Contest 1 by ACE

2013.07.12

Overview

- 1001 [几何] Build the Park I
- 1002 [签名] Careerist
- 1003 [树形DP] Code Geass -- Lelouch's Plan
- 1004 [最小割] Escape Game
- 1005 [模拟] Alice and Bob and Cue Sports
- 1006 [枚举] Water Level

1001 Build the Park I

- 对于每个三角形区域，分八种情况讨论：

  （假设3个顶点分别为A,B,C）
- 全挖、全填、A点挖、A点填、B点挖、B点填、C点挖、C点填。
- 对于全挖全填区域：

  直接计算斜截三棱柱体积。
- 对于部分挖填区域：

  计算三棱锥体积；

  计算斜截三棱柱体积+三棱锥体积-三棱柱体积。

1002 Careerist

- 每个国家的人数不能超过总玩家数的一半，即 max_team_num <= n/2;
- 每个野心家（Careerist）独自为一个队伍，而不是所有的野心家为一个队伍，或所有同一国家的野心家一个队伍。

- 【关于 rejudge】由于对n=1的情况没有特判，并且题目的定义也不符合现实。所以把数据范围改为n>=2，并进行了rejudge。Rejudge之后状态有发生改变的人可以找我要bg。

1003 Code Geass -- Lelouch's Plan

有一个森林结构的上下级关系（若干有根树），一个点的上级是它所有的祖先。有3种风险系数：

1、ai：i的初始的风险系数

2、bi：当一个被控制的祖先向i发出命令时i的风险系数

3、ci：控制i的风险系数

可以控制的人数有上限

一个祖先可以控制多个后代

控制可以是远程的（不需要连续传递）

树形DP

- 从下往上处理
- 通过状态表达需求
- f[i][j][k]表示i为根的子树里，有i个人被控制的情况下，子树状态为k时的最小风险系数。
- k=0表示子树中有一些人使用来bi作为自己的风险系数，但是他们还不存在一个被控制的祖先
- k=1表示子树中没有人使用bi作为自己的风险系数，或者所有以bi作为自己风险系数的人，都是合法的，即有一个被控制的祖先。

转移1

- 对于节点i，首先将所有子树的状态汇总：
- i的状态1,只能由子树状态1来构成
- i的状态0,可以由子树任意状态构成

- 要确保所有子树的状态都被加入

转移2

- 然后是i的状态内部转移：
- 状态1,可以选择3种情况：
  - 1、选ai
  - 2、选bi，状态变成0
  - 3、选ci，控制人数加1
- 状态2，可以选择3种情况：
  - 1、选ai
  - 2、选bi
  - 3、选ci，控制人数加1，状态变成1

- 最后把森林的状态汇总即可。
- 复杂度是$O(n*m^2)$

1004 Escape Game

- 一个环上有若干个点，点之间的距离不均匀且给出。另有若干个人要从其中某个点走到另外一个点，起/终点也给出。
- 每个人的 cost 为走过的各条边 cost 之和。
- 没有规定人走的顺序，但是必须一个人走完下一个人才能走。
- 某个人经过一条边 e，如果他前面有 k 个人用反方向走过了 e，那么耗费增加 k * cost(e)
- 求所有人 cost 之和的最小值

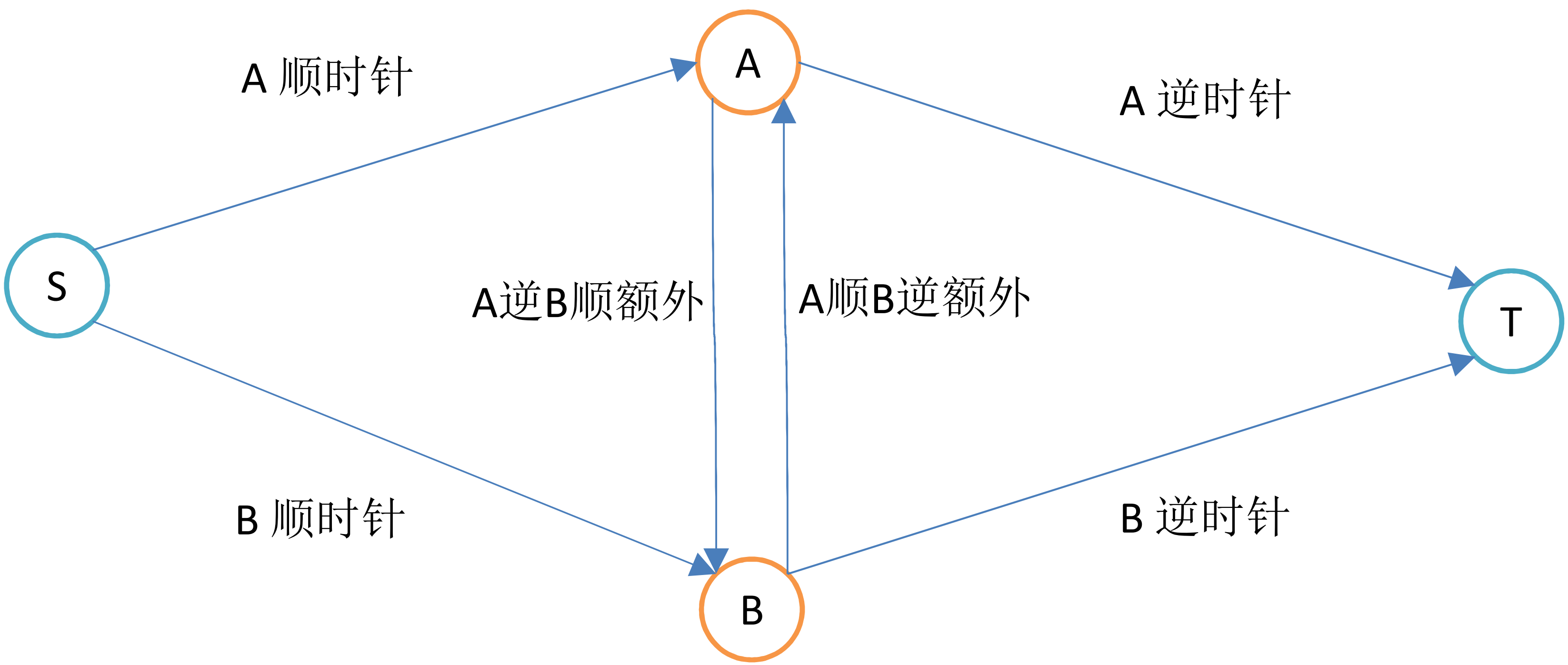
Escape Game 分析

- 每个人只有顺时针/逆时针的走法
- 将增加的 k * cost(e) 分配给他前面以不同方向走过此边的 k 个人，变成 k 个两两之间的关系
- 即：
- 将所有人分成两个集合（顺/逆），任意从两个集合中各挑选出一个元素，他们之间都有 sigma[cost(e)] 为权重的关系

Escape Game 解法

- 最小割，建图时每个人为一个点，属于源/汇侧的点分别对应逆/顺时针的决策
- 从 S 连到 A 为 A 顺时针走的总 cost
- 从 A 连到 T 为 A 逆时针走的总 cost
- 两个人之间相互连两条为（逆/顺）和（顺/逆）重叠的边的额外 cost
- 则每一个割集代表一种决策的总 cost

Escape Game 解法



1005 Alice and Bob and Cue Sports

- 两个人轮流击球，每次的目标球为当前桌上分值最小的球。
- 在不犯规的情况下击入目标球，则可以继续击球，否则交换击球权。

犯规（foul）

- 白球没碰到任何球：目标球分值
- 白球第一次同时碰到两个或两个以上球：碰到球的最高分值
- 白球第一次只碰到一个球，但不是目标球：碰到球的分值。
- 白球（cue ball）落袋
  - 碰到球：碰到球的最高分值
  - 没碰到球：目标球分值

得分

- 击入球的得分与前面犯规的罚分分别计算。
- 在不犯规的情况下，击入的球中包含目标球，则本次击球得分为所有落袋的球的分值总和。
- 否则一律是对手加上所有落袋球分值之和。

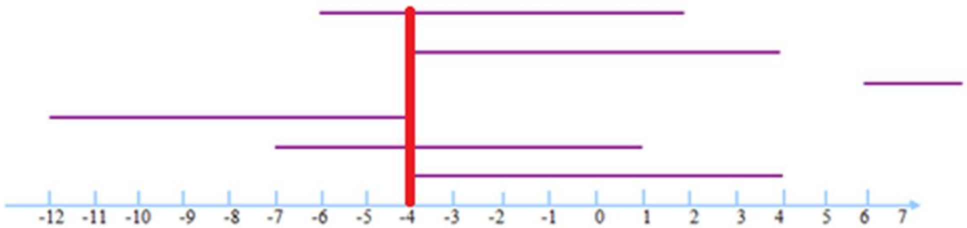
1006 Water Level

- 题目大意：给定n个数的序列，进行至多两次操作$(x_i , c_i)(i =0,1)$。每次操作可以将$A_{x_i} \sim A_n$的值加上c_i。求操作完成后最多能使多少个A_i在$1\sim n$之间

一次操作：

每个数有一个调整区间[1 - Ai , n - Ai]，那么问题就转化成求某一个调整值被区间覆盖最多次。枚举调整值(2*n个)，扫一遍就可以做到。O(n^2)

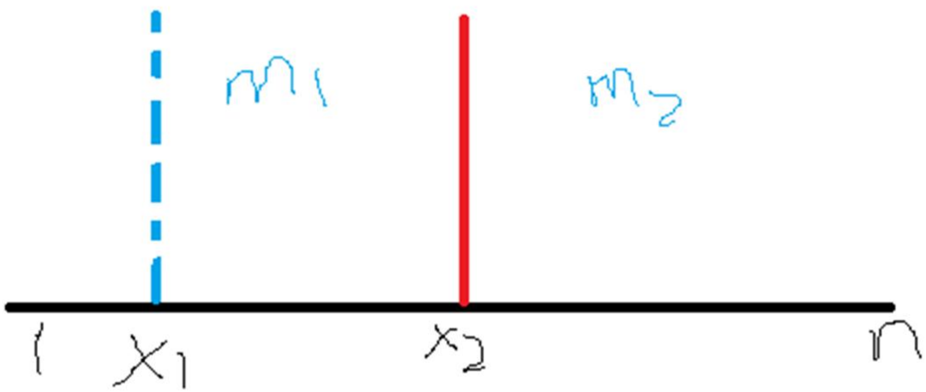
如下图中c = -4为最优解。



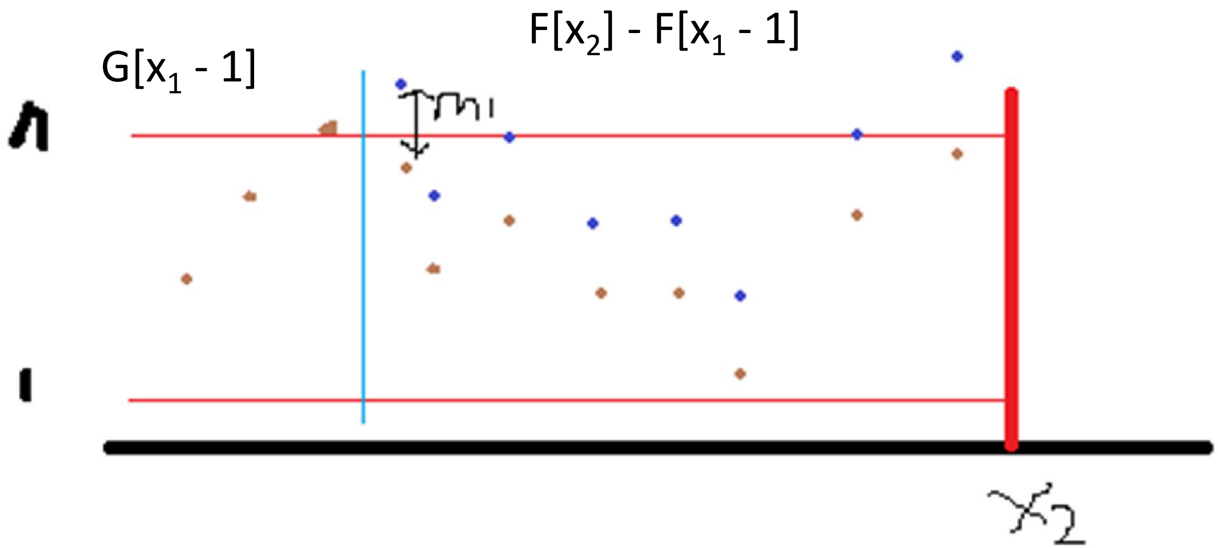
两次次操作：

假设两次操作为(X_0,C_0)，(X_1,C_1)，那么操作等效于将$[A_{x0},A_{x1-1}]$修改加上C_0，$[A_{x1},A_n]$修改加上C_0+C_1。

我们可以枚举X1的位置，然后把问题变成两段求最优解的问题。



X2右段：m2可以与一次操作一样，从后往前扫一遍即可。ANS2[j]记录改变[Aj , An]所能够获得的最大段数。



设G[x]等于在原始状态下，A_1~A_x合法的值的个数。
设F[x]等于在新的状态下（即A_1~A_x都加上m1），A_1~A_x合法的值的个数。
那么左段的最大值：

$$sum = \max\{F[x_2] - F[x_1 - 1] + G[x_1 - 1]\}$$
$$= \max\{F[x_2] - (F[x_1 - 1] - G[x_1 - 1])\}$$

因为$F[x_2]$是常量。所以我们只需要在确定m1的情况下，从前往后扫一遍X_2，不断更新min(F[i] - G[i])即可。

$$ans = \max\{sum + ANS2[x_2 + 1]\}$$

那么最后的复杂度就是O(n^2)