



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача А. DAG

Задачата за намиране на най-дълъг път в произволен граф е NP-пълна и се решава с пълно изчерпване. Когато в графа **няма цикли** задачата има ефективно и сравнително лесно за реализация решение, илюстрирано със следния псевдо код:

<p>Algorithm DAG-SHORTEST-PATHS(G, s)</p> <pre> 1 compute a topological sorting of G 2 for each vertex $v \in V[G]$ 3 $d[v] \leftarrow \infty$ 4 $\pi[v] \leftarrow nil$ 5 $d[s] \leftarrow 0$ 6 for each u taken in topologically sorted order 7 for each $v \in adj(u)$ 8 if $d[v] > d[u] + w(u, v)$ 9 $d[v] \leftarrow d[u] + w(u, v)$ 10 $\pi[v] \leftarrow u$ </pre>	<p>Algorithm DAG-LONGEST-PATHS(G, s)</p> <pre> 1 compute a topological sorting of G 2 for each vertex $v \in V[G]$ 3 $d[v] \leftarrow 0$ 4 $\pi[v] \leftarrow nil$ 5 $d[s] \leftarrow 0$ 6 for each u taken in topologically sorted order 7 for each $v \in adj(u)$ 8 if $d[v] < d[u] + w(u, v)$ 9 $d[v] \leftarrow d[u] + w(u, v)$ 10 $\pi[v] \leftarrow u$ </pre>
---	---

Даден ви е **насочен ацикличесен граф** с $5 \leq N \leq 10000$ върха и M ($N \leq M \leq 100000$) ребра, с положителни тегла, не по-големи от 10000. Искане се да намерите дължините на най-късия и най-дългия път, започващи от зададен стартов връх S и завършващи в кои да е други върха на графа, както и да изведете самите пътища.

За всеки тест на първия ред на стандартния вход са зададени N и M . Следват M реда с по три числа - u , v и c , описващи насочено ребро от u до v с цена c . Най-накрая е зададен стартовия връх S . За край на входа служи една нула.

За всеки тест извеждайте на отделни редове дължините на най-късия и най-дългия път, започващи от S , както и самите пътища. Ако съществува повече от едно най-добро решение, извеждайте кое да е от тях. Следвайте изходния формат, показан в примера:

Вход	Изход
6 7	the shortest path from 6 ends in 3, cost: 1
6 5 10	6 3
1 4 11	the longest path from 6 ends in 4, cost: 17
1 2 4	6 3 1 4
3 1 5	
2 4 5	
6 3 1	
6 1 3	
6	
0	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача В. Полети

Авиокомпания предлага M ($1 \leq M \leq 20000$) еднопосочни полета до N ($1 \leq N \leq 20000$) града, като K ($1 \leq K \leq 200$, $K \leq N$) от тези градове са т.н. разпределителни, т.е. там става прекачването от един самолет на друг. Всеки полет от u_i до v_i задължително трябва да минава през поне един от тези разпределители. Предлага се най-много един директен полет между два града в дадена посока и няма полети започващи и завършващи в един и същи град.

Авиокомпания трябва да обработи Q заявки ($1 \leq Q \leq 50000$), на желаещи да летят от град a_i до град b_i . Ако това е възможно, то трябва да се изчисли минималната цена, на която това може да стане.

Вашата задача е да напишете програма, която изчислява броя на полетите, които могат да бъдат реализирани, както и сумарната им цена (тя може да не се побира в 32 битов целочислен тип).

На първия ред на стандартния вход е зададен броя на тестовете. На първия ред на всеки от тях са зададени N , M , K и Q . Следват M реда с по три числа - u_i v_i d_i , описващи директен полет от u_i до v_i , на цена d_i ($1 \leq d_i \leq 10000$), $1 \leq u_i, v_i \leq N$, $u_i \neq v_i$. Следват K реда, на които са зададени номерата на разпределителните. Следват Q реда със заявки - a_i b_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$).

За всеки тест извеждайте по два реда. Първия трябва да съдържа броя на полетите, които могат да бъдат осъществени, а следващия ред съдържа минималната обща цена, на която това може да стане.

Вход	Изход
1	1
3 3 1 2	20
1 2 10	
2 3 10	
2 1 5	
2	
1 3	
3 1	

Пояснение към примера: За първата заявка единственият възможен маршрут е $1 \rightarrow 2 \rightarrow 3$, на цена 20. Втората заявка не може да бъде изпълнена, тъй като няма полети тръгващи от град 3.



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача С. Думи

Езикът на племето „Мулунгу” е съставен само от малките латински букви: ‘a’, ‘b’ и ‘c’. Под дума в този език се разбира всяка последователност от не повече от m букви, в която всяка буква от думата е равна на предходната или е по-назад в лексикографската наредба. Буквата ‘a’ е преди буквата ‘b’ и буквата ‘b’ е преди буквата ‘c’ в лексикографската наредба. Например в изречението “aaabccbabbbcc” има точно три думи: “aaabcc”, “b”, “abbbcc”

Дадено ви е изречение на езика на племето „Мулунгу”, което е не по-дълго от n знака. Вие сте много любопитни и постоянно задавате въпроси от вида: „Колко думи има в интервала от i -тия знак на изречението до j -тия знак на изречението”.

Напишете програма, която по дадено изречение и въпроси отговаря на всеки един от тях.

Вход

За всеки тест на първия ред е зададен текст, съдържащ само малките латински букви: ‘a’, ‘b’ и ‘c’. На следващия ред е зададено едно число k – броят на въпросите. Следват k реда с по две числа i и j – номера на началния и номера на крайния знак от изречението, за който е поставен въпроса. Номерирането на знаците в изречението започва от 1.

Изход

За всеки зададен въпрос на отделен ред изведете по едно число – броя на думите в интервала от i -тия до j -тия знак включително.

Ограничения:

$$1 \leq n, m \leq 20000$$

$$1 \leq k \leq 100000$$

$$1 \leq i \leq j \leq n$$

Вход	Изход
aaabccbabbbcc	3
5	1
1 13	2
3 6	1
3 7	3
8 9	
5 10	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача D. Трибоначи

Редицата на Трибоначи се дефинира по следния начин:

$$\begin{aligned}
 trib_0 &= 0, \\
 trib_1 &= 0, \\
 trib_2 &= 1, \\
 trib_n &= trib_{n-1} + trib_{n-2} + trib_{n-3} \text{ за } n \geq 3.
 \end{aligned}$$

Нека P е естествено число и нека a_n е остатък при деление на $trib_n$ с P (т.е. $a_n = trib_n \% P$).

Напишете програма, която намира номера k на първия елемент на редицата a_n , от който тя започва да се повтаря, т.е. най-малкото естествено число k , за което $a_k = a_0, a_{k+1} = a_1, a_{k+2} = a_2, a_{k+3} = a_3, \dots$.

Вход

За всеки тест на отделен ред на стандартния вход е зададено числото P ($2 \leq P \leq 1000$). Края на входа е маркиран с две нули.

Изход

За всеки тест програмата трябва да извежда числото k , на отделен ред. Ако естествено число k с описаното свойство не съществува, то програмата трябва да извежда -1.

Вход	Изход
2	4
10	124
31	331
0 0	

Обяснение на първия пример:

Редицата на Трибоначи е 0, 0, 1, 1, 2, 4, 7, 13, 24, ...

Редицата a_n е 0, 0, 1, 1, 0, 0, 1, 1, 0, Следователно $k = 4$.



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача Е. Сортиране

Имплементирате сортиращ алгоритъм, който изисква $c \cdot n \cdot \lg(n)$ наносекунди, за да сортира n цели положителни числа. По зададени c ($1 \leq c \leq 100$) и време t ($1 \leq t \leq 2000000000$) се иска да изведете максималния брой числа, които могат да бъдат сортирани за това време, т.е. $c \cdot n \cdot \lg(n) \leq t$. Търсеното число ще е с плаваща запетая и с допустима относителна или абсолютна грешка не по-голяма от $1e-9$.

Имайте в предвид, че \lg е двоичен логаритъм и $\lg(n) = \ln(n) / \ln(2)$, където \ln е натурален логаритъм.

Всеки тестов пример ще бъде зададен на отделен ред на стандартния вход и ще съдържа числата c и t , разделени с интервал. Края на входа е маркиран с "end".

За всеки тестов пример извеждайте търсеното число, на отделен ред на стандартния изход.

Вход	Изход
1 8	4.0
2 16	4.0
37 12392342	23104.999312341137
1 2000000000	7.637495090348122E7
end	

Пояснение към примерите:

В първия пример за 8 наносекунди можем да сортираме 4 числа, тъй като $1 \cdot 4 \cdot \lg(4) = 4 \cdot 2 = 8$.

Във втория пример $c = 2$ и ще ни трябва двойно повече време, за да сортираме 4 числа. В третия пример можем да сортираме почти 23105 числа, но не съвсем.



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача F. Магически квадрати

Казваме, че квадратът на фигурата е *магически*, ако са равни сборовете на числата във всеки ред, всеки стълб и всеки от двата диагонала.

a_1	a_2	a_3
a_4	a_5	a_6
a_7	a_8	a_9

Напишете програма, която по зададени девет цели положителни числа определя колко различни *магически* квадрата могат да се образуват от тях като се използват всички числа.

Вход

За всеки тест на отделен ред на стандартния вход се въвеждат девет цели положителни числа, по-малки от 100.

Изход

За всеки тест на отделен ред на стандартния изход да се изведе отговорът на задачата, както е показано в примера. Ако от зададените числа не може да се образува магически квадрат, програмата да извежда числото 0.

Вход	Изход
1 3 2 4 5 8 7 6 9	Case #1: 8
1 1 1 1 1 1 1 1 1	Case #2: 1

НБУ



НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

София 1635, ул. Монтевидео 21

тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 февруари 2014 г.

Задача G.

Вход

Изход

Вход	Изход