



## PROBLEMA INGEGNERI

È arrivato il momento della manutenzione annuale dei server a Volterra. Una parte della manutenzione può essere svolta dagli umani, ma i cavi che collegano la rete scorrono attraverso una serie di condotti sotterranei troppo stretti per essere attraversati! Per fortuna, un brillante ingegnere ha avuto l'idea di far controllare i computer a dei topi addestrati.

Ogni topo può entrare nella rete da un computer, percorrere i cavi nei condotti sotterranei ed uscire da un altro computer, controllando tutti i computer attraversati. Poiché i topi sono piuttosto pigri, ognuno di essi seguirà un singolo percorso semplice tra il punto di ingresso e quello di uscita.

Il team di manutenzione vuole far controllare alcuni computer ai topi, in modo che tra quelli non controllati rimasti la configurazione dei codici di sicurezza sia "abbastanza bilanciata". Il tuo compito è determinare il numero minimo di topi necessari.

■ **PROBLEMA** Ci sono  $N$  computer, numerati da  $0$  a  $N - 1$ , collegati da  $N - 1$  cavi. La rete è un albero (connesso e privo di cicli).

Un singolo topo può controllare tutti i computer lungo l'unico percorso semplice tra il computer in cui entra e quello da cui esce. Formalmente, se un topo entra nel computer  $S$  ed esce dal computer  $T$  ( $S, T \in \{0, \dots, N - 1\}$ ), allora controllerà tutti i computer  $v_1, v_2, \dots, v_k$  tali che:

- $v_1 = S$  e  $v_k = T$ ,
- per ogni  $1 \leq i < k$ , i computer  $v_i$  e  $v_{i+1}$  sono collegati direttamente da un cavo,
- $k$  è minimale.

Un cavo può essere visitato da più topi, e un computer può essere controllato da più topi.

Ogni computer  $i \in \{0, \dots, N - 1\}$  ha un codice associato  $C_i \geq 1$ . Il team di manutenzione decide una differenza massima accettabile  $D$ . Dopo che tutti i topi hanno terminato, sia  $R \subseteq \{0, \dots, N - 1\}$  l'insieme dei computer *rimanenti* (cioè non controllati). Il team vuole garantire che  $|C_i - C_j| \leq D$  per ogni coppia  $i, j \in R$ .

In altre parole, la differenza tra il codice massimo e quello minimo tra i computer *rimanenti* deve essere al massimo  $D$ .

Ogni topo può controllare un solo percorso. Vuoi minimizzare il numero di topi utilizzati.

■ **IMPLEMENTAZIONE** Devi implementare la funzione

```
int solve(int N, int D, std::vector<int> C,  
          std::vector<int> P, std::vector<int> Q)
```

La funzione riceve come parametri:  $N$ , il numero di computer;  $D$ , la differenza massima accettabile;  $C$ , i codici dei computer; e due vettori di lunghezza  $N - 1$ ,  $P$  e  $Q$ , tali che per ogni  $0 \leq i \leq N - 1$  esiste un cavo tra i computer  $P_i$  e  $Q_i$ .

La funzione deve restituire il numero minimo di topi necessari affinché, dopo che tutti i topi hanno controllato i loro percorsi, i computer rimanenti soddisfino:

$$\max_{i \in R} C_i - \min_{i \in R} C_i \leq D.$$



- **ASSUNZIONI**
- ◆  $1 \leq N \leq 200\,000$ .
  - ◆  $1 \leq C_i \leq 1\,000\,000\,000$  per ogni  $0 \leq i \leq N - 1$ .
  - ◆  $1 \leq D \leq 1\,000\,000\,000$ .
  - ◆  $0 \leq p_i, q_i < N$ ,  $p_i \neq q_i$ , e tutte le coppie  $(p_i, q_i)$  sono distinte.

■ **SUBTASK**

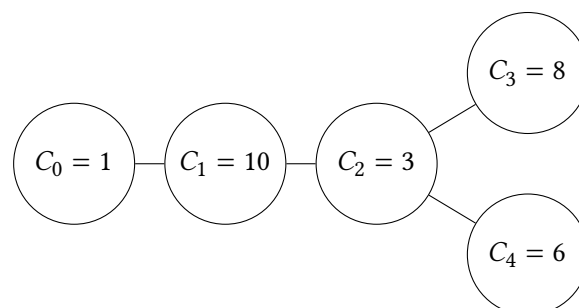
| # | Punti | Assunzioni                                                              |
|---|-------|-------------------------------------------------------------------------|
| 1 | 7     | $N \leq 20$ e $1 \leq C_i \leq 50$ per ogni $0 \leq i \leq N - 1$ .     |
| 2 | 6     | $N \leq 1000$ e $1 \leq C_i \leq 1000$ per ogni $0 \leq i \leq N - 1$ . |
| 3 | 11    | $N \leq 1000$ .                                                         |
| 4 | 16    | $p_i = 0, q_i = i + 1$ per ogni $0 \leq i < N - 1$ .                    |
| 5 | 26    | $N \leq 50000$ .                                                        |
| 6 | 34    | Nessuna limitazione aggiuntiva.                                         |



## ESEMPI

| Input                               | Output |
|-------------------------------------|--------|
| 5 3                                 | 1      |
| 1 10 3 8 6                          |        |
| 0 1                                 |        |
| 1 2                                 |        |
| 2 3                                 |        |
| 2 4                                 |        |
| 20 30                               | 3      |
| 13 36 11 35 4 9 42 9 1 4 11 3 15 31 |        |
| → 46 41 31 17 11 12                 |        |
| 19 5                                |        |
| 19 0                                |        |
| 19 13                               |        |
| 19 9                                |        |
| 19 4                                |        |
| 19 10                               |        |
| 5 1                                 |        |
| 19 18                               |        |
| 0 7                                 |        |
| 5 8                                 |        |
| 19 12                               |        |
| 5 17                                |        |
| 13 16                               |        |
| 5 14                                |        |
| 13 3                                |        |
| 19 6                                |        |
| 5 15                                |        |
| 5 2                                 |        |
| 4 11                                |        |

Nel primo esempio ci sono  $N = 5$  computer e  $D = 3$ . La rete è illustrata nella seguente figura:



Una possibile strategia consiste nell'inviare un singolo topo su un percorso che inizia nel computer 0 e termina nel computer 3, passando lungo il percorso  $0 - 1 - 2 - 3$ . Dopo che questo percorso è stato controllato, la rete risulta sicura.