

Проблем . Меде

Као и све мале девојчице, Катарина обожава плишане меде. Зато је на време почела да се припрема за Дан плишаних меда који се одржава у њеном градићу. На тај дан у свакој продавници играчака сваки посетилац добија на поклон по једног плишаног меду.

Постоји n врста плишаних меда. Катарина је обилазила продавнице и закључила следеће: Нека су продавнице нумерисане бројевима од 1 до m . За i -ту врсту меда постоје бројеви a_i и b_i који говоре да се тај меда продаје у продавницама нумерисаним бројевима из интервала $[a_i, b_i]$.

Пре или касније, Катарина ће желети да има по једног меду од сваке врсте. Зато она жели да сума цена меда које не добије током Дана плишаних меда буде што мања. Ваш задатак је да одредите минималну вредност те суме.

Претпоставља се да Катарина сваку продавницу може посетити највише једанпут и да у продавници може изабрати било ког од меда који се у њој продају.

Улаз. (Улазни подаци се читавају из датотеке `mede.in.`) У првом реду улазне датотеке налази се број n ($n \leq 20.000$), укупан број врста плишаних меда. Затим се у наредних n редова налазе по три броја - у i -том реду бројеви a_i , b_i и c_i ($1 \leq a_i \leq b_i \leq 2.000$, $c_i \leq 10.000.000$) - a_i и b_i су границе интервала бројева продавница у којима се i -та врста меда продаје, а c_i је цена те врсте. Све цене меда су различите.

Излаз. (Излазни подаци се исписују у датотеку `mede.out.`) У излазној датотеци треба да се налази само један број - минимална могућа сума цена меда које Катарина не добије током Дана плишаних меда.

Пример 1.

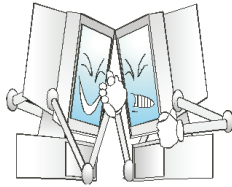
<code>mede.in</code>	<code>mede.out</code>
6	8
1 4 1	
3 4 2	
1 2 3	
2 2 4	
1 1 5	
1 1 6	

Објашњење. Први меда се може пронаћи у продавницама 1, 2, 3 и 4, и његова цена је 1. Други меда се може пронаћи у продавницама 3 и 4, и његова цена је 2, итд. Једно од решења је да у продавници 1 добије меду 6, у продавници 2 меду 4, у продавници 3 меду 1 и у продавници 4 меду 2. Тада јој преостају меде 3 и 5, њихов збир цена је 8.

Пример 2.

<code>mede.in</code>	<code>mede.out</code>
2	0
3 10 15	
15 20 25	

Објашњење. Катарина може добити обе врсте меда, па касније неће морати да купи ниједног меду.



Проблем . Грамзиви Фармери

Последњих неколико месеци, наш стари фармер Бранко, видевши да се доста фармера преселило у град, одлучио је да подели своју земљу осталим фармерима који ће вредно да раде на њој, и тиме спречи да његово месташце нестане.

Фармер Гојко, који је живео недалеко од Бранковог поседа, је управо сазнао за ову акцију. Пожурио је и он да добије неко парче земље, али већ је било касно. Бранко је већ поделио скоро целу земљу. Остало је само једно мало парче земље.

Изненађен оваквим развојем ситуације, фармер Бранко је измислио игру, којом ће поделити остатак своје земље, тако да што више људи добије своје парче. Бранко је поставио n ($n \leq 2000$) стубића у земљу. Онај ко први дође има право да узме било које четвороугаоно парче земље, чија су темена тамо где се налазе стубићи. Правила игре за оне који стигну касније су веома компликована. На вашу срећу, фармер Гојко је стигао први, и сада треба да изабере четири стубића. Помозите му да изабере најбоља 4 стубића, тако да добије земљу највеће површине.

Улаз. (Улазни подаци се читавају из датотеке `farmeri.in`.) У првој реду улаза налази се број n ($4 \leq n \leq 2.000$). У следећих n редова налазе се по два мешовита (реална) броја - x и y координата једног стубића ($-200.000, 0 \leq x, y \leq 200.000, 0$).

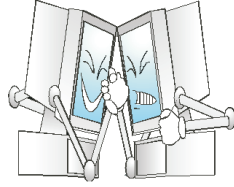
У 50% тест примера, $n \leq 200$.

Излаз. (Излазни подаци се исписују у датотеку `farmeri.out`.) У излазном фајлу требате исписати највећу површину коју фармер Гојко може добити, заокружену на три децимале.

Напомена. Нису сви стубићи колинеарни, тј решење ће увек бити веће од 0.

Пример 1.

<code>farmeri.in</code>	<code>farmeri.out</code>
6	1.500
0 0	
0 1	
1 0	
1 1	
0.5 0.5	
1.5 1.5	



Проблем ш. Књиге

Мали Ђурица је велики обожаваатељ књига и чврсто је решио да прочита што више књига у наредних годину дана. Први, јелте, логичан корак је да оде у библиотеку и тамо потражи штиво за читање, што је Ђурица и урадио. Ради поједностављења, библиотеку ћемо замислити као низ књига. Ђурица сматра да ће читање бити занимљивије ако чита књиге различитих аутора. Сходно томе, прва ствар коју је Ђурица закључио да му је неопходна јесте да преброји колико различитих аутора је учествовало у писању датог подниза узастопних књига.

Ту ступате ви. Баш вас је замолио Ђурица да му помогнете и дате одговор на дато питање. Једна олакшавајућа околност је то што за сваку књигу важи да је писао искључиво један аутор. Уместо имена аутора књиге, Ђурица ће вам дати јединствени број аутора, који је у ствари број из интервала $[1, 100.000]$. Сваки аутор има тачно један јединствени број, и сваки јединствен број одговара тачно једном аутору.

Улаз. (Улазни подаци се читавају из датотеке `knjige.in`.) У првом реду налазе се цели бројеви n ($1 \leq n \leq 100.000$) и K ($1 \leq K \leq 100.000$). У наредном реду се читава n целих бројева из интервала $[1, 100.000]$ који редом представљају јединствене бројеве аутора књига датог низа. Потом се у K редова читавају редом индекси idx_{s_i} и idx_{e_i} ($idx_{s_i} \leq idx_{e_i}$, $1 \leq i \leq K$) који представљају индекс почетка и крај подниза књига, редом, за који треба одговорити колико различитих аутора је учествовало у писању књига тог подниза.

Израз. (Изразни подаци се исписују у датотеку `knjige.out`.) У K редова исписати одговоре на постављена питања, у i -том реду одговор за подниз чији су индекси idx_{s_i} и idx_{e_i} из улаза.

Пример 1.

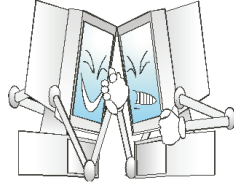
<code>knjige.in</code>	<code>knjige.out</code>
5 6	4
1 2 3 4 3	1
1 5	2
2 2	4
3 5	3
1 4	3
2 5	
2 4	

Пример 2.

<code>knjige.in</code>	<code>knjige.out</code>
4 3	2
1 1 12 12	1
1 4	1
1 2	
3 4	

Напомена.

- У 40% тест примера јединствени бројеви аутора ће се кретати у интервалу $[1, 60]$.
- У 30% тест примера ће и број упита и број различитих аутора бити највише 1000.
- У 60% тест примера ће бити испуњен бар један од претходна два услова.



Проблем . Подела

Дата је повезана мапа n градова са m двосмерних путева. Само два од n градова A и B имају фабрику чоколаде. После краћих договора, људи су одлучили да направе две дисјунктне повезане мреже користећи постојећих m путева, тако да град A дистрибуира чоколаду у свом делу, а град B у свом делу. Сваки град мора да припада тачно једној мрежи. Ако са W_A и W_B означимо укупну дужину путева у мрежама која садрже A и B , потребно је минимизирати већу од укупних дужина мрежа, односно минимизирати израз $\max(W_A, W_B)$. За овај задатак потребно је да предате излазне датотеке за 10 улазних датотока `podela.01.in`, `podela.02.in`, ..., `podela.10.in`, који се налазе у архиви на рачунару. Излази се памте у датотеке `podela.01.out`, `podela.02.out`, ..., `podela.10.out`, при чему број у називу излазне датотеке одговара броју у називу улазне датотеке.

Улаз. У првом реду улаза се налазе два природна броја n и m . У другом реду се налазе индекси градова A и B , док се у следећих m редова налазе описи путева: у сваком реду се налазе три броја x , y и z , који означавају да постоји пут између градова x и z дужине z .

Излаз. У првом реду излаза се налазе два реална броја, који представљају тежине мрежа које садрже град A и град B , редом. Затим следе комплетни описи мрежа: у следећем реду штампати број градова и број путева који припадају мрежи која садржи град A , а затим и све путеве из мреже; у следећем реду штампати број градова и број путева који припадају мрежи која садржи град B , а затим и све путеве из мреже.

Ограничења.

- $1 < n \leq 500$
- мапа градова је увек повезана
- дужине путева су реални бројеви строго већи од нуле, а мањи од 10.000

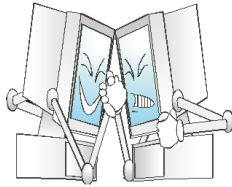
Пример 1.

<code>podela.in</code>	<code>podela.out</code>
6 8	5.0 2.0
1 6	3 2
1 2 3.0	1 2
1 3 3.0	2 5
2 4 2.0	3 2
2 5 2.0	3 4
3 4 1.0	4 6
4 5 3.0	
4 6 1.0	
5 6 4.0	

Оцењивање. Број бодова за сваки тест пример се одређује на следећи начин. Нека је максимална дужина путева у мрежама у вашем решењу $VAL = \max(W_A, W_B)$, а OPT је минимална вредност међу свим такмичарским решењима, укључујући и комисијско решење. Број бодова које добијате се рачуна по формули (заокругљивање се врши на најмањи цео број који није мањи):

$$10^{11-10*VAL/OPT}.$$

Другим речима, ако је ваше решење идентично са најбољим добијате 10 поена. У случају да је решење 10 или више пута веће (лошије) од најбољег решења - добићете 1 поен. Ако је дужина путева коју наведете у излазном фајлу различита од стварне дужине путева за излазне мреже – добијате 0 поена за тај тест пример.



Проблем 1. Речник

Дат је низ речи дужине n којег ћемо у даље тексту звати речник. Речи су састављене од малих слова енглеског алфабета. За текст S који је такође састављен од малих слова енглеског алфабета, кажемо да има разбијање (r_1, r_2, \dots, r_m) уколико:

- r_i припада речнику за $i \in [1, m]$
- $r_1 r_2 \dots r_m = S$ (конкатенација, надовезивање речи)

За речник кажемо да је недвосмислен уколико свака реч која се може добити надовезивањем речи из речника има јединствено разбијање.

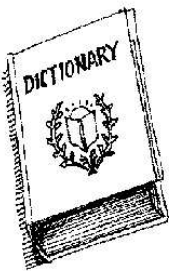
Написати програм који за дати речник испитује да ли је двосмислен или не.

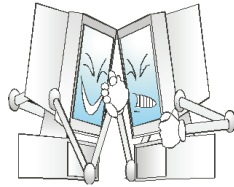
Улаз. (Улазни подаци се учитавају из датотеке `recnik.in`.) У првом реду улазне датотеке налази се природни број $T \leq 10$ који означава број примера. Сваки пример је описан на следећи начин: у првом реду се налази природан број n ($2 \leq n \leq 100$) који означава број речи у речнику. Наредних n линија описују речник (у сваком реду је дата по једна реч). Између примера неће бити празних редова. Речи ће бити састављене од малих слова енглеског алфабета и њихова дужина неће бити већа од 1000.

Издаз. (Издазни подаци се исписују у датотеку `recnik.out`.) За сваки пример у узлазној датотеци, у редоследу којим су дати на улазу, штампати у посебном реду `dvosmislen` уколико је речник двосмислен; у супротном штампати `nedvosmislen`.

Пример 1.

<code>recnik.in</code>	<code>recnik.out</code>
2	<code>dvosmislen</code>
4	<code>nedvosmislen</code>
<code>andre</code>	
<code>and</code>	
<code>jko</code>	
<code>rejko</code>	
3	
<code>a</code>	
<code>ab</code>	
<code>bb</code>	





Проблем 1. NishTel

Компанија *NishTel* тренутно уграђује широм земље свој најновији хит - брзе једносмерне међуградске телефонске линије. До сада су направили m једносмерних линија између неких од n градова. Свака линија повезује нека два града и уколико постоји линија од града A до града B , тада је преко ње могуће из града A звати град B али не и обратно (то је једна од мана система али линије су много јефтиније од старих, а то је оно што је битно). Сваку једносмерну телефонску линију карактерише један број t_i - време (у микросекундама) потребно да се успостави веза између одговарајућих градова.

Град A може да зове град B и индиректно, ако постоји неки низ градова, где је A први а B последњи, и између свака два узастопна града постоји телефонска линија у одговарајућем смеру. Тада је време за успостављање везе између A и B једнако збиру времена за успостављање свих међувеза на путу. Уколико постоји више начина да се успостави веза између нека два града, оператери увек бирају онај код кога је време успостављања минимално.

Познато је да *NishTel* има центре у два града (*NishB* и *ABNish*) као и да је из сваког од та два града могуће звати (директно или индиректно) било који други град. Такође је познато да је *NishTel* запао у дугове и да мора да сруши неке од m линија које су поставили. Они желе да сруше **што више** линија али тако да **минимална времена** потребна за успостављање веза од градова *NishB* и *ABNish* до сваког од осталих градова **остану иста** (укључујући и времена између два центра). Помозите *NishTel*-у да испуни свој план и спасе се стечаја.

Улаз. (Улазни подаци се учитавају из датотеке `nishtel.in`.) У првом реду улазне датотеке налазе се 4 природна броја n , m , A и B који представљају, редом, број градова, број телефонских линија и редне бројеве градова *NishB* и *ABNish* ($n \leq 10^4$, $m \leq 10^5$). Градови су нумерисани бројевима од 1 до n . У наредних m редова налазе се по три природна броја a_i , b_i и t_i који означавају да постоји линија од града a_i до града b_i и да је потребно t_i микросекунди за успостављање везе ($a_i \neq b_i$, $t_i \leq 10^5$). Између два града могу постојати више телефонских линија.

Излаз. (Излазни подаци се исписују у датотеку `nishtel.out`.) У првом и једином реду излазне датотеке исписати највећи број телефонских линија које је могуће срушити.

Пример 1.

<code>nishtel.in</code>	<code>nishtel.out</code>
4 6 1 2	2
1 2 1	
2 1 5	
2 4 1	
1 4 2	
1 3 3	
4 3 2	
2 1 4	

Објашњење. Рушењем друге и четврте телефонске линије, минимална растојања од центара до осталих градова остају иста, док је то немогуће ако срушимо три или више линија.

Напомена. У 50% тест примера биће $n \leq 10^3$.