Contest 4 by Digimon

Summary

Problem	AC	WA	PE	RTE	FPE	SF	TLE	MLE	OLE	CE	Submit
Agumon	0	8	0	0	0	0	13	2	0	0	23
Gabumon	6	16	0	1	0	0	0	0	0	0	23
Gomamon	24	82	0	0	0	0	4	0	0	2	112
Palmon	16	43	0	0	0	3	5	0	0	0	67
Piyomon	3	10	0	0	6	0	0	0	0	1	20
Tentomon	0	0	0	0	0	0	0	0	0	0	0
Summary	49	159	0	1	6	3	22	2	0	3	245

Acceleration

- 首先，该题有一个这样的性质，在保证最优的情况下，加速次数只能是0,1,2 .. n，或者是无穷次。
- 证明如下，对于两个加速点a，b，a为距b最近的加速点，假设最优方案要经过b的话。
- ab间一次来回，需要花费(3/4)*(2*d(a,b))的时间，但是速度翻了4倍，于是b到终点时间减少了3/4。

- 所以：
- 若ab距离的两倍小于等于b到终点最短距离，则在ab间无限加速更快。时间收敛到ab距离的两倍除以b点在开始绕之前的速度，之后瞬间到终点(无限速度）
- 若ab距离的两倍大于b到终点的最短距离，则选择不绕，直接走向终点。

- 1、对于无穷的处理：首先我们对全图做一次floyd，加速点不作为中间节点。就能求出加速点之间的距离（普通点可以忽略）。然后对于每一个加速点i，找到一个与他最近的加速点，距离存为d[i]。
- 2、分层图SPFA：将每个点按加速次数分成若干个点（加速0次，1次……w次），然后正向做一次SPFA，求出每个点在每个加速次数下的最小到达时间。
- 3、算出到终点的最小时间以及加速次数：对于每个加速点，判断一下是直接到终点快还是无限加速(2*d[i]/v)到终点快，在这个基础下再得出最大加速次数。

- 可以注意到，当加速次数超过57次的话，已经超越的double的精度，而SPFA中需要处理的加速次数最高为99次，所以我们不能用double来存储最小时间，而是用3个long long来存储，这样精度就能达到小数点后120位，可以精确地算出到达每个点的时间。
- 时限开成 15s 是考虑到可能有人用 JAVA 的 BigInteger 或 BigDecimal 来计算这题，而 192 上的 JAVA 是非常慢的。

Fetus

- 题意：一个10*20的格子，每个格子里有一个字符，代表的意义分别是:
  - '.'：空格子
  - 'X'：实心砖块
  - 'F'：安全终端，收集齐所有安全终端才能打开终点的门
  - 'S'：起点
  - 'T'：终点
  - 'G'：重力反向装置

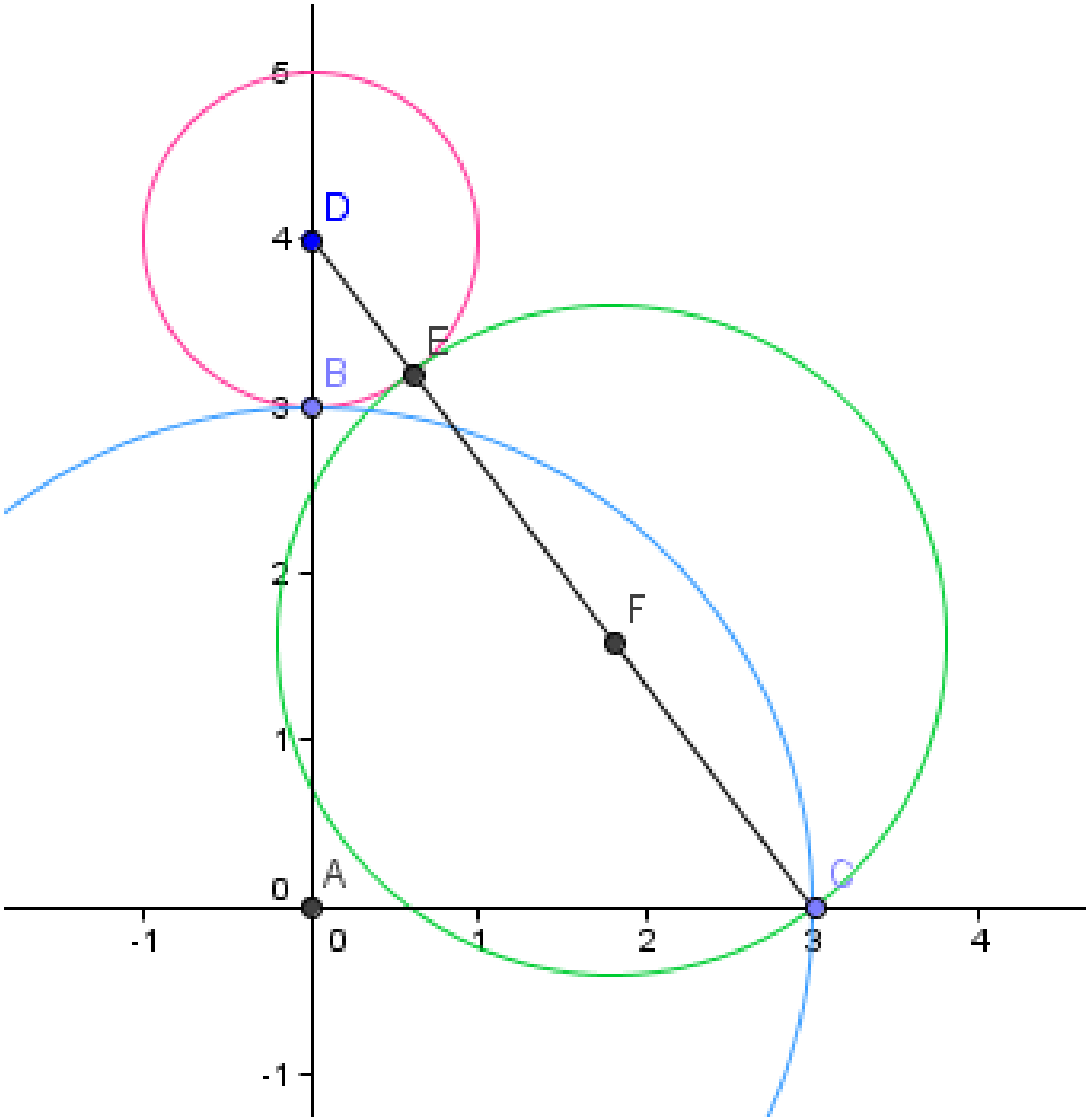
- 格子上下左右都是通的
- 重力方向判断：下面有砖头就是向下，否则向上，注意起点在第1行，上面一格在最后一行，起点在最后一行，下面一格在第1行
- 砖头所在的格子不可进入，其他格子均可进入
- 每一秒，脚下有砖头才能左右移动，否则只能上下掉落

- 一共有200格， 最多有3个terminal，用3位二进制可以表示
- f[a][b][c]为在第a格， 重力方向为b(1 or -1),terminal的访问情况为c的最小时间
- 当脚下没砖
  - 而是 '.' 时，f[x][b][c] = f[a][b][c] + 1
  - 而是 'F' 时，f[x][b][z] = f[a][b][c] + 1
  - 而是 'G' 时，f[x][-b][c] = f[a][b][c] +1

- 当脚下有砖
  - 可以左右移动，情况和上面类似
- 结果就是 min(f[T][1][0],f[T][-1][0])
- 用BFS搜一遍就行了，总共有200*2*8=3200种状态

The Three Guys

- 给出三个人上半身高度和下半身高度，求这三个人组成的三角形的最大面积。
- Sample 是右边这样的：
  - B – D – E 是第一个人
  - E – F – C 是第二个人
  - C – A – B 是第三个人
- 总面积 30 * 40 / 2 = 600



- 分类讨论很麻烦
- 情况总数远比你想象的多
- 最佳做法：搜索/枚举

- 各位总结一下自己为什么错那么多次吧……

BBQ

- N 份订单，各在 Si 时刻接到订单
- 厨师要花 Ci 的时间完成
- 快递要花 Ri 的时间送出，再花 Ri 的时间返回
- 做法：以时间点为基准进行处理，考察代码的具体实现

- 比如，可以弄三个队列：
- call: 外卖电话什么时候来
- cook: 外卖的制作完成顺序
- wait: 等待送出的外卖顺序

- 每次从前两个队列中得到最早的时间点，然后如果第三个队列存在元素，时间点=min(时间点, 快递员返回的时间)
- 然后将此时间点内发生的事情处理完

Simple Equation

- 解方程AX + BY = XY
- 给出A ， B， M， 求解X ， Y; 其中X>=M,有多解情况时输出X+Y最小的方案，若还有多解情况，输出X最小的方案
- 数据范围：1 <= A, B <= 10 ^ 9 and 1 <= M <= 10 ^ 18

- $AX + BY = XY$ 可转化为 $(x - a)(y - b) = ab$
- 因此只要对 ab 分解质因数即可求出 x 和 y 的所有组合
- 注意枚举的 x 与 y 可以控制在 <= sqrt(ab) 内，这样可以保证时间复杂度

Training

- 三个人组队训练，只要有两个人以上出席训练就能产生收益
- 以线段的形式给出每个人不能出席训练的区间
- 每个人训练的总时间有限制
- 一天的时间可以以**任意方式**切割
- 三个人都能训练不代表三个人都要来训练，没准三人训练还不如某两人训练
- 收益按天算，如果条件允许，他们可以前半天前两个人一起练，后半天三人一起练，那收益会是0.5*a+0.5*d

- 预处理 + 线性规划
- 预处理将所有时间分成五种：有至少两人不能来，某一个人不能来（共三种），所有人都能来
- 可以采用O(MlogT + T)的线段树，也可以借鉴扫描线的思想做到O(T + M)或者O(MlogM + T)
- 也有可能有更快的处理方法

- 所有人都不能来的部分对答案没贡献，可以直接不管，接下来就是怎么构造线性规划来求出答案
- 为了方便说明，下面下标为u的变量均代表Vest和cYz的组合，v的变量代表cYz和Concloudy的组合，w的变量代表Concloudy和Vest的，x代表三个人的
- 通过预处理，我们可以知道只有组合u能训练的时间为tu，v的为tv，w的为tw，以及一起的tx

- 设三人一起训练的时间为sx
- 组合u取自tu的量为pu，取自tx的量为su
- 同理再取pv, sv, pw, sw

- 为什么要设7个变量？4个不就够了么

$$\begin{cases} su + sw + sx + pu + pw \le ka \\ su + sv + sx + pu + pv \le kb \\ sw + sv + sx + pw + pv \le kc \\ pu \le tu \\ pv \le tv \\ pw \le tw \\ su + sw + sv + sx \le kc \end{cases}$$

- 答案求max$[a * (su + pu) + b * (sv + pv) + c * (sw + pw) + d * sx]$

- 使用线性规划单纯形法(simplex algorithm)即可解决
- 可以去看一下集训队论文《浅谈信息学竞赛中的线性规划——简洁高效的单纯形法实现与应用》
- 浙大模板里就有，不会也可以直接贴
- 不知道能不能用费用流去做

完