

# NOIP 模拟赛 2

GGN&HJQ

2020 年 8 月 16 日

题目名称	多米诺骨牌	抽卡	归程
英文名称	domino	card	return
源文件名	domino.cpp	card.cpp	return.cpp
输入文件名	domino.in	card.in	return.in
输出文件名	domino.out	card.out	return.out
题目类型	传统	传统	传统
时间限制	1s	1s	1s
空间限制	512MB	512MB	512MB
测试点数目	10	10	10
每个测试点分值	10	10	10

## 注意事项

1. 评测环境:ubuntu18.04 lts 64 位,CPU 为 Intel Core i7-8550U
2. 评测软件:lemon
3. 编译工具:g++10.1.0(如果担心 Compile Error 可以先发给 cppascalinux 进行编译测试)
3. 编译命令:g++ -o %s %s.cpp -lm (%s 为题目英文名)
4. 比较方式: 全文比较, 忽略行末空格和文末回车
5. 请不要直接从题面中复制样例

# 1 多米诺骨牌

## 1.1 题目背景

HJQwQ 在玩多米诺骨牌, 但他已经玩腻了摆放然后推倒的玩法. 于是他找来了国际象棋棋盘, 开始往上面摆骨牌. 这时他突发奇想: 棋盘上最多能摆多少个不相邻的骨牌? 但是他还要去打 MC, 于是把这个问题留给了你

## 1.2 题目描述

有一个  $n$  行  $m$  列 ( $n * m$ ) 的棋盘, 在它的上面放若干张  $1*2$  或  $2*1$  的骨牌, 要求骨牌必须完全位于棋盘内, 每个格子至多只被一张骨牌覆盖, 且这些骨牌两两不相邻 (我们定义两张骨牌是”相邻”的, 当且仅当这两个骨牌至少有一条公共边或一个公共顶点, 如图 1-1 和 1-2 中的两个骨牌是相邻的, 而 1-3 中的两个骨牌不相邻), 求最多可以摆放骨牌的数目

## 1.3 输入格式

第一行一个正整数  $T$ , 表示数据组数接下来  $T$  行, 每行两个正整数  $n, m$

## 1.4 输出格式

共  $T$  行, 每行一个整数, 表示对于每组数据的答案

## 1.5 输入输出样例

domino.in	domino.out
5	1
1 2	2
3 2	5
5 5	5
4 6	9
5 8	

## 1.6 样例解释

对于 5 组数据, 各给出一种可行的方案 (图 2-1→2-5), 不难证明都是最大方案

## 1.7 数据范围

测试点编号	$n$	$m$
1	$= 1$	$\leq 10$
2,3	$\leq 2$	$\leq 10^5$
4,5	$\leq 5$	$\leq 10^5$
6,7	$\leq 10$	$\leq 1000$
8,9,10	$\leq 10^5$	$\leq 10^5$

对于 100% 的数据, 有  $1 \leq T \leq 10, 1 \leq n, m \leq 10^5$

## 2 抽卡

### 2.1 题目背景

HJQwQ 在和 GGN 玩一个抽卡游戏. 他们的面前摆放着  $n$  张背面朝上的卡, 每张卡的正面印着一个价值  $v_i$ . HJQwQ 可以从中抽走  $k$  张, 并获得这些卡上所有的价值. 擅长出老千的 HJQwQ 早就看穿了每张卡的价值, 然而机智的 GGN 更早就料到 HJQwQ 会出老千, 于是他又加了一条规则: 取出的  $k$  张卡的位置必须两两不相邻. 这下 HJQwQ 方了: 他该抽走哪些卡, 才能使自己获得的总价值最大?

### 2.2 题目描述

有一个长为  $n$  的数组  $v_i$ , 请从中选出  $k$  个不相邻的位置, 使得这些位置的和最大, 并输出这个和, 你需要对  $k \in [1, \lfloor \frac{n+1}{2} \rfloor]$  都输出一个答案

### 2.3 输入格式

第一行一个正整数  $n$

第二行  $n$  个用空格分隔的正整数  $v_1 \sim v_n$

### 2.4 输出格式

共  $\lfloor \frac{n+1}{2} \rfloor$  行, 每行一个整数, 第  $i$  行的整数表示  $k = i$  时的答案

## 2.5 输入输出样例

### 2.5.1 样例 1

card.in	card.out
10	9
7 3 2 7 2 7 4 9 9 1	16
	23
	30
	31

见下发文件中的 card1.in/card1.out

### 2.5.2 样例解释

$k = 1$  时, 抽  $v_8 = 9$  最大

$k = 2$  时, 抽  $v_1 + v_9 = 16$  最大

$k = 3$  时, 抽  $v_1 + v_4 + v_9 = 23$  最大

$k = 4$  时, 抽  $v_1 + v_4 + v_6 + v_8 = 30$  最大

$k = 5$  时, 抽  $v_1 + v_4 + v_6 + v_8 + v_{10} = 31$  最大

### 2.5.3 样例 2

见下发文件中的 card2.in/card2.out, 此样例数据范围与测试点 3 相同

### 2.5.4 样例 3

见下发文件中的 card3.in/card3.out, 此样例数据范围与测试点 8 相同

## 2.6 数据范围

测试点编号	$n$	$v_i$
1,2	$\leq 10$	$\leq 10^9$
3,4,5	$\leq 2000$	$\leq 10^9$
6,7	$\leq 2 \times 10^5$	$\leq 2$
8,9,10	$\leq 2 \times 10^5$	$\leq 10^9$

对于 100% 的数据, 有  $1 \leq n \leq 2 \times 10^5, 1 \leq v_i \leq 10^9$

## 3 归程

### 3.1 题目背景

HJQwQ 在地下挖了一背包的铁和钻石, 打算坐矿车回家, 原本的铁路是一条直线. 但命运之神 GGN 和他开了一个小玩笑: 他使用魔法, 把 HJQwQ 的铁路变成了一张有向无环图! HJQwQ 在每一个岔道口处, 会随机地沿着一条出边前进, 只有当一个岔道口没有出边, 他才可以停下来.

HJQwQ 自闭了, 但他也不是无计可施: 他可以使用魔法, 但由于他的法力太弱, 只能从图中删除至多一条边 (也可以不删). 他想知道, 怎样才能使旅程尽快结束 (期望意义下)?

### 3.2 题目描述

形式化的表述如下: 有一张  $n$  个点,  $m$  条边的有向无环图  $G(V, E)$ , 每条边  $(u, v)$  有一个边权  $w_{(u,v)}$  和一个长度  $l_{(u,v)}$  (**可能为负**), 若 HJQwQ 在某个点  $u$ , 则他选择边  $(u, v)$  的概率为  $\frac{w_{(u,v)}}{\sum_{(u,i) \in E} w_{(u,i)}}$ , 即当前出边的权值除以所有出边权值之和.

HJQwQ 从 1 号点出发, 按照这个规则不断前进, 直到一个没有出边的点才停止, 他经过的总路程为他经过的边的长度  $l(u, v)$  之和. 在出发之前, 他可以选择一条边  $(u, v)$ , 并从图中删去这条边 (也可以不选). 请求出他经过的总路程长度的期望值最小是多少.

### 3.3 输入格式

第 1 行两个正整数  $n, m$ , 分别表示图中的点数和边数

第 2  $\sim$   $(m + 1)$  行, 每行 4 个整数  $u, v, w, l$ , 表示有一条从  $u$  到  $v$  的有向边, 权值为  $w$ , 长度为  $l$

3.4 输出格式

一行共一个有理数, 表示总路程的期望值的最小值, 结果四舍五入保留三位小数

3.5 输入输出样例

3.5.1 样例 1

return.in	return.out
5 6 1 2 1 3 1 3 2 4 2 3 3 5 2 4 4 1 3 4 2 3 3 5 1 2	5.600

见下发文件的 return1.in 和 return1.out

3.5.2 样例解释

删除边的编号 (0 表示不删边)	总路程期望 (保留三位小数)
0	6.514
1	6.400
2	6.743
3	5.600
4	7.733
5	6.190
6	7.000



### 3.5.3 样例 2

见下发文件的 return2.in 和 return2.out, 此样例数据范围与测试点 3 相同

### 3.5.4 样例 3

见下发文件的 return3.in 和 return3.out, 此样例数据范围与测试点 8 相同

## 3.6 数据范围

测试点编号	$n$	$m$
1,2	$\leq 10$	$\leq 45$
3,4,5	$\leq 1000$	$\leq 2000$
6,7	$\leq 2 \times 10^5$	$= n - 1$
8,9,10	$\leq 2 \times 10^5$	$\leq 4 \times 10^5$

对于 100% 的数据, 有  $1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 4 \times 10^5, 1 \leq w \leq 10, -10 \leq l \leq 10$ , 保证图为有向无环图, 且图内无重边, 自环