

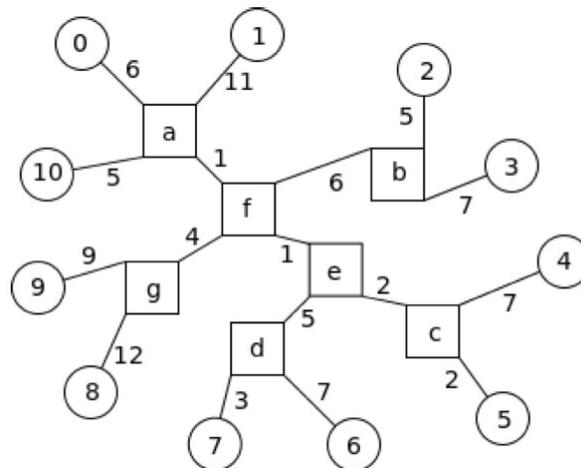
Towns

O Cazaquistão tem N vilarejos, numerados de 0 a $N - 1$. Há também um número desconhecido de cidades (cidades grandes). Os vilarejos e as cidades são conjuntamente chamados de *povoações* (*regiões urbanas*).

Todas as povoações do Cazaquistão são conectadas por uma única rede de autoestradas bidirecionais. Cada autoestrada conecta duas povoações distintas, e cada par de povoações é conectada diretamente por no máximo uma autoestrada. Para cada par de povoações a e b existe um único caminho para ir de a a b usando as autoestradas, sem repetir nenhuma.

Sabe-se que cada vilarejo é diretamente conectado a uma única povoação, e cada cidade é diretamente conectada a três ou mais povoações.

A figura a seguir mostra uma rede de **11** vilarejos e **7** cidades. Vilarejos são indicados por círculos e rotulados por inteiros, cidades são representadas por quadrados e rotuladas por letras.



Toda autoestrada tem um comprimento inteiro positivo. A distância entre duas povoações é a soma mínima dos comprimentos das autoestradas que devem ser percorridas para ir de uma povoação à outra.

Para cada cidade C podemos calcular a distância $r(C)$ para o vilarejo mais distante de C . Uma cidade C é um *centro* se a distância $r(C)$ é a menor, consideradas todas as cidades. A distância entre um centro e o vilarejo mais distante do centro será denotada por R . Assim, R é o menor de todos os valores $r(C)$.

No exemplo acima a povoação mais distante da cidade a é o vilarejo 8 , e a distância entre elas é $r(a) = 1 + 4 + 12 = 17$. Para a cidade g também temos $r(g) = 17$. (Uma das povoações que estão mais distantes de g é a povoação 6 .) O único centro no exemplo acima é a cidade f , onde $r(f) = 16$. Portanto, no exemplo acima, R é 16 .

A remoção de um centro divide a rede em várias componentes conexas. Um centro é *balanceado* se cada uma dessas componentes contém no máximo $\lfloor N/2 \rfloor$ vilarejos. (Enfatizamos que não estamos contando as cidades.) Note que $\lfloor x \rfloor$ denota o maior inteiro que não é maior do que x .

No nosso exemplo, a cidade **f** é um centro. Se removermos **f**, a rede será quebrada em quatro componentes conexas. Estas quatro componentes consistem dos seguintes conjuntos de vilarejos: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$ e $\{8, 9\}$. Nenhuma dessas componentes tem mais do que $\lfloor 11/2 \rfloor = 5$ vilarejos, portanto a cidade **f** é um centro balanceado.

Tarefa

Inicialmente, a única informação que você tem sobre a rede de povoações e autoestradas é o número N de vilarejos. Você não sabe o número de cidades. Você também não sabe nada sobre o mapa das autoestradas no país. Você somente pode obter mais informações fazendo consultas sobre as distâncias entre pares de vilarejos.

Sua tarefa é determinar:

- Em todas as subtarefas: a distância R .
- Nas subtarefas 3 a 6: se existe um centro balanceado na rede.

Você precisa implementar a função `hubDistance`. O avaliador irá submeter vários casos de teste em cada execução. O número de casos de teste por execução é no máximo **40**. Para cada caso de teste o avaliador irá chamar sua função `hubDistance` exatamente uma vez. Certifique-se de sua função inicializa todas as variáveis necessárias cada vez que for chamada.

- `hubDistance(N, sub)`
 - N : o número de vilarejos.
 - `sub`: o número da sub tarefa (explicado na seção Subtarefas).
 - Se `sub` for 1 ou 2, a função pode retornar ou R ou $-R$.
 - Se `sub` for maior do que 2, se existir um centro balanceado então a função deve retornar R , caso contrário deve retornar $-R$.

Sua função `hubDistance` pode obter informação sobre a rede de autoestradas chamando a função `getDistance(i, j)` do avaliador. Esta função retorna a distância entre os vilarejos i e j . Note que se i e j forem iguais, a função retorna **0**. Também retorna **0** quando os argumentos forem inválidos.

Subtarefas

Em cada caso de teste:

- N está entre **6** e **110** inclusive.
- A distância entre quaisquer dois vilarejos distintos está entre 1 e 1,000,000 inclusive.

O número de consultas que seu programa pode fazer é limitado. O limite varia de acordo com a sub tarefa, conforme a tabela abaixo. Se seu programa tenta exceder o limite do número de consultas, será terminado e será considerado que você deu uma resposta incorreta.

subtarefa	pontos	número de consultas	determinar centro balanceado	restrições adicionais
1	13	$\frac{N(N-1)}{2}$	NÃO	nenhuma
2	12	$\lceil \frac{7N}{2} \rceil$	NÃO	nenhuma
3	13	$\frac{N(N-1)}{2}$	SIM	nenhuma
4	10	$\lceil \frac{7N}{2} \rceil$	SIM	cada cidade é ligada a <i>exatamente</i> três povoações
5	13	$5N$	SIM	nenhuma
6	39	$\lceil \frac{7N}{2} \rceil$	SIM	nenhuma

Note que $\lceil x \rceil$ denota o menor inteiro que é maior do que ou igual a x .

Avaliador exemplo

Note que o número da subtarefa é parte da entrada. O avaliador exemplo muda seu comportamento de acordo com o número da subtarefa.

A entrada para o avaliador exemplo é o arquivo `towns.in`, no seguinte formato:

- linha 1: O número da subtarefa e o número de casos de teste.
- linha 2: N_1 , o número de vilarejos no primeiro caso de teste.
- as N_1 linhas seguintes: o j -ésimo número ($1 \leq j \leq N_1$) na i -ésima dessas linhas ($1 \leq i \leq N_1$) é a distância entre os vilarejos $i - 1$ e $j - 1$.
- Seguem os próximos casos de teste. São dados no mesmo formato do primeiro caso de teste.

Para cada caso de teste, o avaliador exemplo imprime o valor devolvido pelo `hubDistance` e o número de chamadas feitas, em linhas separadas.

O arquivo de entrada que corresponde ao exemplo acima é:

```
1 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
```

Este formato está longe de especificar a lista de autoestradas. Note que você pode modificar avaliadores exemplos, de forma a usar um formato de entrada diferente.