

Ниш, 07. март 2010.

Проблем 1. Дворац

Окорели богаташ Божа жели да сагради нови дворац. Како је он веома богат, а ту чињеницу жели свима да покаже, решио је да дворац сагради на три хектара земље и то баш три најскупља хектара. Наиме, добио је мапу дела града у коме жели да сагради дворац. Мапа је облика матрице димензије $n \times m$ при чему свако поље представља један хектар земљишта. За сваки хектар је дата његова цена у милионима евра. Божа сада жели да изабере три повезана поља (два поља су повезана уколико имају заједничку ивицу) тако да је њихова укупна цена максимална.

Помозите Божи да реши овај проблем.

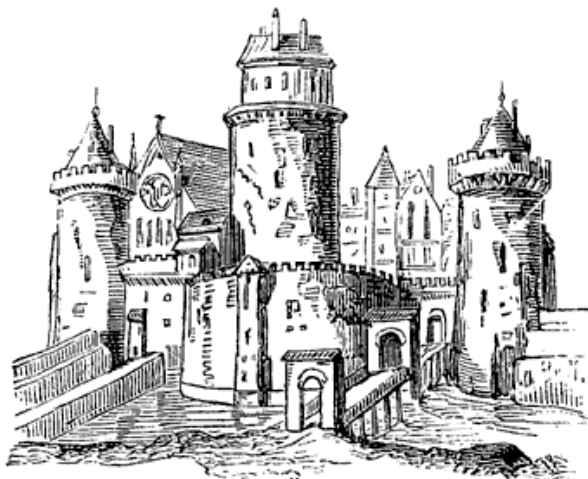
Улаз. (Улазни подаци се налазе у датотеци `dvorac.in`) У првом реду улазне датотеке налази се два природана броја n и m - димензије мапе града ($2 \leq n, m \leq 100$). У следећих n редова налазе се по m природних бројева, не већих од 1000, који представљају цене одговарајућих хектара.

Изназ. (Изназне податке уписати у датотеку `dvorac.out`) У првом и једином реду излазне датотеке исписати максималну цену три сукцесивна хектара.

Пример 1.

<code>dvorac.in</code>	<code>dvorac.out</code>
3 3	24
1 3 7	
2 8 9	
2 2 2	

Објашњење. У датом примеру тражени хектари су на позицијама: (1,3), (2,2) и (2,3). Сума њихових цена је $7 + 8 + 9 = 24$.



Ниш, 07. март 2010.

Проблем 2. Чекови

Џика ради као продавац компјутерске опреме у *Hi-Tech*-у. Једног дана у радњу му долази окорели богаташ Божа, који жели да купи нови лаптоп. Међутим, Џика баш и не воли Божу (јер је овај богат) па је решио да му подвали.

Џика зна да Божа има n чекова и позната му је вредност сваког чека (чекови су већ попуњени и вредности на њима се не могу мењати). На основу тих података он жели да каже цену тако да Божа не може да исплати тачан износ (познато је да уколико Божа помоћу својих чекова не може да исплати тачан износ, истог момента одустаје од куповине и изнервиран напушта објекат). Да не би било превише сумњиво, таква цена мора бити природан број и мора бити минимална. Помозите Џики да одреди такву цену.

Улаз. (Улазни подаци се налазе у датотеци `cekovi.in`) У првом реду улазне датотеке налази се природан број n - број Божиних чекова ($n \leq 10^5$). У следећем реду налазе се n природних бројева a_i , који представљају вредности Божиних чекова ($a_i < 10^9$).

Излаз. (Излазне податке уписати у датотеку `cekovi.out`) У првом и једином реду излазне датотеке исписати тражену цену.

Пример 1.

```
cekovi.in
5
20 1 1 7 3
```

```
cekovi.out
6
```





Ниш, 07. март 2010.

Проблем 3. Премутација

Нека је p низ бројева од 1 до n у произвољном поретку, другим речима пермутација. Низ d дефинишемо као:

$$d[k] = \begin{cases} 1 & , \text{ ако првих } k \text{ елемената низа } p \text{ представљају пермутацију бројева од 1 до } k \\ 0 & , \text{ иначе} \end{cases}$$

Дат је низ d и неки елементи пермутације p . Написати програм који одређује преостале елементе низа p .

Улаз. (Улазни подаци се налазе у датотеци `perm.in`) У првом реду улазне датотеке налазе се два природна броја n и m , који представљају дужину пермутације и број познатих елемената низа p редом ($1 \leq n \leq 10^5$, $0 \leq m \leq n$). У другом реду се налазе n бинарних бројева, 0 или 1, који представљају елементе низа d .

Наредних m редова садрже по два природна броја x и y , $1 \leq x, y \leq n$, која означавају да је x -ти елемент пермутације p једнак y . Ти подаци неће бити међусобно противуречни.

Издаз. (Издазне податке уписати у датотеку `perm.out`) У првом и једином реду издазне датотеке исписати тражену пермутацију. Уколико решење не постоји исписати *'ne postoji'*.

Пример 1.

<code>perm.in</code>	<code>perm.out</code>
5 1	2 3 1 5 4
0 0 1 0 1	
2 3	

Пример 2.

<code>perm.in</code>	<code>perm.out</code>
8 3	ne postoji
0 0 0 1 0 0 1 1	
1 2	
5 6	
2 7	

Објашњење. У другом примеру, како је из улаза $p[2] = 7$ тада никако првих четири елемента неће моћи да чине пермутацију бројева од 1 до 4.

Проблем 4. Јабуре

Два брата имају башту и у њој n дрвета јабука. За свако дрво је познато колико јабука даје годишње. Једног дана одлучено је да се башта подели на два дела једном правом линијом, при чему та линија не сме да пролази ни кроз једно дрво. Једном од браће ће тада припасти све јабуке које може убрати са једне стране линије, а другом ће припасти све јабуке које може убрати са супротне стране.

Помозите браћи да што праведније поделе башту, тј. да апсолутна разлика између јабука које ће припасти браћи буде најмања могућа.

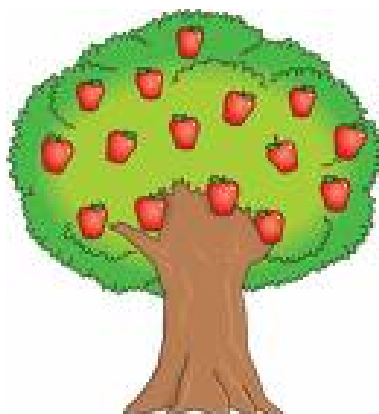
Улаз. (Улазни подаци се налазе у датотеци `jabuke.in`) У првом реду улазне датотеке налази се природан број n који представља број дрвета ($n \leq 300$). У наредних n редова налазе се по 3 цела броја x_i , y_i и c_i ($-10^4 \leq x_i, y_i \leq 10^4$, $1 \leq c_i \leq 10^6$) где су (x_i, y_i) координате i -тог стабла, а c_i број јабука које i -то стабло даје годишње. Пречник стабла је занемарљив и не постоје два стабла са истим координатама.

Изаз. (Изазне податке уписати у датотеку `jabuke.out`) У првом и једином реду излазне датотеке исписати број A - минималну апсолутну разлику између броја јабука које ће припасти браћи при оптималној подели.

Пример 1.

<code>jabuke.in</code>	<code>jabuke.out</code>
4	1
0 0 1	
1 0 8	
1 1 4	
0 1 2	

Објашњење. Могуће је повући праву тако да прво, треће и четврто дрво припадну једном, а друго дрво другом брату. Тада први брат има $1 + 2 + 4 = 7$, а други 8 јабука, па је разлика $|7 - 8| = 1$. Мању разлику је немогуће добити.



Ниш, 07. март 2010.

Проблем 5. Лавиринт

Дат је 2D-лабиринт приказан у облику матрице димензије $n \times m$. Поља могу бити празна или се на њима могу налазити зидови. Неки од празних поља садрже врата или кључеве. Постоји четири различитих типова врата и одговарајућих кључева: плава, жута, црвена и зелена. Кључ отвара само врата одговарајуће боје (логично зар не?).

Дозвољено је кретање међу повезаним пољима (поља су повезана уколико имају заједничку ивицу). Није дозвољено стати на поље на којем се налази зид, као ни изаћи из граница лабиринта. Уколико се на пољу налазе врата, на то поље је дозвољено стати једино уколико је претходно обиђено поље на којем се налази одговарајући кључ.

Дато је почетно поље на које се налазите као и крајње (поље које представља излаз). За дату конфигурацију лабиринта наћи минимални број потеза којима излазите из истог.

Улаз. (Улазни подаци се налазе у датотеци `lavirint.in`) У првом реду улазне датотеке налазе се два природана броја n и m који представљају димензије лабиринта ($1 \leq n, m \leq 100$). У наредних n редова налази се по m карактера (спојених) који описују конфигурацију лабиринта. Једини карактери који се могу наћи у датотеци као и њихова значења, описани су у доњој табели.

Карактер	Значење
#	зид
.	празно поље
*	почетно поље
X	крајње поље
B, Y, R, G	плава, жута, црвена и зелена врата
b, y, r, g	плави, жути, црвени и зелени кључ

Обратити пажњу на то да је могуће имати више крајњи позиција као и ниједно. Такође, у лабиринту се могу наћи више врата и кључева исте боје, а може се десити и да их нема. Стартно поље је јединствено.

Излаз. (Излазне податке уписати у датотеку `lavirint.out`) У првом и једином реду излазне датотеке исписати минимални број потеза потребних да се изађе из лабиринта или -1 уколико то није могуће.

Пример 1.

`lavirint.in`

`3 20`

`#####`

`#XY.gBr.*.Rb.G.GG.y#`

`#####`

`lavirint.out`

`45`

