

## C. Sopsug

Problemname	Sopsug
Time Limit	5 Sekunden
Memory Limit	1 Gigabyte

Grushög ist ein noch nicht fertiggestellter Wohnbezirk ausserhalb von Lund. Momentan wird die notwendige Infrastruktur gebaut, einschliesslich der Müllabfuhr. Wie auch anderswo in Schweden wurde entschieden, eine *sopsug* (Abfallsauganlage) zu benutzen. Die Idee ist es, Müll unterirdisch mittels Luftdruck abzutransportieren.

Es gibt  $N$  Gebäude in Grushög, welche von 0 bis  $N - 1$  durchnummeriert sind. Deine Aufgabe ist es, einige dieser Gebäude mit Rohren zu verbinden. Wenn du ein Rohr von Gebäude  $u$  zu einem anderen Gebäude  $v$  legst, wird der Müll von  $u$  zu  $v$  transportiert (aber nicht in die andere Richtung). Dein Ziel ist es ein Netzwerk von  $N - 1$  Rohren zu bauen, sodass der ganze Müll in einem Gebäude ankommt. Anders gesagt soll das Netzwerk ein Baum mit Wurzel werden, wobei alle Rohre zur Wurzel hin gerichtet sind.

Es wurden jedoch schon  $M$  Rohre zwischen einigen Gebäuden gelegt. Diese *müssen* in deinem Netzwerk enthalten sein. Auch diese Rohre sind gerichtet, können also nur in eine Richtung benutzt werden.

Des Weiteren gibt es auch  $K$  Paare von Gebäuden, zwischen denen du keine Rohre legen darfst. Diese Paare sind geordnet. Das heisst, das wenn es nicht möglich ist ein Rohr zwischen  $u$  und  $v$  zu legen, dann kann es immer noch möglich sein, ein Rohr von  $v$  nach  $u$  zu legen.

### Eingabe

Die erste Zeile enthält drei ganzen Zahlen  $N$ ,  $M$  und  $K$ .

Die nächsten  $M$  Zeilen enthalten jeweils zwei unterschiedliche, ganze Zahlen  $a_i, b_i$ . Das heisst, dass es schon ein Rohr von  $a_i$  nach  $b_i$  gibt.

Die letzten  $K$  Zeilen enthalten jeweils zwei unterschiedliche, ganze Zahlen  $c_i, d_i$ . Das heisst, dass es nicht möglich ist ein Rohr von  $c_i$  nach  $d_i$  zu bauen.

Die  $M + K$  geordneten Paare in der Eingabe sind alle paarweise verschieden. Beachte, dass  $(u, v)$  und  $(v, u)$  als unterschiedliche Paare angesehen werden.

## Ausgabe

Wenn es keine Lösung gibt, gib "NO" aus.

Sonst gib  $N - 1$  Zeilen aus, wobei jede Zeile aus zwei ganzen Zahlen  $u_i, v_i$  besteht. Dies heisst, dass es ein Rohr von  $u_i$  nach  $v_i$  gibt. Du kannst die Rohre in einer beliebigen Reihenfolge ausgeben. Falls es mehrere Lösung gibt, darfst du eine beliebige ausgeben. Beachte, dass die bestehenden  $M$  Rohre Teil deiner Lösung sein müssen.

## Einschränkungen und Bewertung

- $2 \leq N \leq 300\,000$ .
- $0 \leq M \leq 300\,000$ .
- $0 \leq K \leq 300\,000$ .
- $0 \leq a_i, b_i \leq N - 1$  für  $i = 0, 1, \dots, M - 1$ .
- $0 \leq c_i, d_i \leq N - 1$  für  $i = 0, 1, \dots, K - 1$ .

Dein Programm wird an eine Menge von Testgruppen getestet, welche eine gewisse Anzahl an Punkten wert sind. Jede Testgruppe enthält eine Menge von Testfällen. Um die Punkte für eine Testgruppe zu erhalten, musst du alle Testfälle der Testgruppe lösen.

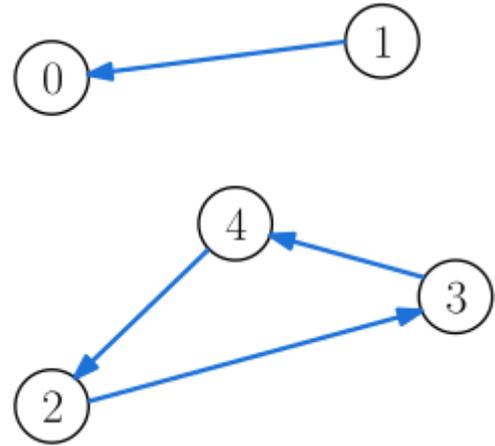
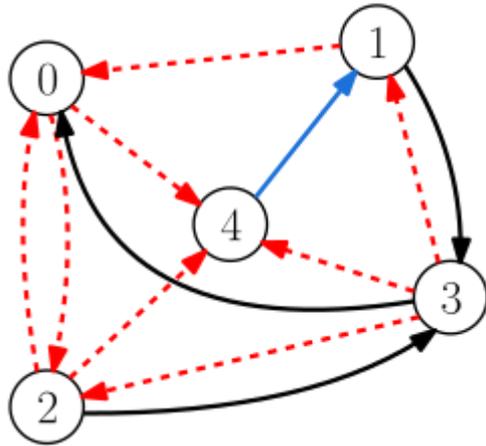
Gruppe	Punkte	Limits
1	12	$M = 0$ und $K = 1$
2	10	$M = 0$ und $K = 2$
3	19	$K = 0$
4	13	$N \leq 100$
5	17	Es gibt eine Lösung mit 0 als Wurzel des Baums
6	11	$M = 0$
7	18	Keine weiteren Einschränkungen

## Beispiel

Das folgende Diagramm zeigt die ersten beiden Beispieltestfällen. Blaue Kanten stellen schon gelegte Rohre dar und gestrichelte, rote Kanten bezeichnen Rohre, welche nicht gelegt werden dürfen.

Die linke Figur zeigt das erste Beispiel mit der Lösung aus der Beispielausgabe. Hierbei stellen schwarze Kanten neu gelegte Rohre dar (also zusätzlich zu dem schon gelegten Rohr von 4 nach 1, welches blau ist). In diesem Netzwerk wird der ganze Müll in Gebäude 0 gesammelt. Dies ist nicht die einzige Lösung. Zum Beispiel kann das Rohr von 1 nach 3 durch ein Rohr von 0 nach 1 ersetzt werden und es handelt sich dabei immer noch um eine Lösung.

Für das zweiten Beispiel sehen wir an der rechten Figur, dass es wegen des Zyklus (2,3,4) unmöglich ist eine Lösung zu konstruieren.



Eingabe	Ausgabe
<pre> 5 1 8 4 1 3 1 3 4 3 2 0 2 0 4 2 4 1 0 2 0 </pre>	<pre> 4 1 3 0 1 3 2 3 </pre>
<pre> 5 4 0 1 0 2 3 3 4 4 2 </pre>	<pre> NO </pre>
<pre> 3 0 1 0 1 </pre>	<pre> 1 0 2 0 </pre>
<pre> 4 0 2 0 1 1 0 </pre>	<pre> 2 0 3 0 1 3 </pre>