Mötesplats

Problem ID: motesplats

Ett **udda** antal (N stycken) vänner bor på var sin nod i ett träd med N noder. Ett *träd* är en sammanhängande graf med exakt $N - 1$ kanter.

Vännerna vill nu alla träffas på en nod i trädet. De har kommit fram till att de vill mötas på den noden som minimerar summan av distanserna till vännerna, och har frågat dig om du kan hjälpa dem att hitta denna optimala mötesplats. *Distansen* $\text{dist}(a, b)$ mellan två noder a och b i trädet är antalet kanter på vägen mellan a och b. Så formellt sett vill du hitta noden x som minimerar $\sum_{i=1}^{n} \text{dist}(x, i)$.

Detta tänker du är ett lätt problem, och börjar genast koda en lösning. Men det finns en tvist! Vännerna har dåligt minne och kommer inte ihåg hur trädet ser ut. Dock kommer de ihåg följande: givet tre **olika** vänner a, b, c kan de med säkerhet säga att de brukar mötas (när det bara är dem tre) på plats x, där x är noden som minimerar $\text{dist}(x, a) + \text{dist}(x, b) + \text{dist}(x, c)$. Notera att x inte nödvändigtvis är en av noderna a, b eller c.

I detta problem läser du alltså inte in grafen direkt i indatan, utan får istället fråga upp till $Q - 1$ frågor på formen: *Var brukar vännerna a, b, c mötas?* Med hjälp av dessa frågor vill du hitta den optimala mötesplatsen.

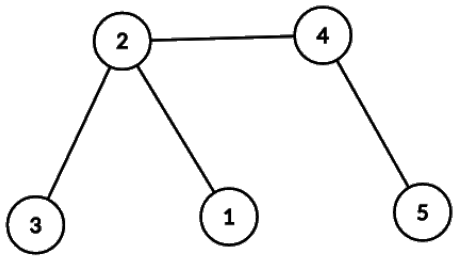

Figure 1: Illustration av trädet i exempelfallet. Nod 2 är den optimala mötesplatsen.

Interaktion

I första raden av indatan finns två heltal N och Q, antalet noder i trädet och antalet queries du får göra. Det är garanterat att N är udda och att $Q = 500\,000$ i alla testfall.

Sedan får du göra följande typer av queries:

- "`? a b c`" där $1 \leq a, b, c \leq N$ är 3 **olika** vänner. Därefter kommer domarprogrammet svara med en rad med ett heltal x ($1 \leq x \leq N$), noden som minimerar $\text{dist}(x, a) + \text{dist}(x, b) + \text{dist}(x, c)$.
- "`! x`" där $1 \leq x \leq N$ är den optimala mötesplatsen. Efter att du har printat denna query ska ditt program avsluta utan att skriva ut något mer. Om x är noden som minimerar $\sum_{i=1}^{n} \text{dist}(x, i)$ får du AC på testfallet, annars WA.

Om du gör fler än Q queries (inklusive den sista på formen "`! x`") kommer ditt program avslutas och du får WA på testfallet.

Se till att flusha outputen efter varje query, annars kan du få Time Limit Exceeded. I C++ kan detta göras med exempelvis `cout « flush` eller `fflush(stdout);` i Python med `stdout.flush();` och i Java med `System.out.flush()`.

Poängsättning

Din lösning kommer att testas på en mängd testfallsgrupper. För att få poäng för en grupp så måste du klara alla testfall i gruppen.

Grupp	Poängvärde	Gränser
1	10	$3 \leq N \leq 99$ och alla noder har grad högst 2, dvs trädet är en linje
2	12	$3 \leq N \leq 999$ och alla noder har grad högst 2, dvs trädet är en linje
3	21	$3 \leq N \leq 24\,999$ och alla noder har grad högst 2, dvs trädet är en linje
4	14	$3 \leq N \leq 99$
5	15	$3 \leq N \leq 999$
6	28	$3 \leq N \leq 24\,999$

Exempelfall

Sample Interaction 1

Read	Write
`5 500000`	
	`? 1 2 3`
`2`	
	`? 5 2 4`
`4`	
	`? 3 5 1`
`2`	
	`! 2`