

# Problem Fake News

Der Präsident von Utopia, Präsident Trumpf, ist für seine großen Erfolge über Landesgrenzen hinweg bekannt. U. a. hat er anderen Ländern geholfen, bessere *Deals* für den Handel mit seinem eigenen Land auszuarbeiten, er setzt sich tatkräftig aufgrund seiner ausgezeichneten diplomatischen Fähigkeiten für den Weltfrieden ein und kämpft an vorderster Front gegen den weltweiten Klimawandel. Daher nennen ihn die Menschen in Utopia oft auch nur *very stable genius*.

Präsident Trumpf hat jedoch ein Problem: Die Menschen in Utopia lieben es über ihren hochgeschätzten Präsidenten zu reden, verstehen allerdings seine Aussagen und Politik falsch. Somit entstehen schädliche *Fake News*. Um das Problem anzugehen, beauftragt Präsident Trumpf, John, einen Mitarbeiter der Nationalen Superhelden Agentur (NSA), ein System zu entwickeln, sodass alle Telefonleitungen und somit alle Gespräche in Utopia überwacht werden können. Eine Telefonleitung gilt als überwacht, wenn einer der beiden Anschlüsse mit einem Überwachungssystem modifiziert wurde.

Leider stehen John nur begrenzte finanzielle Mittel zur Verfügung, da das meiste Geld für ein Denkmal des Präsidenten Trumpf in Form einer Mauer an der Grenze zu Mexika fließt. Um das Projekt dennoch umzusetzen, versucht John so wenig Anschlüsse wie möglich anzuzapfen. Helfen Sie John das System umzusetzen, indem Sie ein Programm schreiben, welches John hilft eine möglichst minimale Anzahl an Anschlüssen zu finden, um alle Leitungen zu überwachen.

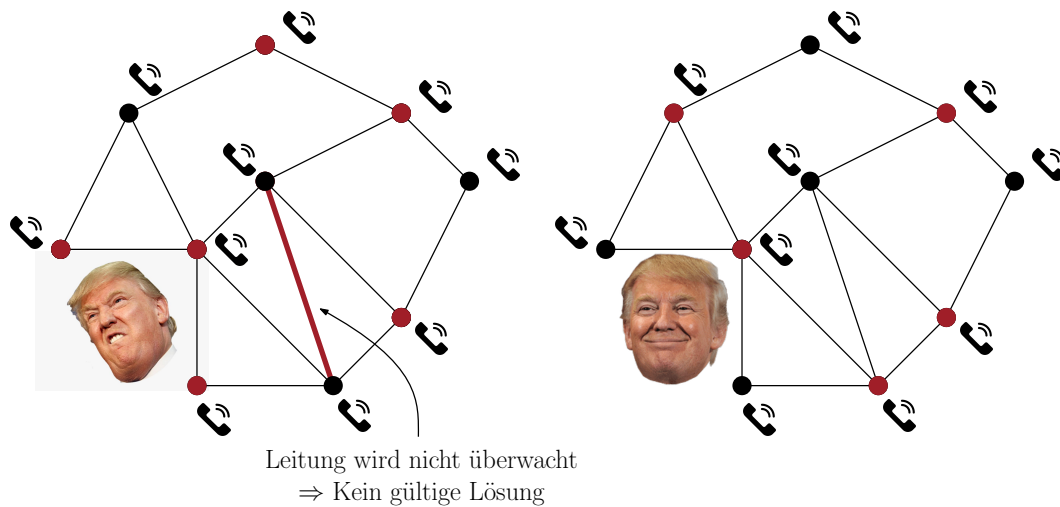


Figure 1: Zwei Beispiellösungen. Rote Knotenpunkte stellen überwachte Telefonanschlüsse dar. Im linken Beispiel wird eine Leitung nicht überwacht und ist daher keine gültige Lösung. Im rechten Beispiel werden 5 Telefonanschlüsse überwacht und alle Leitungen sind abgedeckt. Des Weiteren gibt es keine andere Lösung mit weniger angezapften Anschlüssen.

*Alle Ähnlichkeiten mit lebenden Personen und realen Handlungen sind rein zufällig.*

## Eingabeformat

Sie erhalten von uns jeweils Programme (in *Python*, *Java* und *C++*), die das Einlesen der Eingabe für Sie übernehmen. Falls die Sprache Ihrer Wahl nicht abgedeckt ist, lesen Sie die Eingabe wie folgt ein:

Die erste Zeile der Eingabe enthält die Anzahl an Telefonanschlüssen  $150 \leq n \leq 75000$  und die Anzahl an Leitungen  $300 \leq m \leq 75000$ . Es folgen  $m$  Zeilen in der Form  $1 \leq u, v \leq n$ , welche die Leitungen spezifizieren.

Es wird garantiert, dass zwischen zwei Anschlüssen nur eine Leitung existiert. Des Weiteren ist kein Anschluss mit sich selbst über eine Leitung verbunden.

## Ausgabeformat

Die erste Zeile enthält die Anzahl  $k$  an angezapften Leitungen. In der nächsten Zeile schreiben Sie die Nummern der  $k$  angezapften Leitungen. Erstellen Sie für jede Eingabedatei eine Ausgabedatei mit dem Namen `<input-file-name>.sol`.

## Beispieleingabe

```
p td 4 4      # Ignorieren Sie p td, n = 4 und m = 4
1 2
2 3
3 4
4 1
```

## Beispielausgabe

```
2
1 3
```

## Hinweis

Lösungen die nicht minimal sind, sind auch gültige Lösungen.