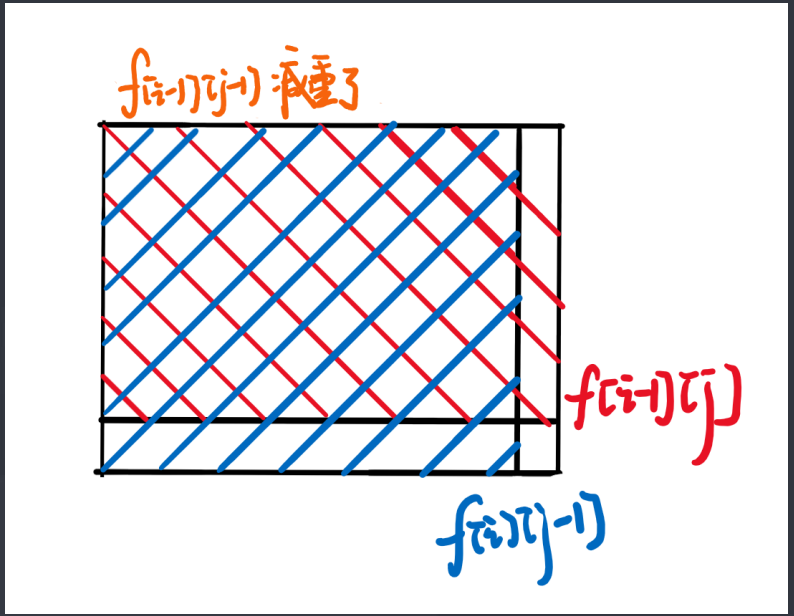
L 太大了数组开不了，
其实和区间有关的点是20万个，
离散化

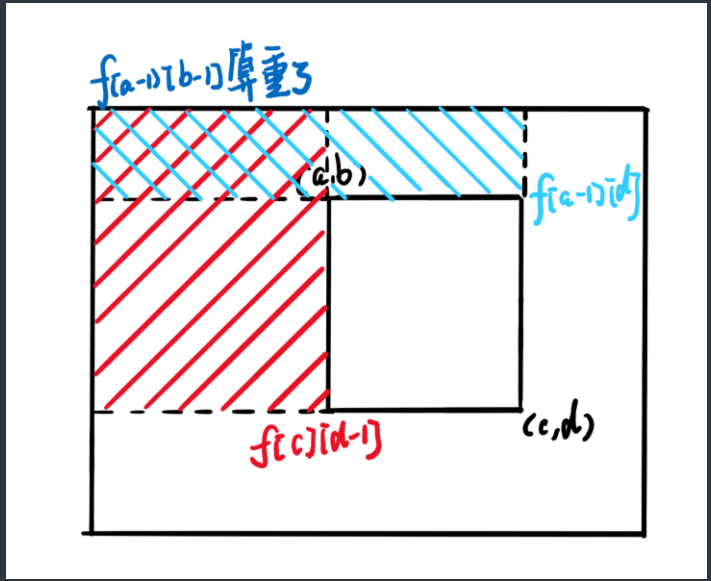
例6： NC20032激光炸弹

- 一种新型的激光炸弹，可以摧毁一个边长为R的正方形内的所有的目标。现在地图上有n(N<=10000)个目标，用整数Xi,Yi(其值在[0,5000])表示目标在地图上的位置，每个目标都有一个价值。激光炸弹的投放是通过卫星定位的，但其有一个缺点，就是其爆破范围，即那个边长为R的正方形的边必须和x，y轴平行。
- 问一颗炸弹最多能炸掉地图上总价值为多少的目标

例6：NC20032激光炸弹

- 怎么把前缀和扩展到二维?

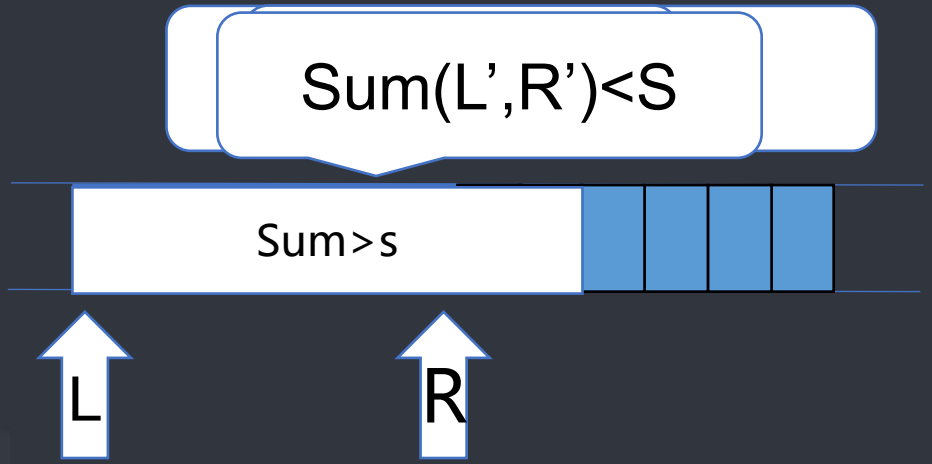




例7：NC107658 poj3061 Subsequence

- 给定长度为n的整数数列以及整数S，求出总和不小于S的连续子串的长度的最小值，如果解不存在，输出0。

尺取



例8： NC18386 字符串

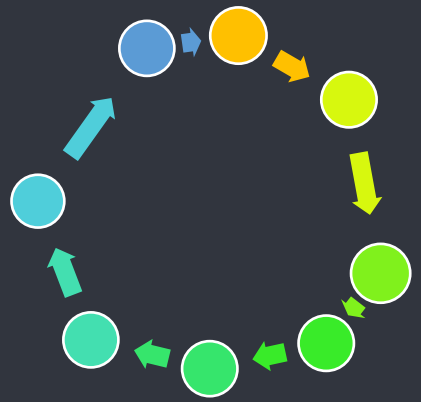
- 小N现在有一个字符串S。他把这这个字符串的所有子串都挑了出来。一个S的子串T是合法的，当且仅当T中包含了所有的小写字母。小N希望知道所有的合法的S的子串中，长度最短是多少。

子串是连续的
左右间已经找齐了，不用再继续找了（右不用再移动了）
左界右移时，右界限不一定右移

例9：NC207040 丢手绢

- 牛客幼儿园的小朋友们围成了一个圆圈准备玩丢手绢的游戏，但是小朋友们太小了，不能围成一个均匀的圆圈，即每个小朋友的间隔可能会不一致。为了大家能够愉快的玩耍，我们需要知道离得最远的两个小朋友离得有多远（如果太远的话牛老师就要来帮忙调整队形啦！）。
- 因为是玩丢手绢，所以小朋友只能沿着圆圈外围跑，所以我们定义两个小朋友的距离为沿着圆圈顺时针走或者逆时针走的最近距离。
- $2 \le N \le 100000$

也是尺取，第一次i和j找到之后，j不往回走



例10：NC20241 [SCOI2005]扫雷MINE

- 相信大家都玩过扫雷的游戏。那是在一个n*m的矩阵里面有一些雷，要你根据一些信息找出雷来。
- 万圣节到了，“余”人国流行起了一种简单的扫雷游戏，这个游戏规则和扫雷一样，如果某个格子没有雷，那么它里面的数字 表示和它8连通的格子里面雷的数目。
- 现在棋盘是n×2的，第一列里面某些格子是雷，而第二列没有雷，如下图： 由于第一列的雷可能有多种方案满足第二列的数的限制，你的任务即根据第二列的信息确定第一列雷有多少种摆放方案。
- N<=100000

- 因为第二列的情况已经确定了，那么我们只需要枚举第一列的第1个雷放还是不放，然后根据第二列第一个数就能确定第二列第二个位置放还是不放，依此类推，直到确定最后一行的情况，或者发现矛盾为止（方案数最多等于2，只要第一个格子确定了放不放雷，之后的就确定了）。

例11：NC 106350 Flip Game

- 有一个4*4的灯泡，每次按某一个点可以使得其本身以及其上下左右共五个的灯的开关反向。给定初始状态（每个灯泡的亮或者灭），问：能否把所有灯都灭掉?

加强

- 有一个N*M的灯泡，(N<=10,M<=100)，每次按某一个点可以使得其本身以及其上下左右共五个的灯的开关反向。给定初始状态（每个灯泡的亮或者灭），问：能否把所有灯都灭掉?

位运算介绍

- <<左移
- >>右移
- |或
- &与
- ~取反
- ^异或
- 请一定注意位运算优先级问题！！！
- 请一定注意位运算优先级问题！！！
- 请一定注意位运算优先级问题！！！

5	* / %	乘法/除法/取余
6	+ -	加号/减号
7	<< >>	位左移/位右移
8	< <=	小于/小于等于
	> >=	大于/大于等于
9	== !=	等于/不等于
10	&	按位与
11	^	按位异或
12	|	按位或
13	&&	与运算
14	||	或运算
15	?:	三目运算符

- a=a ^ b;
  b=a ^ b;
  a=a ^ b;
- 实现了什么功能?

位运算基础

- 去掉最后一位
- x >> 1
- 在最后加一个0
- x << 1
- 在最后加一个1
- (x<< 1)+1
- 把最后一位变成1
- x | 1
- 把最后一位变成0
- (x | 1) - 1
- 最后一位取反
- x ^ 1
- 把右数第k位变成1
- x | (1 << (k-1))
- 把右数第k位变成0
- x & (~ (1 << (k-1)))
- 右数第k位取反
- x ^ (1 < < (k-1))

贪心

什么是贪心?

- 贪心算法（又称贪婪算法）是指：在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部最优解。
- 能够使用贪心算法的问题都是能严格证明贪心出的局部最优解就是所求的全局最优解的。

- 人类的本能——每次都选看起来最好的!

例1：排队接水

- 有n个人在一个水龙头前排队接水，假如每个人接水的时间为Ti，请编程找出这n个人排队的一种顺序，使得n个人的平均等待时间最小。
- n<=1000，ti<=1e6

例2： NC16783 拼数

- 设有n个正整数，将它们连接成一排，组成一个最大的多位整数。
- 例如：n=3时，3个整数13，312，343，连成的最大整数为34331213
- 又如：n=4时，4个整数7，13，4，246，连成的最大整数为7424613
- (n<=1000，每个数都在int范围内)

例3：区间覆盖（工作安排）

- 在0到L的数轴上有n个区间[li,ri]，现在需要你选出其中尽量多个区间，使得其两两不相交。（n<=100000）

区间编号	0	1	2	3	4	5	6	7	8	9	10	11
li	1	3	0	3	2	5	6	4	10	8	15	15
ri	3	4	7	8	9	10	12	14	15	18	19	20

例4：活动安排

- 给n个活动，每个活动需要一段时间Ci来完成，并且有一个截止时间Di，当完成时间ti大于截止时间完成时，会扣除ti-Di分，让你找出如何使自己所扣分的最大值最小。（n<=100000）

例5： NC 16561国王的游戏

- 恰逢 H 国国庆，国王邀请 n 位大臣来玩一个有奖游戏。首先，他让每个大臣在左、右手上面分别写下一个整数，国王自己也在左、右手上各写一个整数。然后，让这 n 位大臣排成一排，国王站在队伍的最前面。排好队后，所有的大臣都会获得国王奖赏的若干金币，每位大臣获得的金币数分别是：排在该大臣前面的所有人的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。

例5：NC 16561国王的游戏

- 国王不希望某一个大臣获得特别多的奖赏，所以他想请你帮他重新安排一下队伍的顺序，使得获得奖赏最多的大臣，所获奖赏尽可能的少。注意，国王的位置始终在队伍的最前面。
- 求获得奖赏最多的大臣获得的奖赏。
- $1 \le n \le 1,000$，$0 < a,b < 10000$。

- 交换相邻两个人的顺序不影响前面的人的钱数也不影响后面人的钱数

例6：NC25043 Protecting the Flower

- 农夫有n头牛跑到花田上吃草，农夫要把它们送回自己的牛舍，所化的时间分别为 t_i，（单程时间为t_i），每头牛留在草坪上每花田间内吃花量分别为d_i。花田上花最少被破坏的数量为多少?

- 首先我们知道，交换相邻两头牛的顺序既不影响前面的牛破坏的花的数量也不影响后面的牛破坏的花的数量。和上题一样的过程：

例7： NC 200190 矩阵消除游戏

- 牛妹在玩一个名为矩阵消除的游戏，矩阵的大小是n行m列，第i行第j列的单元格的权值为ai,j，牛妹可以进行k个回合的游戏，在每个回合，牛妹可以选择一行或者选择一列，然后将这一行或者这一列的所有单元格中的权值变为0，同时牛妹的分数会加上这一行或者这一列中的所有单元格的权值的和。
- 牛妹想最大化她的得分，球球你帮帮她吧！
- $1 \le n,m \le 15$
- $1 \le a_{i,j} \le 1e6$
- $1 \le k \le n*m$

例8：NC18979 毒瘤xor

- 小a有N个数a1, a2, ..., aN，给出q个询问，每次询问给出区间[L, R]，现在请你找到一个数X，使得
  1、0⩽X<2^31
  2、$\sum_{i=L}^{R} X \oplus a[i]$ 最大，⊕表示异或操作
- n, q ≤ 105

- 我们知道异或运算的规律是：异或1取反，异或0不变，那么对于本题如果原来的数这一位是1，那我们肯定希望它异或一个0；如果原来是0，肯定希望异或1。
- 那么我们只需要求出[l,r]这个区间里面的数在每一位是1多还是0多，1多那么x的这一位就是0，0多x的这一位就是1。
- 求[l,r]区间里面每一位有几个1只需要用前缀和维护即可，即求出前i个数每一位又几个1，然后相减。

例9：NC17857 起床困难综合症

- 21 世纪，许多人得了一种奇怪的病：起床困难综合症，其临床表现为：起床难，起床后精神不佳。作为一名青春阳光好少年，atm 一直坚持与起床困难综合症作斗争。通过研究相关文献，他找到了该病的发病原因：在深邃的太平洋海底中，出现了一条名为 drd 的巨龙，它掌握着睡眠之精髓，能随意延长大家的睡眠时间。正是由于 drd 的活动，起床困难综合症愈演愈烈，以惊人的速度在世界上传播。为了彻底消灭这种病，atm 决定前往海底，消灭这条恶龙。

  历经千辛万苦，atm 终于来到了 drd 所在的地方，准备与其展开艰苦卓绝的战斗。drd 有着十分特殊的技能，他的防御战线能够使用一定的运算来改变他受到的伤害。具体说来，drd 的防御战线由 n 扇防御门组成。每扇防御门包括一个运算 op 和一个参数 t，其中运算一定是 OR,XOR,AND 中的一种，参数则一定为非负整数。如果还未通过防御门时攻击力为 x，则其通过这扇防御门后攻击力将变为 x op t。最终drd 受到的伤害为对方初始攻击力 x 依次经过所有 n 扇防御门后转变得到的攻击力。

- 由于atm 水平有限，他的初始攻击力只能为 0 到 m 之间的一个整数（即他的初始攻击力只能在 0, 1, ... , m 中任选，但在通过防御门之后的攻击力不受 m 的限制）。为了节省体力，他希望通过选择合适的初始攻击力使得他的攻击能让 drd受到最大的伤害，请你帮他计算一下，他的一次攻击最多能使 drd 受到多少伤害。

测试点编号	n, m 的规模	约定	备注
1	$2 \le n \le 100, m = 0$	$0 \le t \le 10^9$ op 一定为 OR, XOR, AND 中的一种	
2	$2 \le n \le 1{,}000$ $1 \le m \le 1{,}000$		
3			
4	$2 \le n, m \le 10^5$		存在一扇防御门为 AND 0
5			所有防御门的操作均相同
6			
7	$2 \le n \le 10^5$ $2 \le m \le 10^9$		所有防御门的操作均相同
8			
9			
10			

- 位运算是可以一位一位独立讨论的，且这个题3类操作对于每一个二进制位都是独立影响的——所以我们从最高位开始看——
  如果当前位填上0在运算之后能变成1，那么就填0；
  否则如果当前位填上1在运算之后能变成1，且当前位填1没有超过范围那么就填1（如果填01都能变1，显然还是应该填0这样给后面更多机会）；
  如果都不能变成1或者当前位填1超出范围就还是填0。
  这个判断也并不需要一位一位每次都去做全部操作，我们只需要把000...000和111...111拿去做一次操作然后看对应的位的情况即可。

- 作业题单
- https://ac.nowcoder.com/acm/problem/collection/1195
- 先做例题

- **先做例题，剩下的题按以下顺序：**
- 铺地毯
- 「土」秘法地震
- 牛牛的木板
- Subsequence
- 数学考试
- 糖糖别胡说，我真的不是签到题目
- 排座椅
- Quasi Binary
- 奇♂妙拆分
- Protecting the Flowers
- codeJan与旅行
- 「土」巨石滚滚