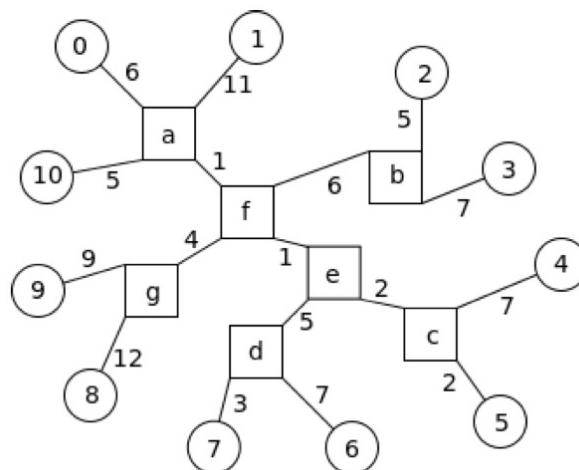


# Towns

В Казахстан има  $N$  села, номерирани от  $0$  до  $N - 1$  и неизвестен брой градове. Селата и градовете наричаме с общото име *селища*.

Всички селища са свързани в мрежа от двупосочни магистрали, като всяка магистрала на мрежата свързва две различни селища. Всеки две селища са свързани с най-много една магистрала. Села са само тези селища, които са свързани директно с точно един град. Няма градове, които да не са свързани с поне едно селище или да са свързани с точно две селища, т.е. всеки град е свързан с поне три селища. Освен това, за всеки две селища има единствен начин да се стигне от едното до другото така, че по никоя от магистралите не се минава повече от един път.

На фигурата е показана мрежа с **11** села и **7** града. Селата са представени с кръгчета и означени с цели числа, а градовете са представени с квадратчета и означени с букви.



Всяка магистрала има дължина — положително цяло число. Разстоянието между две селища е сумата от дължините на всички магистрали, по които трябва да се премине за да се стигне от едното селище до другото.

За всеки град  $C$  да означим с  $r(C)$  разстоянието до най-отдалеченото от него село. Градът  $C$  с минимално  $r(C)$  наричаме *център*. Да означим с  $R$  максималното разстояние от центъра до някое от селата, т.е.  $R$  е най-малката от всички стойности  $R(C)$ .

В примера на фигурата, най-отдалечено от града  $a$  е селото  $8$  и  $r(a) = 1 + 4 + 12 = 17$ . За града  $g$ , също имаме  $r(g) = 17$ , като едно от селата на такова разстояние от  $g$  е  $6$ . Единствен център на мрежата, в този пример, е градът  $f$  с  $r(f) = 16$ . Затова, в случая,  $R$  е  $16$ .

Ако премахнем центъра, мрежата ще се разпадне на няколко отделни подмрежи. Ще наричаме един център на мрежата *балансиран*, ако след като го премахнем, всяка от получените се

подмрежи има най-много  $\lfloor N/2 \rfloor$  села, където  $\lfloor x \rfloor$  е най-голямото цяло не надминаващо  $x$  (броят на градовете в подмрежата е без значение).

В примера, ако премахнем центъра  $f$ , мрежата ще се разпадне на 4 подмрежи със следните подмножества от села:  $\{0, 1, 10\}$ ,  $\{2, 3\}$ ,  $\{4, 5, 6, 7\}$  и  $\{8, 9\}$ . Никоя от тези мрежи не е с повече от  $\lfloor 11/2 \rfloor = 5$  села, и значи  $f$  е балансиран център.

## Задача

В началото, единствената информация, която ще имате за мрежата от селища и магистрали е броят  $N$  на селата. Броят на градовете не е известен, както и кои селища са свързани с магистрали. Единственият начин да получите информация е, да задаване въпроси за разстоянията между двойки села.

Задачата е да се определи:

- За всяка подзадача: разстоянието  $R$ .
- За подзадачите от 3 to 6: има ли мрежата балансиран център.

Вашата програма трябва да имплементира функцията `hubDistance()`. Грейдърът ще провери правилността ѝ с няколко тестови примера (не повече от **100**) при едно изпълнение. За всеки тестов пример, грейдърът ще извика функцията `hubDistance()` веднъж. Направете така, че функцията да инициализира всички нужни променливи при всяко извикване.

- `hubDistance(N, sub)`
  - $N$ : брой на селата.
  - `sub`: номер на подзадачата (дефиницията на всяка подзадача е дадено в раздела Подзадачи).
  - Ако `sub` е 1 или 2, функцията може да върне  $R$  или  $-R$ .
  - Ако `sub` е повече от 2, тогава функцията трябва да върне  $R$ , ако мрежата има балансиран център или  $-R$ , ако няма.

Вашата функция `hubDistance()` може да получава информация за мрежата чрез извикване на функцията `getDistance(i, j)`, която връща разстоянието между селата  $i$  and  $j$ . Забележете, че ако  $i$  съвпада с  $j$ , функцията връща **0**. Тя връща **0** и ако бъде извикана с невалиден аргумент.

## Подзадачи

Във всеки тестов пример:

- $N$  е между **6** и **110**, включително.
- Разстоянието между всеки две села е между 1 и 1,000,000, включително.

Броят на заявките, които програмата може да направи при едно извикване е ограничен, според номера на подзадачата, както е показано в таблицата по-долу. Ако програмата опита да направи повече от позволените заявки, изпълнението ще бъде прекъснато и ще се приеме, че е пресметнала неверен резултат.

подзадача	точки	брой заявки	търсене на балансиран център	Допълнителни ограничения
1	13	$\frac{n(n-1)}{2}$	НЕ	няма
2	12	$\lceil \frac{7n}{2} \rceil$	НЕ	няма
3	13	$\frac{n(n-1)}{2}$	ДА	няма
4	10	$\lceil \frac{7n}{2} \rceil$	ДА	от всеки град излизат <i>точно</i> три магистрали
5	13	$5n$	ДА	няма
6	39	$\lceil \frac{7n}{2} \rceil$	ДА	няма

Забележете, че  $\lceil x \rceil$  е най малкото цяло по-голямо от или равно на  $x$ .

### Примерен грейдър

Забележете, че номерът на подзадачата е част от входните данни. Примерният грейдър променя поведението си в съответствие с номера на подзадачата.

Примерният грейдър чете входа от файла `towns.in`, който има следният формат:

- ред 1: Брой на тестовите примери.
- ред 2:  $N_1$ , брой на селата в първия тестов пример.
- следващите  $N_1$  реда:  $j$ -тото число ( $1 \leq j \leq N_1$ ) в  $i$ -тия от тези редове ( $1 \leq i \leq N_1$ ) е разстоянието между селата  $i - 1$  and  $j - 1$ .
- Следват останалите тестови примери, форматираны по същия начин.

За всеки тестов пример, грейдърът извежда стойността, върната от функцията `hubDistance()` и броя на заявките за отделните тестови примери.

Входът за примера даден на фигурата е:

```

1
11
0 17 18 20 17 12 20 16 23 20 11
17 0 23 25 22 17 25 21 28 25 16
18 23 0 12 21 16 24 20 27 24 17
20 25 12 0 23 18 26 22 29 26 19
17 22 21 23 0 9 21 17 26 23 16
12 17 16 18 9 0 16 12 21 18 11
20 25 24 26 21 16 0 10 29 26 19
16 21 20 22 17 12 10 0 25 22 15
23 28 27 29 26 21 29 25 0 21 22
20 25 24 26 23 18 26 22 21 0 19
11 16 17 19 16 11 19 15 22 19 0

```

Този формат е доста различен от задаването на мрежата чрез описване на магистралите. Забележете, че можете да изменяте примерния грейдър, за да приема различен формат.