## A

题意：对一颗树，有操作A x k，将x的点权+k，x的孩子的点权+k+1，x的孩子的孩子的点权+k+2…给定树的形状，及所有的操作。求树的根节点，使得整棵树的点权和最小。

将操作A x k分为两部分，x的子树全部+k，x的子树中每个节点加上自身到x的距离。对每个节点记录四个值，num[u]对u操作的次数，f[u]对u操作一次整棵子树加的第二部分的和，sz[u]子树u的大小，sum[u]对u操作的和。两次dfs即可。

## B

题意：有一颗树，每个节点有一个取值区间[li, ri]，要为每个节点赋一个值Fi（li <= Fi <= ri），使得任意相邻的节点互素。然后对每个节点统计Fi在所有可能中的和。

设dp1[u][i]表示u的子树都合法，且F[u]=i的方案数。

dp2[u][i]表示u的子树都合法，且F[u]是i的倍数的方案数。，

显然dp2[u][i] = sum(dp1[u][j]) (i | j)

对于每一个u，dp1[u][i] = dp1[u][i] \* sum( u[j] \* dp2[u][j]) j | i

其中u[i]是莫比乌斯反演系数。就是容斥一下 ==

跑两遍dfs即可。

## C

题意：给定n个节点组成的树，1为敌方基地，叶子结点为我方结点。我们可以在每个结点安放炮台，至多放一个炮台，然后就可以打炮，每个结点有ki种炮，每种炮有一个花费和一个能量（能量对应着打掉敌人多少hp，敌人hp<=0就会死掉）。敌人可能往一个结点的每条分支跑，并且路径不可预测，所以要想保证守住阵地，就要使得敌人走任意路径都打不到阵地。安放炮台的花费不能超过m，问怎么安放炮台，才能使得防御能力最强。

设dp[u][i]为考虑u的子树，花费为i的情况下，最强的防御能力是多少。

Dp[u][i]等于u的子节点v中最弱的防御能力加上u自身的防御能力

dp[u][i] = max( min(dp[v][vj]) )

树上转移一下就好。

## D

题意：给定S和m，对每一个k，有多少长度为m的DNA序列T满足LCS(S, T) = k。|S| <= 15。

考虑dpi,j 表示S的前i个和T 的前j个的LCT。 注意到，如果我们枚举了一个长度为i的串，我们关心哪些信息？有意义的只有dpi,∗的值！对于i，我们暴力记录所有dpi,∗的值来。注意到dpi,j 和dpi,j−1最多差1，因此状态数量很少。

## E

题意：给出n个数每个数以2^ai \* 3^bi的形式给出，求这n个数所有的子集，并求子集的lcm的和。

对于一个选出的集合，它的lcm就等于2^max(ai) \* 3^max(bi)，那么，将ai从小到大排序，依次枚举，那么当前数和前面数组成的集合中，lcm的结果的2的幂次一定等于当前枚举的幂次，剩下的问题就是如何计算子集数和3的幂次。将每个数按bi的大小编号，每次要将这个数插入到其对应编号的位置，那么，我们就可以发现，当前要插入的数的位置，前面的数，3的幂次都小于等于它，后面的数，3的幂次都大于等于它。因此，对于前面的数，3的幂次就是当前枚举的幂次，后面的数，3的幂次是它们原来的幂次。这样，用线段树维护两个值，一个值是区间内的子集数，另一个数各子集的3的幂次的最大值的和。

## F

题意：n个点m条边的有向图，求不同拓扑序的个数。n<=40,m<=20

因为m<=20，所以最大的连通块有21个点，对每一个连通块状压，然后求个组合数即可。

## G

题意：有N个人，K家银行，Q块金币，现在要去抢银行，假设抢第i家银行，派了p个人，花了d块金币，可以抢到F [i,p, d]块金币，F[i,p,d]通过以下式子递推。

f [i, p, d] =0 (p <= 0 or d <= 0)

f [i, p, d]= f [i, p - 1, d - e i] + f [i, p - 1, d] (1 < p <= N,1 <= d <= Q)

and f [i, 1, d] comes from

f [i, 1, d] = A[i] \* f[i, 1, d - 1] ^ 2 + B[i] \* f[i, 1, d - 1] ^ 2 + C[i] (1 <= d <= Q)

假设抢一家银行总共抢到x块金币，你可以分到(x/(p+1)) % M块金币。

人可以重复抢银行，但是Q个金币不可以重复使用。求你最多可以得到多少金币。

直接裸dp会T。但是注意到f [i, p, d]= f [i, p - 1, d - e i] + f [i, p - 1, d]这个递推式，我们可以转化为f [i, p, d] = x[1] \* f[i, 1, 1] + x[2] \* f[i, 1, 2] + … + x[Q] \* f[i, 1, Q]。显然，x[i]都是比较小的数，对于每一个i我们可以先预处理出x[i]数组，这样就可以把求f[i, p, d]的时间优化掉，就可以过了。

## H

题意：给一颗有边权的树，用简单路径去覆盖树上的所有边，每条路径的花费是边权和+K，求最小花费。

考虑任意一条边，被覆盖次数最多为2。因为若有三条路径穿过这条边，那么可以通过路径的交换来使得往上的路径变成1条。

Dp[u][0/1/2]表示从u向上走的边分别为0/1/2所需要的最小花费，考虑到所有情况转移即可。

## I

题意：一颗左右儿子必须同时存在的二叉树，叶子节点有点权。树中任意一个点，左子树的权值和必须等于右子树的权值和。给一个序列，求最长的子序列，使得该子序列满足上述条件的某棵树的中序遍历。

满足条件的子序列满足c\*2^k，枚举c，分别处理。Dp[i][s]表示处理到第i位，已选数的和为s，可以发现新添加的元素必须满足条件s & (-s) >= k，依次转移即可。

## J

题意：求区间[A,B]内有多少个数x满足x%f(x)=0，其中f(x)为x的数位和。

枚举数位和s，预处理dp[i][j][k]表示长度为i，数位和为，j模s为k的数有多少个。按位DP即可。

## K

题意：有一个栈，最开始没东西，现在过来了很多2,4,8,16，路过每个值的时候你可以将其加入栈中，并得到这么多的分数，如果栈顶的两个值相同，他们会合并成一个新的值，为他们的和，并得到这么多的分数。问你如何选择能使分数最大

会发现若要栈内元素合并，有效部分一定是栈最后那部分单调递减的后缀，所以只需维护单调递减后缀即可。Dp[i][s]表示处理到第i个数，单调递减后缀和为s，新添加的元素k必须满足s & (-s) >= k，否则清空栈。依次转移即可。

## L

题意：有n个房间，由n-1条隧道连通起来，实际上就形成了一棵树，从结点1出发，开始走，在每个结点i都有3种可能： 1.被杀死，回到结点1处（概率为ki）2.找到出口，走出迷宫 （概率为ei）3.和该点相连有m条边，随机走一条。 求走出迷宫所要走的边数的期望值。

设 E[i]表示在结点i处，要走出迷宫所要走的边数的期望。E[1]即为所求。

叶子结点： E[i] = ki\*E[1] + ei\*0 + (1-ki-ei)\*(E[father[i]] + 1)

= ki\*E[1] + (1-ki-ei)\*E[father[i]] + (1-ki-ei);

非叶子结点：（m为与结点相连的边数）

E[i] = ki\*E[1] + ei\*0 + (1-ki-ei)/m\*( E[father[i]]+1 + ∑( E[child[i]]+1 ) )

= ki\*E[1] + (1-ki-ei)/m\*E[father[i]] + (1-ki-ei)/m\*∑(E[child[i]]) + (1-ki-ei);

设对每个结点：E[i] = Ai\*E[1] + Bi\*E[father[i]] + Ci;

从叶子节点不断向上代入即可。

## M

题意：有n个人排队等着在官网上激活游戏。Tomato排在第m个。

对于队列中的第一个人。有一下情况：

1、激活失败，留在队列中等待下一次激活（概率为p1)

2、失去连接，出队列，然后排在队列的最后（概率为p2）

3、激活成功，离开队列（概率为p3）

4、服务器瘫痪，服务器停止激活，所有人都无法激活了。

求服务器瘫痪时Tomato在队列中的位置<=k的概率

设dp[i][j]表示i个人排队,Tomato排在第j个位置，达到目标状态的概率(j<=i)。dp[n][m]就是所求

j==1: dp[i][1]=p1\*dp[i][1]+p2\*dp[i][i]+p4;

2<=j<=k: dp[i][j]=p1\*dp[i][j]+p2\*dp[i][j-1]+p3\*dp[i-1][j-1]+p4;

k<j<=i: dp[i][j]=p1\*dp[i][j]+p2\*dp[i][j-1]+p3\*dp[i-1][j-1];

其中:

p=p2/(1-p1); p31=p3/(1-p1); p41=p4/(1-p1)

若第i-1层的dp值已经全部求出，那么第i层的值dp[i][j]构成一个环，设dp[i][1] = a x + b，扫一圈最后可求出x。继而求出所有的dp[i][j]。依次按照i从小到大递推即可。

## N

题意：有n个怪兽，每个怪兽都有一个破坏值di，奥特曼初始的体力值H=2047，每次奥特曼都随机的选择一个小怪兽并打死，打死之后体力值变为H&di，H=0的时候奥特曼就挂掉了，求奥特曼至少打死K只小怪兽的概率

dp[i][j][k]为考虑前i个怪兽，打死了j只怪兽，体力值为k，的方案数。枚举当前的体力值k>0，然后再枚举一个小怪兽a[l]，如果k&a[l]=0，就当做最后一个打死的小怪兽为l。注意若之前已经打死了a[l]，k&a[l]=k!=0。

## O

题意：求区间[l, r]内有多少数x满足，x可以被x的十进制中每一个非0位整除。

Dp[i][s][k]表示长度为i，出现过的数位集合为s，模2520为k的数有多少个。按位dp即可。2520是1~9的最小公倍数，知道了模2520的值即可知道模1~9任意一个数的值。

## P

题意：给一个只有0 2 4的序列，0必须修改为2或4，然后从左端开始合并，相同的两个数合并成一个数，且值为两个数的和。问有多少种方法使得最后生成的序列中存在>=2^k的数。

Dp[i][s]表示处理到第i位，已选数的和为s，可以发现新添加的元素必须满足条件s & (-s) >= k，否则s情况。依次转移即可。

## Q

题意：给一个字符串s，只包含前20个小写字母。Q个查询，每次查询给定k<=5个字母，求有多少个子串满足这k个字母出现的次数均为偶数。

高维前缀和

**for**(int i=1;i<=n;i++) {

now ^= 1 << (s[i] - 'a');

cnt[now]++;

}

**for**(int j = 0; j < 20; j++)

**for**(int i = 0; i < (1 << 20); i++)

**if**(!(i >> j & 1)) {

cnt[i ^ (1 << j)] += cnt[i];

}

cnt[i]表示sum(num[j])，其中i & j = j

2^k枚举并容斥求出每一种情况的个数，统计答案即可。

## R

题意：n个人要比赛，第i个人的成绩在[li, ri]内均匀分布，求p[i][j]，第i个人排在第j位的概率。

将区间离散化。若有k个人在同一个区间内，任意一个人排在任意一个位置的概率均为1/k。

枚举某一个在某一个区间，求出dp[i][j]为有i个人在这个区间之前、有j个人在这个区间之后、有n-i-j个人和这个人在同一个区间的概率，然后枚举这个人在区间内的排位，累计答案即可。

## S

题意：称只包含4和7的数为lucky数，现在给两个长度相同的lucky数l和r，令中间的lucky数分别为a1, a2, … , an。求a1·a2 + a2·a3 + ... + an - 1·an

设dp[i]为长度为i的lucy数的答案。长度为i+1的lucky数可以写成4\*10^i + ai或7\*10^i+ai，将式子拆开即可。

统计[l, r]之间的答案，按位dp即可。

## T

题意：一个公司获得了一个厂房n(10^5)天的使用权和一笔启动资金C(10^9)，准备在n天里租借机器生产来获得收益，可以租借的机器有M(10^5)个，每个机器有四个值，D,P,R,G (D<=n, P,R,G都是10^9)，表明你可以在第D天花费P费用（首先手里必须有那么多钱）租借这个机器，从D+1天开始该机器每天产生G的收益，在你不需要机器时可以卖掉这个机器，一次获得R的钱。需要注意的是：厂房里只能停留一台机器；不能再购买和卖出机器的那天操作机器，但是可以再同一天卖掉一台机器再买一台；在第n+1天，必须卖掉手里的机器；问n+1天后能获得的最大资金

将机器按照D排序，f[i]表示在D[i]时刻卖掉手里的机器手里最多多少钱。

f[k] = max(f[j] - P[j] + R[j] + G[j] \* (D[k] - D[j] - 1))。其中f[j] >= P[j]

设f[i] – P[i] + R[i] + G[i] \* (D[k] – D[i] – 1) > f[j] - P[j] + R[j] + G[j] \* (D[k] - D[j] - 1)

令h[i] = -(f[i] – P[i] + R[i] – G[i] \* D[i] – G[i])，则

h[j] – h[i] > D[k] \* (G[j] – G[i])

显然是斜率优化的模型，但是G[i]并不具有单调性，用cdq分治，分治之后左端按照G[i]排序，就OK了。

## U

题意：一个只包含X, B, W的字符串，要把X替换成B或W，要求替换之后，有一段长度为k的子串s1全部由B组成，且有一端长度为k的子串s2全部由W组成，且s1在s2之前。

设dp[i][0]为处理到第i个串，s1还未曾出现的方案数

dp[i][1]为处理到第i个串，s1出现过，s2还未曾出现的方案数

dp[i][2]为s1和s2都出现且s1在s2之前的方案数。

dp[n][2]即为要求的答案

转移的时候，发现总能求出dp[i][0/1/2]中的其中两个，又因为三个的总方案数是确定的，所以三个都可以确定。

## V

题意：有n个人站成一排，要把他们分成连续的m段，在同一段的任意两个人i和j，都有一个陌生值，求所有陌生值的和的最小值。

设dp[k][t]为将前k个人分为t段的最优值。则有

dp[k][t] = min( dp[i][t – 1] + w[i][k] ) (i < k)

设dp[i][t – 1] + w[i][k] < dp[j][t – 1] + w[j][k] (i < j)

则(dp[i][t – 1] – dp[j][t – 1]) < w[j][k] – w[i][k]

当i<j满足上式时，i要比j更优，否则j更优。

上式左端是一个定值，右端和k相关，且随着k的增大，w[j][k]-w[i][k]不断变小。

所以我们二分一下分界点，在分界点之前i值更优，在分界点之后j值更优。

根据分界点维护一个单调队列优化即可。

## W

题意：一条直线上有n个点，每两个相邻点之间的距离给出。有m只小鸡，第i只小鸡在第h[i]个点玩，在t[i]时刻玩完了开始在h[i]这个点等饲养员来带走。共有p个饲养员，初始时都在1号点。请安排饲养员的出发时间，使得所有小鸡等待的时间和最小。

对于每只小鸡，可以计算出等待时间正好为0时饲养员的出发时间d[i]。饲养员出发的时间一定是某一个d[i]。将所有的d[i]按照从小到大排序。饲养员领取的小鸡的d[i]一定是连续的。

设dp[k][t]为第t个饲养员在d[k]时刻出发，则有

dp[k][t] = min( dp[i][t – 1] + d[k] \* f[i][k] – sum[k] + sum[i]) i < k

一个显然的的斜率优化模型。

## X

题意：给出一个有且仅有一个环的图, 顶点数<= 10^5, 环的大小在3~30之间, 现在初始的时候随机选一个点作为起点, 然后每次选择与当前所在的点相邻的没有走过的点进入, 直到不能走为止, 求最终停在各个点的概率, 输出其中最大的5个的和

很恶心的树形DP。

f[u] 从u的子树（不包括u）走到u的概率。

然后求出从环上子树走到环上某一点，然后开始绕环走，走到环上某个点的概率。

最后求从环上绕到某一个点，然后走下去的概率。

终止点只可能是叶子节点和环上度为2的点。

## Y

题意：有26种错误，有c（1000）个限制条件t k，就是说第t种错误如果犯了k的整数倍次，就不会接受惩罚。要求正好犯n（10^18）个错误，并且不被惩罚的方案数。犯错的不同顺序视为不同方案。所有k的乘积不超过123

显然，没有限制的错误是不能犯的。k=1的错误随便犯。

设dp[i][j]为已经犯了i个错误，对每个k!=1的限制条件当前的模k的模数，因为k的乘积不超过123个，所以j的个数不超过123个，写出转移矩阵，用矩阵快速幂转移即可。

## Z

题意：一个图，有k<=8个景点（景点可能有多个入口），从1号点入，且从1号点出，每个景点可能在某些点得到钥匙，有钥匙参观景点有一个时间花费（较小），没有钥匙有另一个时间花费（较大），问参观完所有景点需要的最短时间。

Dp[u][s1][s2]表示，当前在u节点，拿到的钥匙集合为s1，参观过的景点集合为s2。建图跑最短路，或者递推转移。Dp[u][s1][s2]有两个状态转移方式，参考u内的某一个景点，或者走到另一个点，拿到能拿的所有钥匙并参观完所有有钥匙的景点。