

VI ТУРНИР ЗА КУПАТА НА ДЕКАНА НА ФМИ
02.12.2007 г., Вътрешно състезание

Зад. А. Хакер

Админът espr0 управлява голяма компютърна мрежа от N ($1 \leq N \leq 100\,000$) компютъра. За съжаление, в мрежата му се появил хакер. Админът не знае зад кой компютър се крие злосторникът, но му е известно, че той ползва HackOS ver. 2.0. За късмет, espr0 е открил загадъчен бгг (или anti-anti-hack feature) в HackOS. Бггът се състои в това, че при изпращане на няколко PING ECHO req пакети до компютър с инсталирана HackOS 2.0, запитаният компютър връща само един отговор и пропуска останалите пакети (нормалните компютри връщат по един отговор за всеки PING ECHO req пакет). espr0 е изготвил следния план за залавяне на виновника – изпраща по два PING-a до всеки компютър от мрежата и следи кой от всичките е отговорил само веднъж. Речено – сторено, обаче, поради различните закъснения в мрежите и проблеми с маршрутизацията, отговорите дошли в много разбъркан ред. Помогнете на админа да разбере кой все пак е злосторника, като откриете номера на компютъра, който е отговорил само веднъж.

Вход. Данните ще бъдат зададени на стандартния вход и съдържат описания на няколко мрежи. На първият ред ще е число T – броят на мрежите. Описанието на всяка мрежа започва с броя N на компютрите в мрежата. Следват $2N-1$ числа – номерата на отговорилите компютри, в реда на получаване на пакетите. Номерата на компютрите са цели числа в интервала $[0..2\,000\,000\,000]$. Гарантирано е, че всеки номер на компютър се среща точно два пъти, с изключение на номера на компютъра, на който работи хакерът.

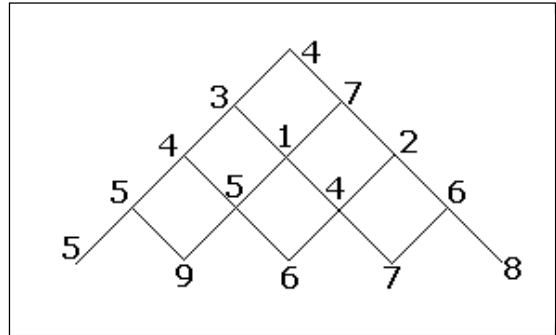
Изход. За всеки тест програмата трябва да изведе на един ред на стандартния изход едно число – номера на компютъра, зад който се крие хакерът.

Примерен вход:	Съответен изход:
2	3
3	15
2 1 3 1 2	
7	
1 10 15 4 10 19 1 9 8 4 9 19	
8	

VI ТУРНИР ЗА КУПАТА НА ДЕКАНА НА ФМИ
02.12.2007 г., Вътрешно състезание

В. Квазипирамида

Синът на фермерът espr1t Станчо се опитва да мине за програмист, но мрази пирамидите! Не, не тези, които обещават да раздават по 100% лихва за всеки вложен в тях лев, а тези за които непрекъснато му говори преподавателят му по Алгоритми. Както навярно добре знаете, едно пълно двоично дърво, по върховете на което са поставени числа наричаме пирамида, ако числото във всеки връх, който не е лист, е по-малко от или е равно на числата, поставени в двата му сина. Очевидно спекулирайки с



понятието, преподавателят поставил следната задача. Даден е ориентиран ацикличен граф, върховете на който могат да се наредят в N нива така, че в I -тото ниво има I върха, за $I = 1, 2, \dots, N$. Освен това, за $I = 1, 2, \dots, N-1$, J -тият връх в I -тото ниво е баща точно на J -тия и $(J+1)$ -вия връх в $(I+1)$ -вото ниво, а върховете в N -тото ниво нямат синове. Такъв граф преподавателят нарекол **квазимрежа от ред N** (на фигурата е показана квазимрежа от ред 5). Всеки връх в квазимрежата, **заедно с всичките си наследници** поражда квазимрежа от по-малък ред. По върховете на квазимрежата са поставени цели положителни числа. Преподавателят нарекол **квазипирамида** такава квазимрежа, в която за всеки връх, който има синове, числото във върха е по-малко от или равно на числата, поставени в двата му сина. Например, всички върхове от мрежата на фигурата, които са в трето ниво пораждат квазипирамиди от ред 3. Задачата е, по зададена квазимрежа да се разместят числа така, че да се получи квазипирамида от колкото може по-висок ред, като две числа могат да си сменят местата, **само ако са в едно и също ниво**.

Вход. На първия ред на стандартния вход е зададен броят T на тестовите примери, които програмата трябва да реши. Всеки тест започва с ред, на който е зададено числото N ($3 \leq N \leq 1000$). I -тия от следващите N реда съдържа числата от I -тото ниво на квазирешетката, никое от които не е по-голямо от 1 000 000.

Изход. За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход едно число – максимален ред на квазипирамида, получена при разместване на числата .

Примерен вход:	Съответен изход:
2	1
2	3
3	
1 2	
5	
4	
3 7	
4 1 2	
5 5 4 6	
5 9 6 7 8	

VI ТУРНИР ЗА КУПАТА НА ДЕКАНА НА ФМИ
02.12.2007 г., Вътрешно състезание

Зад. С. Проекти

Зимната сесия във ФМИ приближава и лекторът `espr1t`, водещ традиционния избран курс “Вертексни Алгебри и Глобално Конформно-Инвариантна Квантова Теория на Полето” е задал K ($0 \leq K \leq 22$) курсови проекта на своите N ($2 \leq N \leq 9$) студента. `espr1t` е разпределил проектите по случаен начин и изисква само две неща: всеки проект трябва да се разработва от точно двама души; освен това всеки студент не трябва да бъде натоварен с повече от M ($1 \leq M \leq 5$) проекта (имайте предвид, че числото K няма да надхвърля максималният възможен брой проекти, който позволява да се спазят горните две правила). За радост на студентите, сървърът за `upload` на проектите не поддържа сигурна идентификация, което значи, че лекторът няма как да разбере дали даден проект наистина е направен от хората, на които е бил първоначално възложен. Студентите решили да се възползват от това и решили да разпределят проектите помежду си както им е най-удобно. Било установено, че проектът с номер k би бил оценен с $S_{k,i,j}$ точки, ако бъде изработен от студентите с номера i и j (точките са цели числа от 0 до 100 включително). Търсеното преразпределение не може да нарушава правилата, формулирани от `espr1t`. Освен това се иска сумарният брой точки от всички проекти да е максимален. Намерете този максимум!

Вход. На първия ред на стандартния вход ще бъде зададен броят на тестовите примери T . За всеки тест на първия ред ще бъдат зададени числата N , K и M . Следват K таблици от $N \times N$ числа, като числото $S_{k,i,j}$ е j -тото число на i -тия ред от k -тата таблица. Таблиците са коректни, т.е. симетрични са спрямо главния диагонал, а самият диагонал съдържа нули (което означава, че ако се опитате да изпратите един проект, подписвайки го два пъти с името на студента i , ще получите в резултат 0 точки, както ви се полага).

Изход. За всеки тестов пример програмата трябва да изведе на стандартния изход едно число – максималния общ брой точки, който може да се постигне при оптимално разпределение на всички проекти.

Примерен вход:	Съответен изход:
1 3 3 2 0 9 1 9 0 1 1 1 0 0 8 1 8 0 1 1 1 0 0 9 1 9 0 1 1 1 0	11

VI ТУРНИР ЗА КУПАТА НА ДЕКАНА НА ФМИ
02.12.2007 г., Вътрешно състезание

D. Маршрутки

В град ГНИ има N спирки на маршрутни таксите ($2 \leq N \leq 1000$), номерирани с числата от 1 до N . Купонджията `espr1t` се намира на спирка 1 и иска да отиде до спирка N , като плати минимална цена. За целта може да използва една или повече маршрутки. В града има M маршрутни линии ($1 \leq M \leq 20\,000$). Всяка от тях предлага **еднопосочен превоз между две спирки**. Цената се определя от изминатото разстояние – C стотинки на километър за всички маршрутки в града ($1 \leq C \leq 100$). За i -тата маршрутка са известни числата S_i , E_i и D_i : начална спирка, крайна спирка и разстоянието в километри, съответно ($1 \leq S_i \neq E_i \leq N$, $1 \leq D_i \leq 1000$, D_i е цяло).

Тъй като задачата изглежда прекалено лесна, а и от Маршрутния Транспорт на ГНИ решили, че така прекалено много хора щели да научат да смятат алгоритъма на Дейкстра наум, на пазара били пуснати Далаверски Карти (ДК). Всяка ДК обезпечава определен брой безплатни километри при всяко ползване на маршрутка (да означим този брой с k). Цялото k е отпечатано на ДК и се избира от клиента при закупуването на ДК. ДК също струват пари: цената на ДК с k безплатни километра е $P * k$ стотинки ($C \leq P \leq 10\,000$, P е цяло). Схемата е следната: след като се качи в някоя маршрутка, пътникът не плаща първите k километра от пътя (ако целия път не надвишава k километра, пътникът не плаща нищо). ДК важат за всички маршрутки и могат да се ползват произволен брой пъти (закупуването е само веднъж). `espr1t` се чуди как точно да пътува и дали да си купи ДК (и ако да, каква). Помогнете му: сметнете колко стотинки най-малко ще трябва да плати. Избраният път не е от значение – `espr1t` не бърза.

Вход. Програмата трябва да обработи няколко тестови примера. Първият ред на стандартния вход съдържа броя T на тестовете. Първият ред на всеки тест съдържа числата N , M , C и P , разделени с интервали. Всеки от следващите M реда описва един маршрут и съдържа числата S_i , E_i и D_i , разделени с интервали. Всички маршрути са еднопосочни и може да има повече от един маршрут между две спирки. Гарантирано е, че има път от спирка 1 до спирка N .

Изход. За всеки тест програмата трябва да изведе на стандартния изход минималната цена (в стотинки), която `espr1t` трябва да плати, за да стигне от спирка 1 до спирка N . Цената включва разходите за ДК (ако има такава) и парите, платени в маршрутките.

Примерен вход:	Съответен изход:
2	6
5 5 1 3	8
1 2 5	
1 3 2	
2 3 1	
3 4 3	
4 5 1	
7 8 1 4	
1 2 4	
1 4 2	
2 3 4	
2 5 5	
3 4 1	
4 5 2	
5 6 2	
6 7 2	

VI ТУРНИР ЗА КУПАТА НА ДЕКАНА НА ФМИ
02.12.2007 г., Вътрешно състезание

Зад. Е. Цикъл

Учените от Националния институт за изследване на паралелни процеси с удоволствие разбраха за предстоящия Турнир за Купата на Декана. Още повече ги зарадва фактът, че в това състезание участват хора, които изпитват удоволствие от решаването на алгоритмични проблеми. Затова решиха, вместо да натоварват себе си, да поизпотят малко вас. Оказва се, че от известно време ги измъчва проблем, който играе ключова роля в няколко от основните им проекти. Затова неговото решаване е от особено важно значение за тях. Проблемът се състои в следното. Учените разглеждат ориентиран граф, с не повече от 1000 върха и не повече от 2000 ребра. На всяко ребро e на графа е съпоставена дължина – цяло число $L(e)$, не по-голямо от 1000. Учените искат да намерят цикъл, който минава по някои от дадените ребра (но не повече от един път по едно и също ребро) и такъв, че средната му дължина (т.е. сумата от дължините на ребрата, които участват в цикъла, разделена на броя им) да е максимална. Например, ако в цикъла участват ребрата e_1, e_2, \dots, e_k , то искаме стойността $(L(e_1) + L(e_2) + \dots + L(e_k)) / k$ да бъде възможно най-голяма. Помогнете на учените да решат този проблем. Вашата задача е, да напишете програма, която да намери и изведе максималната средна дължина, която се достига върху някой от циклите в зададен граф, не повтарящ ребра.

Вход. На първия ред на стандартния вход ще бъде зададен броят T на тестовете. Всеки тест започва с ред, на който са зададени две цели числа N и M – броят на върховете и броят на ребрата в графа, съответно. Всеки от следващите M реда съдържа по три цели числа a, b и c , което означава, че между върховете a и b има ребро с дължина c , $1 \leq a \leq N$, $1 \leq b \leq N$, $1 \leq c \leq 1000$. В графа може да има примки, както и повече от едно ребро между два върха, като всяко от тях може да използвате най-много по един път.

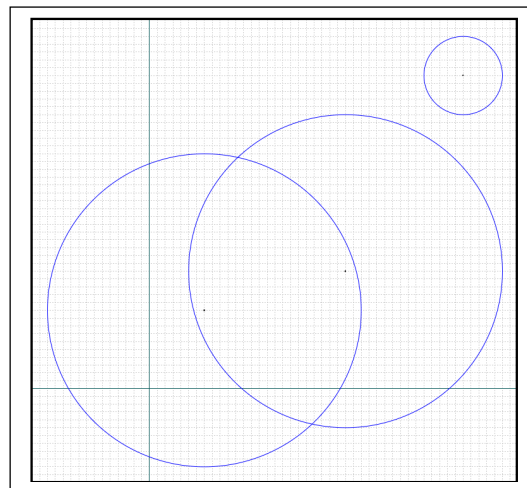
Изход. За всеки тест, на отделен ред на стандартния изход, програмата трябва да изведе намерената максимална средна дължина на цикъл от дадения граф, закръглена до 2 знака след десетичната точка. (Когато извеждате резултата, закръглявайте по общоприетия начин, т.е. 21.665 да бъде изписано като 21.67, а 21.664 като 21.66.)

Примерен вход:	Съответен изход:
2	21.67
3 3	7.67
1 2 20	
2 3 21	
3 1 24	
6 8	
1 2 10	
2 3 10	
3 4 9	
4 2 2	
4 1 1	
3 5 9	
5 6 7	
6 3 7	

VI ТУРНИР ЗА КУПАТА НА ДЕКАНА НА ФМИ
02.12.2007 г., Вътрешно състезание

Е. Наследство

Фермерът esprlt оставил голямо наследство на синовете си Генчо, Пенчо и Станчо – няколкостотин декара нива! Само че наследниците му не искали да се занимават с цялата нива и всеки си избрал по една част от бащината нива, с формата на кръг. Оказало се, обаче, че при подялбата на наследството Генчо, Пенчо и Станчо не проверили, да не би избраните от тях парчета да имат общи части и тъкмо това бил проблемът – данъчните служби отказали да правят сложни сметки с лица на кръгове и части от тях и поискали данък за всеки от собствениците поотделно, пропорционален на площта на съответния кръг. Това пък означава, че след като нивите на Генчо и Пенчо имат обща част, то те плащат два пъти за общата част.



След дълги преговори, данъчните се съгласили да се избегне “двойното данъчно облагане”, но поискали да се впише в данъчния формуляр общата площ на трите кръга. За да могат да контролират процеса на плащането, не само за себе си, но и в помощ на своите съселяни, които също имат подобни “кръгли” проблеми, тримата братя решили да създадат оф-шорна фирма GPS Solutions, Ltd. за извършване на “кръгови земемерски пресмятания” и ви предлагат да напишете програма за намиране на общото лице на три кръгообразни нивички. И тежко ви и горко ако сгрешите в сметката – ревизорите ще глобят собствениците солено, а те ще си го изкарат на вас!

Вход. На стандартния вход има няколко описания на разположенията на нивите. Всяко описание е разположено на отделен ред и се състои от 9 цели числа $X_1, Y_1, R_1, X_2, Y_2, R_2, X_3, Y_3$ и R_3 , координатите на центровете и радиусите на нивите на Генчо, Пенчо и Станчо, съответно ($-100 \leq X_i, Y_i \leq 100; 1 \leq R_i \leq 100$), разделени с интервали. Координатите на центровете и радиусите са зададени в метри.

Изход. За всяко описание на ниви програмата трябва да изведе на отделен ред на стандартния изход по едно цяло число – общата площ на трите ниви, в квадратни метри, закръглена надолу до цяло число (неплащането на данък за някакъв си половин квадратен метър не е проблем за стабилна данъчна система, като нашата).

Примерен вход:	Съответен изход:
7 10 20 25 15 20 40 40 5	2054
100 100 100 100 100 50 -100 -100 100	62831