

Computer-Netzwerk

Die N Computer im Computerraum einer Schule sind mit Kabeln vernetzt. Zwischen je zwei verschiedenen Computern gibt es entweder eine direkte Verbindung per Kabel oder eine Verbindung über “Drittcomputer”. Nachrichten, die zwischen zwei Computern verschickt werden, nehmen immer die kürzeste Route, mit minimal vielen Drittcomputern.

Adam und Billy nutzen verschiedene Computer a und b in diesem Raum und wollen die kürzeste Route zwischen ihren Computern bestimmen. Die Kabelverbindungen kennen sie nicht, aber sie können Nachrichten zwischen den Computern verschicken und dabei die Anzahl der Drittcomputer auf der Route zwischen den Computern bestimmen.

Hilf Adam und Billy:

Aufgabe

Bestimme die kürzeste Route zwischen Computern a und b mit Hilfe einer begrenzten Anzahl von Nachrichten.

Implementierung

Implementiere eine Prozedur `findRoute(N, a, b)` mit den folgenden Parametern:

- N – die Anzahl der Computer (die Computer sind von 1 bis N nummeriert)
- a, b – die Nummern von Adams and Billys Computern ($a \neq b$ und $1 \leq a, b \leq N$)

Deine Prozedur `findRoute` kann die Funktion `ping(i, j)` aufrufen. Deren Parameter sind die Nummern zweier verschiedener Computer ($i \neq j$ and $1 \leq i, j \leq N$). Sie liefert die Anzahl der Drittcomputer auf der Route von i nach j .

Deine Prozedur `findRoute` muss eine kürzeste Route einer Nachricht von a nach b angeben, indem sie wiederholt die Prozedur `travelTo(k)` aufruft. Deren Parameter ist die Nummer des jeweils nächsten Computers auf der Route ($1 \leq k \leq N$). Die Nachricht startet bei a , und bei einem Aufruf von `travelTo(k)` geht sie zu Computer k .

Damit ein Testfall als gelöst gilt, müssen – zusätzlich zu Laufzeit- und Speicherlimits – folgende Bedingungen eingehalten werden:

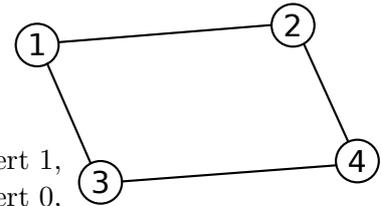
- Wenn die Prozedur `findRoute` terminiert, muss die Nachricht bei b sein.
- Zwei Computer, die auf der Route der Nachricht unmittelbar aufeinander folgen, müssen direkt per Kabel verbunden sein.
- Die Nachricht muss eine kürzeste Route nehmen.
- Die Anzahl M der Aufrufe von `ping` darf eine gegebene Grenze nicht überschreiten (s. Abschnitt Bewertung).
- Die Funktion `ping` und die Prozedur `travelTo` dürfen nur mit erlaubten Parameterwerten aufgerufen werden.

Beispiel

Schau dir das Beispiel in der Abbildung unten an (Kreise sind Computer, Linien sind Kabel). Es gibt $N = 4$ Computer, Adam und Billy nutzen die Computer $a = 1$ und $b = 4$.

Nun wird deine Prozedur `findRoute` aufgerufen:

```
findRoute(4, 1, 4).
```



Sie könnte sich z.B. so verhalten:

```
ping(1, 4) wird aufgerufen und liefert 1,  
ping(1, 2) wird aufgerufen und liefert 0,  
ping(2, 4) wird aufgerufen und liefert 0.
```

Diese Information genügt, um die folgende kürzeste Route von 1 nach 4 zu bestimmen: $1 \rightarrow 2 \rightarrow 4$. Diese Route sollte nun so angegeben werden:

```
travelTo(2) wird aufgerufen,  
travelTo(4) wird aufgerufen,  
findRoute terminiert.
```

Bewertung

In allen Subtasks gilt: $2 \leq N \leq 1000$

Subtask 1 (25 Punkte): zwischen je zwei Computern gibt es genau eine kürzeste Route; M darf nicht größer sein als $2N$.

Subtask 2 (25 Punkte): M darf nicht größer sein als N^2 .

Subtask 3 (25 Punkte): M darf nicht größer sein als $4N$.

Subtask 4 (25 Punkte): M darf nicht größer sein als $2N$.

Constraints

Time Limit: 1 s.

Memory Limit: 64 MB.

Experimentieren

Der Beispielgrader auf deinem Computer liest vom Standardinput. Die erste Eingabezeile sollte vier Integer enthalten: N, a, b und das Limit für M (die Anzahl der ping-Aufrufe). Die nächsten N Zeilen sollten je N Integer enthalten, die das Netzwerk beschreiben: Die j -te Zahl in Zeile i ($i \neq j$) gibt die Anzahl der Drittcomputer auf einer kürzesten Route von i nach j an. Die Einträge für $i = j$ werden nicht beachtet.

Die folgende Eingabe entspricht dem obigen Beispiel, mit einem Limit von 100 für M .



```
4 1 4 100
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0
```