A、坎巴拉奶牛天空计划

二分+计算几何

若飞行器在t与火星相遇那么在大于t的时间必定会相遇。假设相遇时间为t那么可以算出

火星此刻的坐标定为s，若飞行器能在t时间内到达s点那么最快时间小于t，若不能到达s则在大于t的时间中。在小于t时间的情况到达s点，那么最快的方法分两种情况，一种是飞行器与s在一侧这样直接走直线最快，若不在一侧则路线为飞行器起点与不可进入区域半径为r的圆切线加上s与该园的切线再加上两个切点所对应的圆弧长度。最后再二分搜索一下时间就行了

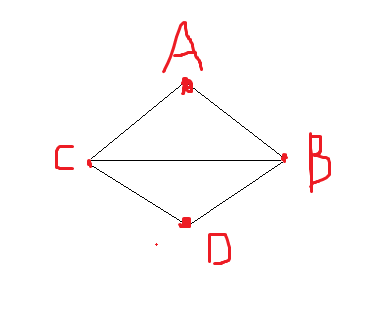
B、奶牛学算术

X转换为二进制，如果满足等式，则在X为1的位上，Y（二进制）只能为0，X为1的位上Y可以为0或1，所以只要一遍对X做进制转换(不存)一边判断为1还是0，然后Y填下去直到第k个。

C、奶牛之路

显然当x为大于1的整数时满足x^i>x^0+x^1+x^2+...+x^(i-1).因此只需要从大到小

枚举边，判断能不能删就行了。情况1：如果是三个点构成的三角形则删除权值最大那条边是没问题的。情况2：如下图



考虑BC能否被删，如果边BD被删了，那么BC就成为连接B和D消耗为2的一条路中的边，因此路能否被删需要考虑每次删除后的新图中，对已被删除的边（u，v）统计u和v两点能同时连向的第三个点的数目，如果只剩1个，那么该点与u，v的连接的路是不能删的。具体实现就用一个二维数组a[v][u]表示v和u同时连向的第三个点的数目，每删一条路就更新一下数组就行。

D、奶牛集装箱I

贪心。

考虑相邻的两个集装箱，i和i+1(i在上面)，如果这两个集装箱交换，那么对于其他集装箱是没有影响的。如果i和i+1不交换则有

i的超重：W1+W2+…+Wi-1 –Ci ①

i+1的超重：W1+W2+…+Wi-1 +Wi–Ci+1②

交换i和i+1

i的超重：W1+W2+…+Wi-1 +Wi-Ci③

i+1的超重: W1+W2+…+Wi-1-Ci+1④

明显②>③，④>①假设原顺序是最优的，那么后来交换后肯定不是最优的，即②<④，即Wi+Ci<Wi+1+Ci+1，即最优序列一定满足Wi+Ci<Wi+1+Ci+1。因此按照Wi+Ci<Wi+1+Ci+1排序即可。

E、奶牛集装箱II

动态规划斜率优化。

b[n]=0,那么我们的目的就是用最小花费撞开第n个集装箱。定义dp[i]，表示把第i个集装箱撞开时，最小的花费。则有dp[i]=dp[j]+b[j]\*a[i]，j<=i，取最小值。这样复杂度是n^2数据量是10^5，明显会超时，优化一下。把上面的方程变换一下成dp[j]=dp[i]-a[i]\*b[j].在j变化的时候i是不变的，因此方程可以看成一个斜率为-a[i]的直线且过点(b[j],dp[j])，dp[i]为直线的纵截距。因此问题转换为求截距最小。设j1，j2为j<=i的其中两个值。要比较dp[j1]=dp[i]-a[i]\*b[j1]和dp[j2]=dp[i]-a[i]\*b[j2]哪个求出的dp[i]更大。两式化简如果(dp[j2]-dp[j1])/(b[j1]-b[j2])<a[i].则方案j2优于j1，这样就可以删除j1方案，永远都不会再次出现当中，最后再用单调队列维护一下即可，复杂度优化到O(n)。

F、奶牛吃金币

动态规划。

dp[i][j][k]表示走了i步第一个牛在j行第二个牛在k行。可以通过走的步数和行算出列。

转移方程就是dp[i][j][k]=max(dp[i-1][j][k]+t,dp[i-1][j-1][k-1]+t,dp[i-1][j-1][k]+t,dp[i-1][j][k-1]+t).

分别表示两牛都是从左往右走，两个都是从上往下走，一个从上往下一个从左往右，一个从左往右一个从上往下。其中t是走第i步吃到的金币，通过j，k算出两头牛的坐标，并计算下一步吃到的金币并且相加，如果j=k表明第i步两头牛走到同一个格子，这样只算一个格子的金币而不是相加。

G、奶牛之夜

动态规划

灯的数量只有20,状态压缩 用dp[i]表示i状态时走的距离 i的二进制形式中1表示对应灯亮,最后求出dp[(1<<n)-1]即所有灯全亮 。状态转移 dp[i] = max( dp[i] , dp[j] + distance(j->i) )。算一下照射到地面的长度。

H、奶牛的超十进制算术？

水题不解释